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 \( (p + O(\sqrt{p})) \)-approximation for this problem, after only poly-logarithmic adaptive rounds and polynomial queries to the valuation oracle function. Furthermore, our algorithm achieves a \( (p + O(1)) \)-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.

1 Introduction

Several large-scale 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Approx.</th>
<th>Adaptivity</th>
<th>Query Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$p$-systems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>REP-SAMPLING</strong></td>
<td>$\approx p + \mathcal{O}(\sqrt{p})$</td>
<td>$\mathcal{O}(\sqrt{p} \log n \log \frac{L}{p} \log r)$</td>
<td>$\mathcal{O}(\sqrt{pn} \log n \log \frac{L}{p} \log r)$</td>
</tr>
<tr>
<td>[Feldman et al., 2020]</td>
<td>$\approx p + \mathcal{O}(\sqrt{p})$</td>
<td>$\mathcal{O}(pn \log n)$</td>
<td>$\mathcal{O}(pn \log n)$</td>
</tr>
<tr>
<td><strong>SIMULTANEOUS-GREEDYS</strong></td>
<td>$p + \mathcal{O}(\sqrt{p})$</td>
<td>$\mathcal{O}(\sqrt{pr})$</td>
<td>$\mathcal{O}(\sqrt{pn}r)$</td>
</tr>
<tr>
<td>[Feldman et al., 2017]</td>
<td>$p + \mathcal{O}(\sqrt{p})$</td>
<td>$\mathcal{O}(\sqrt{pr})$</td>
<td>$\mathcal{O}(\sqrt{pn}r)$</td>
</tr>
<tr>
<td><strong>FACTOR</strong></td>
<td>$\approx 2p$</td>
<td>$\mathcal{O}(pr)$</td>
<td>$\mathcal{O}(pnr)$</td>
</tr>
<tr>
<td>[Mirzasoleiman et al., 2016]</td>
<td>$\approx 3p$</td>
<td>$\mathcal{O}(pr)$</td>
<td>$\mathcal{O}(pnr)$</td>
</tr>
<tr>
<td><strong>REPEATED-GREEDY</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Gupta et al., 2010]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>$p$-extendable systems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>REP-SAMPLING</strong></td>
<td>$\approx p + \mathcal{O}(1)$</td>
<td>$\mathcal{O}(\log n \log^2 r)$</td>
<td>$\mathcal{O}(n \log n \log^2 r)$</td>
</tr>
<tr>
<td>[this work]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FASTSGS</strong></td>
<td>$\approx p + \mathcal{O}(1)$</td>
<td>$\mathcal{O}(p^2 n \log n)$</td>
<td>$\mathcal{O}(p^2 n \log n)$</td>
</tr>
<tr>
<td>[Feldman et al., 2020]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SIMULTANEOUS-GREEDYS</strong></td>
<td>$p + \mathcal{O}(1)$</td>
<td>$\mathcal{O}(pr)$</td>
<td>$\mathcal{O}(p^2 r n)$</td>
</tr>
<tr>
<td>[Feldman et al., 2020]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SAMPLE-GREEDY</strong></td>
<td>$p + \mathcal{O}(1)$</td>
<td>$\mathcal{O}(r)$</td>
<td>$\mathcal{O}(n r)$</td>
</tr>
<tr>
<td>[Feldman et al., 2017]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parallel Greedy</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>†[Chekuri and Quanrud, 2019]</td>
<td>$\approx \frac{4p+4}{1+\alpha(1)}$</td>
<td>$\mathcal{O}(\log n \log r)$</td>
<td>$\mathcal{O}(n \log n \log r)$</td>
</tr>
</tbody>
</table>

†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.

