Adaptive Sampling for Fast Constrained Maximization of Submodular Functions

Francesco Quinzan
Hasso Plattner Institute
Potsdam, Germany

Vanja Doskoč
Hasso Plattner Institute
Potsdam, Germany

Andreas Göbel
Hasso Plattner Institute
Potsdam, Germany

Tobias Friedrich
Hasso Plattner Institute
Potsdam, Germany

Abstract

Several large-scale machine learning tasks, such as data summarization, can be approached by maximizing functions that satisfy submodularity. These optimization problems often involve complex side constraints, imposed by the underlying application. In this paper, we develop an algorithm with poly-logarithmic adaptivity for non-monotone submodular maximization under general side constraints. The adaptive complexity of a problem is the minimal number of sequential rounds required to achieve the objective.
Our algorithm is suitable to maximize a non-monotone submodular function under a p-system side constraint, and it achieves a $(p + \mathcal{O}(\sqrt{p}))$-approximation for this problem, after only poly-logarithmic adaptive rounds and polynomial queries to the valuation oracle function. Furthermore, our algorithm achieves a $(p + \mathcal{O}(1))$-approximation when the given side constraint is a p-extendible system.
This algorithm yields an exponential speed-up, with respect to the adaptivity, over any other known constant-factor approximation algorithm for this problem. It also competes with previous known results in terms of the query complexity. We perform various experiments on various real-world applications. We find that, in comparison with commonly used heuristics, our algorithm performs better on these instances.

Proceedings of the 24$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 Introduction

Several machine learning optimization problems consist of maximizing submodular functions. Examples include subset selection [Das and Kempe, 2018], data summarization [Lin and Bilmes, 2010, Mirzasoleiman et al., 2016], and Bayesian experimental design [Chaloner and Verdinelli, 1995, Krause et al., 2008]. These problems often involve constraints imposed by the underlying application. For instance, in video summarization tasks several constraints on the solution space arise based on qualitative features and contextual information [Mirzasoleiman et al., 2016].

The problem of maximizing a submodular function is NP-hard [Feige, 1998]. However, several approximation algorithms for this problem have been discovered over the years. For monotone submodular functions, the classical result of Nemhauser et al. [1978] shows that a simple greedy algorithm provides a $(1 - 1/e)$-approximation guarantee for the maximization of a *monotone* submodular function under a uniform constraint. If an additional matroid constraint is imposed on the solution space, then greedy achieves a $(1/2)$-approximation guarantee on this problem [Fisher et al., 1978]. A constant-factor approximation guarantee can also be achieved in the case of a knapsack constraint [Sviridenko, 2004].
More complex constraints require more complex heuristics. Several algorithms have been discovered, to maximize a monotone submodular function under general side constraints such as p-systems and multiple knapsacks [Badanidiyuru and Vondrák, 2014, Chekuri and Pál, 2005]. These algorithms include streaming algorithms [Badanidiyuru et al., 2014, Chekuri et al., 2015, Chakrabarti and Kale, 2015], centralized algorithms [Badanidiyuru and Vondrák, 2014, Mirzasoleiman et al., 2015], and distributed algorithms [Mirzasoleiman et al., 2013, Kumar et al., 2015].

Many algorithms have also been proposed, to maximize *non-monotone* submodular functions under a variety of constraints [Feldman et al., 2011, Chekuri et al., 2014,

Table 1: Results for non-monotone submodular maximization with a p-system side constraint. Here, n is the problem size, r is the maximum size of a feasible solution, and p the the parameter for the side constraint. The results on the adaptivity for previously known algorithms follow from the adaptivity of the greedy algorithm. Note also that all bounds on the adaptivity and query complexity for p-systems are parameterized by p. Whether it is possible to obtain bounds independent of p for this problem remains an open question.

	p-systems		
Algorithm	**Approx.**	**Adaptivity**	**Query Complexity**
REP-SAMPLING [this work]	$\approx p + \mathcal{O}\left(\sqrt{p}\right)$	$\mathcal{O}\left(\sqrt{p}\log n \log \frac{r}{p} \log r\right)$	$\mathcal{O}\left(\sqrt{p} n \log n \log \frac{r}{p} \log r\right)$
FASTSGS [Feldman et al., 2020]	$\approx p + \mathcal{O}\left(\sqrt{p}\right)$	$\mathcal{O}\left(pn \log n\right)$	$\mathcal{O}\left(pn \log n\right)$.
SIMULTANEOUSGREEDYS [Feldman et al., 2020]	$p + \mathcal{O}\left(\sqrt{p}\right)$	$\mathcal{O}\left(\sqrt{p} r\right)$	$\mathcal{O}\left(prn\right)$
REPEATEDGREEDY [Feldman et al., 2017]	$p + \mathcal{O}\left(\sqrt{p}\right)$	$\mathcal{O}\left(\sqrt{p} r\right)$	$\mathcal{O}\left(\sqrt{p} nr\right)$
FANTOM [Mirzasoleiman et al., 2016]	$\approx 2p$	$\mathcal{O}\left(pr\right)$	$\mathcal{O}\left(pnr\right)$
REPEATEDGREEDY [Gupta et al., 2010]	$\approx 3p$	$\mathcal{O}\left(pr\right)$	$\mathcal{O}\left(pnr\right)$
	p-extendable systems		
REP-SAMPLING [this work]	$\approx p + \mathcal{O}\left(1\right)$	$\mathcal{O}\left(\log n \log^2 r\right)$	$\mathcal{O}\left(n \log n \log^2 r\right)$
FASTSGS [Feldman et al., 2020]	$\approx p + \mathcal{O}\left(1\right)$	$\mathcal{O}\left(p^2 n \log n\right)$	$\mathcal{O}\left(p^2 n \log n\right)$.
SIMULTANEOUSGREEDYS [Feldman et al., 2020]	$p + \mathcal{O}\left(1\right)$	$\mathcal{O}\left(pr\right)$	$\mathcal{O}\left(p^2 rn\right)$
SAMPLEGREEDY [Feldman et al., 2017]	$p + \mathcal{O}\left(1\right)$	$\mathcal{O}\left(r\right)$	$\mathcal{O}\left(nr\right)$
	p-matchoids		
Parallel Greedy† [Chekuri and Quanrud, 2019]	$\approx \frac{4p+4}{1+o(1)}$	$\mathcal{O}\left(\log n \log r\right)$	$\mathcal{O}\left(n \log n \log r\right)$

†The Parallel Greedy algorithm requires access to the rank oracle for the underlying p-matchoid system. This oracle is strictly less general then the independence oracle required by all other algorithms in Table 1.

Gupta et al., 2010, Lee et al., 2009, Feige et al., 2011, Buchbinder et al., 2015]. These algorithms yield good approximation guarantees, but their run time is polynomial in the number of data-points, and polynomial in the number of additional side constraints.

Recently, algorithms were discovered to maximize a non-monotone submodular function under very general side constraints [Mirzasoleiman et al., 2016, Feldman et al., 2017]. These constant-factor approximation algorithms scale polynomially in the number of data-points, but also in the number of additional side constraints.

In some cases, approximation algorithms do not exhibit increasingly worse run time in the number of constraints. This is the case when maximizing a submodular function under *p-extendible systems* or *p-matchoid* side constraints [Feldman et al., 2017, Chekuri and Quanrud, 2019]. These side constraints are strictly less general than those studied in Mirzasoleiman et al. [2016], but they are general enough to capture a variety of interesting applications.

Submodular functions are learnable in the standard PAC and PMAC models [Valiant, 1984, Balcan and Harvey, 2011]: given a collection of sampled sets and their submodular function values, it is possible to produce a surrogate that mimics the behavior of that function, on samples drawn from the same distribution. However, submodular objectives cannot be optimized from the training data we use to learn them [Balkanski et al., 2017, Rosenfeld et al., 2018]. The reason is that, when learning from samples, resulting surrogate functions can be inapproximable, and their global optima can be far away from the true optimum.

Using an adaptive sampling framework [Thompson, 1990], it is possible to design algorithms that reach a constant-factor approximation guarantee in poly-logarithmic adaptive rounds for submodular maximization, both in the monotone [Balkanski and Singer, 2018a, Balkanski et al., 2019, Balkanski and Singer, 2018b, Fahrbach et al., 2019b, Ene and Nguyen, 2019] and non-monotone case [Balkanski et al., 2018, Ene et al., 2019, Fahrbach et al., 2019a]. At each adaptive round, calls to the value oracle function are queried independently. However, lower-bounds for algorithms with low adaptivity are also known [Li et al., 2020, Balkanski and Singer, 2018b].

Our contribution. Focusing on sampling techniques, we study the problem of maximizing a non-monotone submodular function, to which we have oracle access. Furthermore, we consider general p-system and p-extendible system side constraints for this problem.

Our algorithm has access to the side constraint structure via an oracle. Standard oracle models in the literature are: the *independence oracle*, which takes as input a set and returns whether that set is a feasible solution; the *rank oracle*, that returns the maximum cardinality of any feasible solution contained in a given input set; and the *span oracle*, which for an input set S and a point $\{e\}$ it returns whether or not $S \cup \{e\}$ has a higher rank than S. In this work, we assume access to the independence oracle, which is the most general oracle model of the three.

In this work, we develop the first algorithm with poly-logarithmic adaptivity suitable to maximize a non-monotone submodular function under a p-system side constraint and a p-extendible system. In contrast to all previous algorithms with low adaptivity, our algorithm only requires access to the independence oracle for the side constraints. This algorithm achieves strong approximation guarantees and run time, competing with known algorithms for this problem (see Table 1). We study the performance of our algorithm in two real-world applications, video summarization and Bayesian experimental design. We test our algorithm against other commonly used heuristics for this problem, and show that our algorithm comes out on top.

Our paper is organized as follows. We define the problem in Section 2, and we describe our algorithm in Section 3. Our theoretical analysis is presented in Sections 4-6. Applications and experiments are discussed in Sections 7-9. We conclude in Section 10.

2 Problem Description

Submodularity. In this paper, we study optimization problems that can be approached by maximizing an oracle function that, given a solution set, estimates its quality. We assume that oracle functions are *submodular*.

Definition 1 (Submodularity)**.** *Given a finite set V, we call a set function $f\colon 2^V \to \mathbb{R}_{\geq 0}$* submodular *if for all $S, U \subseteq V$ we have that $f(S) + f(U) \geq f(S \cup U) + f(S \cap U)$.*

Note that we only consider functions that do not attain negative values. This is because submodular functions with negative values cannot be maximized, even approximately (see Feige et al. [2011]).

p-Systems. We study the problem of maximizing a submodular function under additional side constraints, defined as a p-system side constraint. As discussed, i.e., in Mirzasoleiman et al. [2016], Gupta et al. [2010], these constraints are significantly more general than standard matroid intersections, and they arise in various domains, such as movie recommendation, video summarization, and revenue maximization.

Given a collection of feasible solutions $\mathcal{I}$ over a ground set V and a set $T \subseteq V$, we denote with $\mathcal{I}\mid_T$ a collection consisting of all sets $S \subseteq T$ that are feasible in $\mathcal{I}$. Furthermore, a base for $\mathcal{I}$ is any maximum feasible set $U \in \mathcal{I}$. We define p-systems as follows.

Definition 2. *A p-system $\mathcal{I}$ over a ground set V is a collection of subsets of V fulfilling the following three axioms:*

- *$\emptyset \in \mathcal{I}$;*
- *for any two sets $S \subseteq \Omega \subseteq V$, if $\Omega \in \mathcal{I}$ then $S \in \mathcal{I}$;*
- *for any set $T \subseteq V$ and any bases $S, U \in \mathcal{I}\mid_T$ it holds $|S| \leq p\,|U|$.*

The second defining axiom is referred to as subset-closure or downward-closed property. With this notation, we study the following problem.

Problem 1. *Given a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ and a p-system $\mathcal{I}$, find a set $S \subseteq V$ maximizing $f(S)$ such that $S \in \mathcal{I}$.*

p-extendible Systems. We also consider a family of side constraints of intermediate generality, commonly referred to as p-extendible systems. These side constraints are strictly less general than p-systems, but they capture various types of constraints found in practical applications.

Our main motivation in studying these constraints is that they admit algorithms that obtain strong approximation guarantees, in much less time than in the case of the p-systems. Hence, algorithms for p-extendible systems scale much better than for general p-systems.

Algorithm 1: RAND-SEQUENCE$(X, S, \mathcal{I})$

$A \leftarrow \emptyset$;
while $X \neq \emptyset$ **do**
  sort the points $\{x_i\}_i = X$ randomly;
  $\eta \leftarrow \max\{j \colon S \cup A \cup \{x_i\}_{i \leq j} \in \mathcal{I}\}$;
  $A \leftarrow A \cup \{x_1, \dots, x_\eta\}$;
  $X \leftarrow \{e \in X \setminus (S \cup A) \colon S \cup A \cup e \in \mathcal{I}\}$;
end while
return A;

Algorithm 2: RAND-SAMPLING$(f, V, \mathcal{I}, \lambda, \varepsilon, \varphi_1)$.

$S \leftarrow \emptyset$;
$X \leftarrow \arg\max_e\{f(e) \colon e \in V \wedge e \in \mathcal{I}\}$;
$\delta \leftarrow f(X)$, $\delta_0 \leftarrow \lambda f(X)$;
while $\delta \geq \delta_0$ **do**
  while $X \neq \emptyset$ **do**
    $\{a_j\}_{j \in J} \leftarrow$ RAND-SEQUENCE$(X, S, \mathcal{I})$;
    $\eta \leftarrow$ BINARY-SEARCH$(J, \min\{j \in J \colon |X_j| < (1-\varepsilon)|X|\})$ with $X_j = \{e \in X \colon f(e \mid S \cup \{a_1, \dots, a_{j-1}\}) \geq \delta \wedge S \cup \{a_1, \dots, a_{j-1}\} \cup e \in \mathcal{I}\}$
    $A \leftarrow$ UNIF-SAMPLING$(\{a_1, \dots, a_{\eta-1}\}, \varphi_1)$;
    $X \leftarrow X_\eta$; $S \leftarrow S \cup A$;
  end while
  $\delta \leftarrow (1-\varepsilon)\delta$;
  $X \leftarrow \{e \in V \colon f(e \mid S) \geq \delta \wedge S \cup e \in \mathcal{I}\}$;
end while
return S;

These p-extendible systems were first studied by Mestre [2006], and they are defined as follows.

Definition 3. *A p-extendible system $\mathcal{I}$ over a ground set V is a p-system, that fulfills the following additional axiom: for every pair of sets $S, \Omega \in \mathcal{I}$ with $S \subset \Omega$, and for every element $e \notin S$, there exists a set $U \subseteq \Omega \setminus S$ of size $|U| \leq p$ such that $\Omega \setminus U \cup \{e\} \in \mathcal{I}$.*

These side constraints generalize matroid intersections and p-matchoids. While being strictly less general than p-systems, this definition captures many interesting constraints, such as the intersection of matroids [Mestre, 2006]. In this paper, we also study the following problem.

Problem 2. *Given a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ and a p-extendible system $\mathcal{I}$, find a set $S \subseteq V$ maximizing $f(S)$ such that $S \in \mathcal{I}$.*

Adaptivity. An algorithm is T-adaptive if every query $f(S)$ for the f-value of a solution S occurs at a round $i \in [T]$ such that S is independent of the values $f(S')$ of all other queries at round i, with at most polynomial queries at each round in the problem size. The query complexity is the number of calls to the evaluation oracle function.

Notation. For any submodular evaluation oracle function $f \colon 2^V \to \mathbb{R}$ and sets $S, U \subseteq V$, we define the marginal value of S with respect to U as $f(U \mid S) = f(S \cup U) - f(S)$.
Throughout the paper, we always use the notation introduced in Problem 1: we denote with f the evaluation oracle function, with V the ground set, and with $\mathcal{I}$ the p-system side constraint. We denote with OPT a solution to Problem 1, and we denote with n the size of the ground set V, i.e., n is the number of singletons in our solution space. We also denote with r the maximum size of a feasible solution.
The notation introduced in Algorithm 1-3 is used consistently throughout the paper.

3 Algorithms

Our method consists of three parts (see Algorithms 1-3). We call these algorithms RAND-SEQUENCE, RAND-SAMPLING, and REP-SAMPLING respectively. These algorithms also call the BINARY-SEARCH and UNIF-SAMPLING sub-routines. The following is a description of each algorithm and sub-routine.

RAND-SEQUENCE. It is based on the work of Karp et al. [1988]. Given as input a ground set X, a current solution S, and a p-system $\mathcal{I}$, this algorithm finds a random set A such that $S \cup A$ is a base for $\mathcal{I}$.

RAND-SAMPLING. This algorithms generalizes a sampling algorithm proposed in Balkanski et al. [2019] to non-monotone submodular maximization. This algorithm requires as input an oracle function f, a ground set V, a p-system or p-extendible system $\mathcal{I}$, and parameters $\lambda, \varepsilon, \varphi_1$. The parameter λ determines the total number of iterations for the RAND-SAMPLING, the parameter ε determines the rate with which the variable δ decreases, whereas φ_1 determines the distribution for the UNIF-SAMPLING sub-routine. For a constant δ, points are added to the current solution yielding a marginal contribution upper-bounded by δ. Note that at each adaptive step, the RAND-SAMPLING uses the BINARY-SEARCH and the UNIF-SAMPLING sub-routine. If Algorithm 2 reaches an iteration with $X = \emptyset$, then it decreases the value of δ so that points with lower marginal contribution can be added to the current solution.

BINARY-SEARCH. This sub-routine is just the standard binary search algorithm. It is used to locate an

Algorithm 3: REP-SAMPLING$(f, V, \mathcal{I}, \varepsilon, \varphi_1, \varphi_2, m)$

1: $\lambda \leftarrow \varepsilon(p+1)/m$;
2: **for** $j \leq m$ iterations **do**
3: $\quad \Omega_j \leftarrow$ RAND-SAMPLING$(f, V, \mathcal{I}, \lambda, \varepsilon, \varphi_1)$;
4: $\quad \Lambda_j \leftarrow$ UNIF-SAMPLING(Ω_j, φ_2);
5: $\quad V \leftarrow V \setminus \Omega_j$;
6: **end for**
7: **return** $\arg\max_j \{f(\Omega_j), f(\Lambda_j)\}$;

index η such that $\eta = \min\{j \colon |X_j| \leq (1-\varepsilon)|X|\}$, with X_j, where the index j spans over the set J. This sub-routine uses the fact that, due to submodularity, it holds $|X_j| \leq |X_{j+1}|$ for all $j \in J$.

UNIF-SAMPLING. For a given input set and probability φ, this algorithm samples points of the input set independently, with probability φ.

REP-SAMPLING. This algorithm requires as input an oracle function f a ground set V, a p-system or p-extendible system $\mathcal{I}$ and parameters λ, m, ε and φ_1, φ_2. At each step, the REP-SAMPLING calls Algorithm 2 to find a partial solution Ω_j. Then, Algorithm 3 samples a subset of Ω_j, where each point is drawn independently with probability φ_2. Afterwards, the REP-SAMPLING removes all points of Ω_j from the ground set, and it runs the REP-SAMPLING on the resulting ground set. This procedure is iterated m times.

4 Analysis for p-Systems

In this section, we discuss theoretical run time analysis results for Problem 1. We remark that **all proofs can be found in the full version, see Quinzan et al. [2021]**. Approximation guarantees for Algorithm 3 follow from the following general theorem.

Theorem 1. *Fix constants $\varepsilon \in (0,1)$, $m \geq 2$, $\varphi_1 = 1$, and $\varphi_2 = 1/2$. Denote with Ω^* the output of Algorithm 3. Then,*

$$f(\mathrm{OPT}) \leq m\left(\frac{(1+\varepsilon)(p+1)}{(1-\varepsilon)^2(m-1)} + 2\right)\mathbb{E}\left[\, f(\Omega^*)\,\right].$$

A proof of this theorem is given in Quinzan et al. [2021].

We estimate the number of adaptive rounds until Algorithm 3 reaches the desired approximation guarantee. The following lemma holds.

Lemma 1. *Fix constants $\varepsilon \in (0,1)$, $\varphi_1, \varphi_2 \in [0,1]$ and $m \geq 0$. Then Algorithm 3 terminates after $\mathcal{O}\left(\frac{m}{\varepsilon^2}\log\left(\frac{r}{p\varepsilon}\right)\log r \log n\right)$ rounds of adaptivity. Furthermore, Algorithm 3 has query complexity of $\mathcal{O}\left(\frac{mn}{\varepsilon^2}\log\left(\frac{r}{p\varepsilon}\right)\log r \log n\right)$.*

A proof of this result is given in Quinzan et al. [2021]. The following lemma follows from Theorem 1 and Lemma 1.

Lemma 2. *Fix a constant $\varepsilon \in (0,1)$, and define parameters $m = 1 + \lceil\sqrt{(p+1)/2}\rceil$, $\varphi_1 = 1$, and $\varphi_2 = 1/2$. Denote with Ω^* the optimal solution found by Algorithm 3. Then,*

$$f(\mathrm{OPT}) \leq \frac{1-\varepsilon}{(1-\varepsilon)^2}\left(p + 2\sqrt{2(p+1)} + 5\right)\mathbb{E}\left[\, f(\Omega^*)\,\right].$$

Furthermore, with this parameter choice Algorithm 3 terminates after $\mathcal{O}\left(\frac{\sqrt{p}}{\varepsilon^2}\log n \log\left(\frac{r}{p\varepsilon}\right)\log r\right)$ rounds of adaptivity, and its query complexity is $\mathcal{O}\left(\frac{\sqrt{p}n}{\varepsilon^2}\log n \log\left(\frac{r}{p\varepsilon}\right)\log r\right)$.

A proof is given in Quinzan et al. [2021]. We remark that there exists an algorithm with constant adaptivity for unconstrained non-monotone submodular maximization that achieves an approximation guarantee arbitrarily close to 1/2 (see Chen et al. [2019]). Using this algorithm as a sub-routine in line 4 of Algorithm 3 yields a constant-factor improvement over the approximation guarantee of Lemma 2, without affecting the upper-bound on the adaptivity. However, this algorithm requires access to a continuous extension of the value oracle f, whereas Algorithm 3 only requires access to f.

5 Analysis for p-extendible Systems

In this section, we perform the theoretical analysis for the REP-SAMPLING, when maximizing a non-monotone submodular function under a p-extendible system side constraint, as in Problem 2. We prove that, with different sets of input parameters, our algorithm has adaptivity and query complexity that is not dependent on p. Again, all proofs can be found in the full version, see Quinzan et al. [2021]. The following theorem holds.

Lemma 3. *Fix parameters $\varepsilon \in (0,1)$, $m = 1$, $\varphi_1 = (p+1)^{-1}$, and $\varphi_2 \in [0,1]$. Denote with Ω^* the output of Algorithm 3. Then,*

$$f(\mathrm{OPT}) \leq \frac{(1+\varepsilon)(p+1)^2}{p(1-\varepsilon)^2}\mathbb{E}\left[\, f(\Omega^*)\,\right].$$

With this parameter choice, Algorithm 3 terminates after $\mathcal{O}\left(\varepsilon^{-2}\log n \log\left(\frac{r}{\varepsilon}\right)\log r\right)$ rounds of adaptivity, and it requires $\mathcal{O}\left(\frac{n}{\varepsilon^2}\log n \log\left(\frac{r}{\varepsilon}\right)\log r\right)$ function evaluations.

For a proof of this result see Quinzan et al. [2021]. The proof of this lemma is based on the work of Feldman et al. [2017], together with the fact that Algorithm 2 yields expected marginal increase lower-bounded by the best possible greedy improvement, up to a multiplicative constant.

We remark that Lemma 3 also holds when side constraints are p-matchoids and the intersections of matroids, since p-extendible systems are a generalization of both.

6 Query Complexity and Adaptivity of the Independence Oracle

We conclude our analysis with a general discussion on the performance of Algorithm 3 in the number of calls to the independence oracle for the p-system constraint. The independence oracle takes as input a set S, and returns as output a Boolean value, true if the given set is independent in $\mathcal{I}$ and false otherwise. The following lemma holds.

Lemma 4. *Fix parameters $\varepsilon \in (0,1)$, $m \geq 1$, and $\varphi_1, \varphi_2 \in [0,1]$. Then Algorithm 3 requires expected $\mathcal{O}\left(\frac{m\sqrt{n}}{\varepsilon^2}\log\left(\frac{r}{p\varepsilon}\right)\log r \log n\right)$ rounds of independent calls to the oracle for the p-system constraint. Furthermore, the total number of calls to the independence system is $\mathcal{O}\left(\frac{mn^{3/2}}{\varepsilon^2}\log\left(\frac{r}{p\varepsilon}\right)\log r \log n\right)$.*

A proof of this result is given in Quinzan et al. [2021], and it follows from the work of Karp et al. [1988]. Note that the rounds of independent calls to the oracle are sub-linear, but not poly-logarithmic in the problem size. The reason is that Algorithm 1 requires $\mathcal{O}\left(\sqrt{n}\right)$ rounds of independent calls to the oracle for the p-system. We are not aware of any algorithm that finds a base in less than $\mathcal{O}\left(\sqrt{n}\right)$ rounds. Furthermore, it is well-known that there is no algorithm that obtains an approximation guarantee that is constant in the problem size for Problem 1, than $\tilde{\Omega}(n^{1/3})$ steps of independent calls to the oracle for the p-system constraint (see Karp et al. [1988], Balkanski et al. [2019]).

For a p-system $\mathcal{I}$, the rank of a set S is the maximum cardinality of its intersection with a maximum independent set in $\mathcal{I}$. Given access to an oracle that returns the rank of a set in $\mathcal{I}$, it is possible to design an algorithm that finds a maximum independent set of a p-system in $\mathcal{O}\left(\log n^2\right)$ rounds of independent calls to the rank oracle (see Karp et al. [1988]). However, this work focuses on general constraints where the rank of a set is not known.

7 Experimental Framework

In our set of experiments, we implement the REPSAMPLING as describe in Algorithm 3. We always test our algorithm against these algorithms:

- FANTOM. This algorithm, which iterates a density greedy algorithm multiple times, is studied in Gupta et al. [2010] and Mirzasoleiman et al. [2016].
- REPEATEDGREEDY. This algorithm, studied in Feldman et al. [2017], consists of iterating a greedy algorithm multiple times. It uses Algorithm 1 in Buchbinder et al. [2015] as a sub-routine.
- FASTSGS. This algorithm is studied in Feldman et al. [2020], and it is essentially a fast implementation of the SIMULTANEOUSGREEDYS Feldman et al. [2020]. This algorithm updates multiple solutions concurrently, and it picks the best of them.
- SAMPLEGREEDY. This algorithm is specifically designed to handle p-extendible systems (see Feldman et al. [2017]). This algorithm samples points independently at random, and then it builds a greedy solution over the resulting set.

Note that these algorithms only require access to the independence oracle for the side constraints. In our experiments we do not consider algorithms that require access to the rank oracle, since they are impractical for our applications. We perform two sets of experiments, on the following applications:

- **Video Summarization**. This problem asks to find a set of representative frames for a given video. We use Determinantal Point Processes to select a diverse set of frame. In order to get better summaries, we employ a face-recognition tool to identify faces in each segment. This experiment is described in Section 8, and the results are displayed in Figure 1.
- **Bayesian D-Optimality**. Here, the goal is to design an experiment that maximizes the expected utility of the outcome, using preliminary observations. We use observations from the Berkeley Earth data-set to select thermal stations around the world, to measure the temperature with. This experiment is described in Section 9, and the results are displayed in Figure 2.

The code and the datasets are available upon request.

8 Video Summarization

We study an application of our setting to a data summarization task: Given a video consisting of ordered frames, choose a subset of frames that gives a descriptive overview of the video. An effective way to select a diverse set of items is to apply Determinantal Point

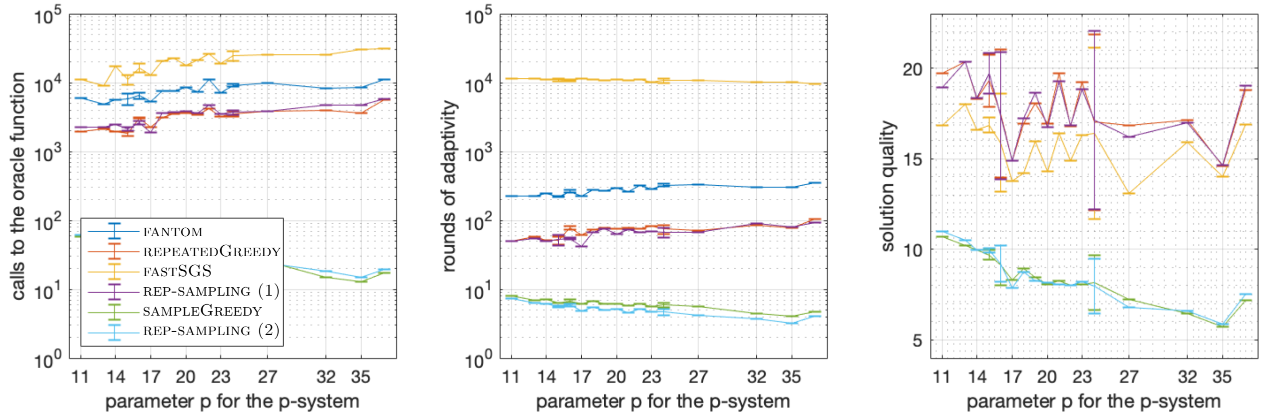


Figure 1: Results for the experiments on Video Summarization on movie segments taken from FLIC [Sapp and Taskar, 2013]. Each plot shows the average performance over segments with fixed p. Error bars correspond to the best and worst case. Note that the y-axis in the two leftmost plots uses a logarithmic scale. The FastSGS uses parameters $\ell = \lfloor 2 + \sqrt{p+1} \rfloor$ and $\varepsilon = 0.1$; the rep-sampling (1) uses parameters $\varepsilon = 0.1$, $m = 1 + \lceil \sqrt{(p+1)/2} \rceil$, $\varphi_1 = 1$, $\varphi_2 = 0.5$; the rep-sampling (2) uses parameters $\varepsilon = 0.01$, $m = 1$, $\varphi_1 = (1+p)^{-1}$, $\varphi_2 = 1$.

Processes [Macchi, 1975]. For a thorough survey on Determinantal Point Processes and their applications, we refer the reader to Kulesza and Taskar [2012].

For a set of items $V = \{1, \ldots, n\}$, a Determinantal Point Process (DPP) defines a discrete probability distribution over all subsets $S \subseteq V$ as $\mathsf{Pr}(S) = \det_L(S)/\det(L+I)$, where $L \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix, $\det_L(S) := \det((L_{i,j})_{i,j \in S})$ is the determinant of the sub-matrix of L indexed by S and I is the $n \times n$-identity matrix. Intuitively, if L expresses pairwise similarity, then the DPP prefers diversity.

In our setting, each item corresponds to a frame of a video segment. For each frame i, we compute a feature vector $\mathbf{f}_i$, consist of both visual features, such as color and SIFT features [Kulesza and Taskar, 2011], and qualitative information, such as size, colorfulness and luminosity. Following Gong et al. [2014], we parameterize L as $L_{i,j} := z_i^T W^T W z_j$, where $z_i := \tanh(U\mathbf{f}_i)$. We then learn the parameters U and W using a neural network.

We select a representative summary by maximizing the function $\log\det_L(S)$. We impose the following additional side constraints. First, we impose an upper-bound on the maximum number of frames of each summary. Then, we partition each video into segments, and define a partition matroid to select at most ℓ_j frames in each segment j. Following Mirzasoleiman et al. [2018], Feldman et al. [2018], we also use a face-recognition tool to identify actors in each movie, and select a summary containing at most k_i frames showing face i. This additional constraint corresponds to a p-system $\mathcal{I} = \{S \subseteq V : |S \cap V_i| \leq k_i\}$, with V_i all frames containing face i.[1] In our experiments, the parameters k_i are always set to a fixed constant for all videos. Hence, the only variable that affects p is the total number of distinct faces in each movie.

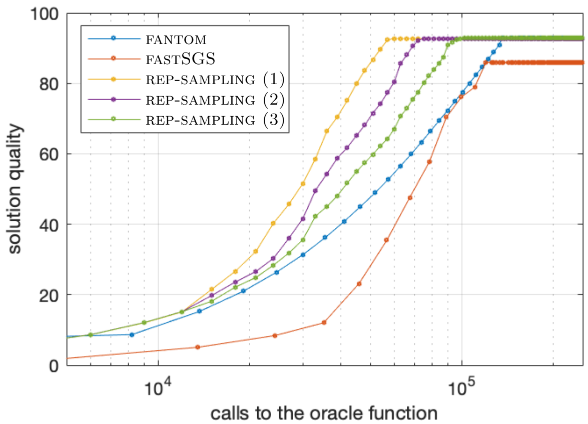
For our experimental investigation, we use movies from the Frames Labeled In Cinema (FLIC) data-set [Sapp and Taskar, 2013]. We consider all movies in this data-set with at least 200 frames, as to highlight performance when dealing with large problem size.

The results are displayed in Figure 1, where we describe the parameter choice for each algorithm. For each non-deterministic algorithm, results are the sample mean of 100 independent runs. We observe that, for different parameter choice, our algorithm outperforms fantom and the FastSGS, and it has better adaptivity than the greedy algorithms. The solution quality for the sampleGreedy and rep-sampling, with parameters as in Lemma 3, is worse on these instances.

9 Bayesian D-Optimality

Bayesian experimental design provides a general framework to select a set of experiments, that maximize the expected utility of the outcome. Formally, we want to estimate the parameter θ of a function $y = f_\theta(x) + w$, where w is an error. In this framework, the input x is generated by a set of experiments. Assuming that parameters are equipped with a prior, Bayesian optimality criteria are useful in identifying the right experiments

[1] The parameter p is estimated by counting the total number of distinct faces i that appear in more than k_i frames.



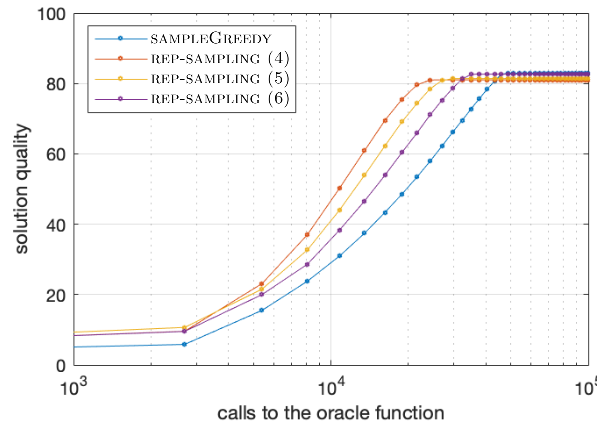


Figure 2: Solution quality for fixed time budget for Bayesian D-Optimality. Results for the randomized algorithms are the average over 100 independent runs. Note that the x-axis of both plots uses a logarithmic scale. For a limited number of calls to the oracle function, the REP-SAMPLING yields best performance. The REP-SAMPLING (1)-(3) use parameters $m = 1 + \lceil\sqrt{(p+1)/2}\rceil, \varphi_1 = 1, \varphi_2 = 0.5$ and $\varepsilon = 0.7, 0.5, 0.3$ respectively. The REP-SAMPLING (4)-(6) use parameters $m = 1, \varphi_1 = (p+1)^{-1}, \varphi_2 = 1$ and, again, $\varepsilon = 0.7, 0.5, 0.3$. The FASTSGS uses parameters $\ell = \lfloor 2 + \sqrt{p+1}\rfloor$ and $\varepsilon = 0.25$.

to perform, in order to generate the input x.

We focus on linear regressions of the form $y = \theta^T X + w$, with $y, w \in \mathbb{R}^n$, $\theta \in \mathbb{R}^m$ and $X \in \mathbb{R}^{m\times n}$. Furthermore, we assume independent and homoscedastic noise. We approach experimental design with the D-optimality criterion, although other methods can be used to this end [Krause et al., 2008]. This criterion consists of maximizing the determinant of the Fisher information matrix. As shown in Sebastiani and Wynn [2000], for regressions as described above the D-optimality criterion is equivalent to maximizing the entropy.

We apply the Bayesian D-optimality criterion to the following setting. Consider a data-set consisting of monthly temperatures measured by thermal stations at different locations, over a period of time. We want to collect data to perform a regression for a model explaining the temperature variation from one measurements to the other one. Here, collecting temperatures with a single station corresponds to performing an experiment, and the goal is to identify appropriate stations to perform future measurements with.

Assuming independent and homoscedastic noise, we search for a feasible set of stations maximizing the entropy. Since temperature variation series follow a Gaussian process [Krause et al., 2008, Friedrich et al., 2019], the entropy is defined as $\mathcal{H}(S) = \frac{1+\ln(2\pi)}{2}|S| + \frac{1}{2}\ln\det_\Sigma(S)$, with S a subset of stations, and Σ the covariance matrix. The function $\det_\Sigma(S)$ is the determinant of the covariance matrix corresponding to a set S of stations. Note that the function $\mathcal{H}(S)$ is submodular and non-monotone. We consider an upper-bound on the solution size as a side constraint. Furthermore, we group stations that are located in the same geographical area, and we impose an upper-bound on the number of stations that can be chosen in each group. This additional constraint is useful when stations are not distributed uniformly across a territory (see Friedrich et al. [2019]). In our experiment, geographical areas correspond to continents. We remark that, if only a single cardinality constraint is given, then Bayesian optimality criteria can be optimized well via regularized Determinantal Point Processes [Derezinski et al., 2020].

For our experiments we consider the Berkeley Earth climate data-set (http://berkeleyearth.org/data/). This data-set combines 1.6 billion temperature reports from 16 preexisting data archives, for over 39,000 unique stations worldwide. We run all algorithms with parameters described as in Figure 2 for a fixed time budget.

In Figure 2, we report on the average solution quality achieved by each algorithm, after a fixed number of oracle calls. We observe that the REP-SAMPLING gets to a good solution more quickly that the other algorithms. All algorithms find similar solution qualities, for unlimited time budget, with the FANTOM and REP-SAMPLING. slightly outperforming the other algorithms.

10 Conclusion

In this paper, we develop the first algorithm for non-monotone submodular maximization under p-system and p-extendible system side constraints, with polylogarithm adaptivity (see Lemma 2 and Theorem 3).

This algorithm also competes with previous known results in terms of the query complexity and approximation guarantee (see Table 1).
We consider two applications and study the performance of our algorithm against several other algorithms suitable for this problem. We observe that our algorithms has superior adaptivity, and that it competes in terms of the query complexity (see Figure 1-2).

11 Acknowledgements

We would like to thank Christopher Weyand for helping the authors to develop some of the code used for the experiments. We would like to thank Martin Schirneck for useful discussion on previous related work. This research has been partly funded by the Federal Ministry of Education and Research of Germany in the framework of KI-LAB-ITSE (project number 01IS19066).

References

Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Proc. of SODA*, pages 1497–1514, 2014.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: massive data summarization on the fly. In *Proc. of KDD*, pages 671–680, 2014.

Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In *proc. of STOC*, pages 793–802, 2011.

Eric Balkanski and Yaron Singer. Approximation guarantees for adaptive sampling. In *Proc. of ICML*, pages 393–402, 2018a.

Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In *Proc. of STOC*, pages 1138–1151, 2018b.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The limitations of optimization from samples. In *Proc. of STOC*, pages 1016–1027, 2017.

Eric Balkanski, Adam Breuer, and Yaron Singer. Non-monotone submodular maximization in exponentially fewer iterations. In *Proc. of NeurIPS*, pages 2359–2370, 2018.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An optimal approximation for submodular maximization under a matroid constraint in the adaptive complexity model. In *Proc. of STOC*, pages 66–77, 2019.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal of Computing*, 44(5):1384–1402, 2015.

Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Mathematical Programming*, 154(1-2):225–247, 2015.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10:273–304, 1995.

Chandra Chekuri and Martin Pál. A recursive greedy algorithm for walks in directed graphs. In *Proc. of FOCS*, pages 245–253, 2005.

Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In Moses Charikar and Edith Cohen, editors, *Proc. of STOC*, pages 78–89, 2019.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal of Computing*, 43(6):1831–1879, 2014.

Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *Proc. of ICALP*, pages 318–330, 2015.

Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proc. of STOC*, pages 102–113, 2019.

Abhimanyu Das and David Kempe. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *Journal of Machine Learning Research*, 19:3:1–3:34, 2018.

Michal Derezinski, Feynman Liang, and Michael W. Mahoney. Bayesian experimental design using regularized determinantal point processes. In *Proc. of AISTATS*, volume 108, pages 3197–3207, 2020.

Alina Ene and Huy L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In *Proc. of SODA*, pages 274–282, 2019.

Alina Ene, Huy L. Nguyen, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In *Proceedings of STOC*, pages 90–101, 2019.

Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In *Proc. of ICML*, pages 1833–1842, 2019a.

Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In *Proc. of SODA*, pages 255–273, 2019b.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal of Computing*, 40(4):1133–1153, 2011.

Moran Feldman, Joseph Naor, and Roy Schwartz. Nonmonotone submodular maximization via a structural continuous greedy algorithm - (extended abstract). In *Proc. of ICALP*, pages 342–353, 2011.

Moran Feldman, Christopher Harshaw, and Amin Karbasi. Greed is good: Near-optimal submodular maximization via greedy optimization. In *Proc. of COLT*, pages 758–784, 2017.

Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with subsampling. In *Proc. of NeurIPS*, pages 730–740, 2018.

Moran Feldman, Christopher Harshaw, and Amin Karbasi. Simultaneous greedys: A swiss army knife for constrained submodular maximization. *CoRR*, abs/2009.13998, 2020.

M. Fisher, George Nemhauser, and Laurence Wolsey. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming*, 14:73–87, 03 1978.

Tobias Friedrich, Andreas Göbel, Frank Neumann, Francesco Quinzan, and Ralf Rothenberger. Greedy maximization of functions with bounded curvature under partition matroid constraints. In *Proc. of AAAI*, pages 2272–2279, 2019.

Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Proc. of NIPS*, pages 2069–2077, 2014.

Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *Proc. of WINE*, pages 246–257, 2010.

Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel search. *Journal of Computer System Science*, 36(2):225–253, 1988.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9: 235–284, 2008.

Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proc. of ICML*, pages 1193–1200, 2011.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012.

Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. *ACM Transaction on Parallel Computing*, 2(3):14:1–14:22, 2015.

Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proc. of STOC*, pages 323–332, 2009.

Wenzheng Li, Paul Liu, and Jan Vondrák. A polynomial lower bound on adaptive complexity of submodular maximization. *CoRR*, abs/2002.09130, 2020.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Proc. of NAACL-HLT*, pages 912–920, 2010.

Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7 (1):83–122, 1975. doi: 10.2307/1425855.

Julián Mestre. Greedy in approximation algorithms. In *Proc. of ESA*, pages 528–539, 2006.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Proc. of NIPS*, pages 2049–2057, 2013.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proc. of AAAI*, pages 1812–1818, 2015.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *Proc. of ICML*, pages 1358–1367, 2016.

Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proc. of AAAI*, pages 1379–1386, 2018.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

Francesco Quinzan, Vanja Doskoč, Andreas Göbel, and Tobias Friedrich. Adaptive sampling for fast constrained maximization of submodular functions. *CoRR*, 2102.06486, 2021. ArXiv preprint (full version).

Nir Rosenfeld, Eric Balkanski, Amir Globerson, and Yaron Singer. Learning to optimize combinatorial functions. In *Proc. of ICML*, pages 4371–4380, 2018.

Benjamin Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proc. of CVPR*, 2013.

P. Sebastiani and H.P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operation Research Letters*, 32(1):41–43, 2004.

Steven K. Thompson. Adaptive cluster sampling. *Journal of the American Statistical Association*, 85(412): 1050–1059, 1990.

Leslie G. Valiant. A theory of the learnable. *Communication of the ACM*, 27(11):1134–1142, 1984.