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. These algorithms scale polynomially in the number of data-points, but they are general enough to capture a variety of interesting applications.
a constant-factor approximation guarantee in poly-logarithmic adaptive rounds for submodular maximiza-

Our contribution. Focusing on sampling techniques, we study the problem of maximizing a non-

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

its quality. We assume that oracle functions are sub-

Definition 1 (Submodularity). Given a finite set \( V \), we call a set function \( f : 2^V \to \mathbb{R}_{\geq 0} \) 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.
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 \( S \subseteq \Omega \), 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 \( f : 2^V \to \mathbb{R} \) 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 \( \text{OPT} \) 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 is used consistently throughout the paper.
Algorithm 3: REP-SAMPLING\( (f,V,\mathcal{I},\varepsilon,\varphi_1,\varphi_2,m) \)
\begin{enumerate}
\item \( \lambda \leftarrow \varepsilon (p+1)/m; \)
\item \textbf{for} \( j \leq m \) \textbf{iterations} \textbf{do}
\item \( \Omega_j \leftarrow \text{RAND-SAMPLING} (f,V,\mathcal{I},\varepsilon,\varphi_1); \)
\item \( \Lambda_j \leftarrow \text{UNIF-SAMPLING} (\Omega_j,\varphi_2); \)
\item \( V \leftarrow V \setminus \Omega_j; \)
\item \textbf{end for}
\item \textbf{return} \( \arg \max_j \{ f(\Omega_j), f(\Lambda_j) \} \)
\end{enumerate}

index \( \eta \) such that \( \eta = \min \{ j : |X_j| \leq (1-\varepsilon)|X| \} \), 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 \( \varphi \), this algorithm samples points of the input set independently, with probability \( \varphi \).

**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 \( \lambda, \varepsilon \) and \( \varphi_1, \varphi_2 \). At each step, the REP-SAMPLING calls Algorithm 2 to find a partial solution \( \Omega_j \). Then, Algorithm 3 samples a subset of \( \Omega_j \), where each point is drawn independently with probability \( \varphi_2 \). Afterwards, the REP-SAMPLING removes all points of \( \Omega_j \) 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 \( \Omega^* \) the output of Algorithm 3. Then,
\[
f(\Omega^*) \leq m \left( \frac{1+\varepsilon}{1-\varepsilon} \right) \frac{(p+1)}{m-1} + 2 \right) \mathbb{E} \left[ f(\Omega^*) \right].
\]

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 \( \mathcal{O} \left( \frac{m}{\varepsilon^2} \log \left( \frac{1}{\varepsilon} \right) \log r \log n \right) \) rounds of adaptivity. Furthermore, Algorithm 3 has query complexity of \( \mathcal{O} \left( \frac{mn}{\varepsilon^2} \log \left( \frac{1}{\varepsilon} \right) \log r \log n \right) \).

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 \( m = 1 + \sqrt{(p+1)/2}, \varphi_1 = 1, \) and \( \varphi_2 = 1/2 \). Denote with \( \Omega^* \) the optimal solution found by Algorithm 3. Then,
\[
f(\Omega^*) \leq \frac{1 - \varepsilon}{(1-\varepsilon)^2} \left( p + 2 \sqrt{2(p+1)} + 5 \right) \mathbb{E} \left[ f(\Omega^*) \right].
\]

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

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 \( \Omega^* \) the output of Algorithm 3. Then,
\[
f(\Omega^*) \leq \frac{(1+\varepsilon)(p+1)^2}{p(1-\varepsilon)^2} \mathbb{E} \left[ f(\Omega^*) \right].
\]

With this parameter choice, Algorithm 3 terminates after \( \mathcal{O} \left( \varepsilon^{-2} \log n \log \left( \frac{1}{\varepsilon} \right) \log r \right) \) rounds of adaptivity, and it requires \( \mathcal{O} \left( \frac{m^2}{\varepsilon^2} \log n \log \left( \frac{1}{\varepsilon} \right) \log r \right) \) 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 $I$ 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 $O\left(\frac{m^{1/2}}{\varepsilon^2} \log \left(\frac{1}{\varepsilon^2}\right) \log r \log n\right)$ rounds of independent calls to the oracle for the $p$-system constraint. Furthermore, the total number of calls to the independence system is $O\left(\frac{mn^{1/2}}{\varepsilon^2} \log \left(\frac{1}{\varepsilon^2}\right) \log r \log n\right)$.

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 $O(\sqrt{n})$ 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 $O(\sqrt{n})$ 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 $\tilde{O}(n^{1/3})$ 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 $I$, the rank of a set $S$ is the maximum cardinality of its intersection with a maximum independent set in $I$. Given access to an oracle that returns the rank of a set in $I$, it is possible to design an algorithm that finds a maximum independent set of a $p$-system in $O(\log n^3)$ 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 REP-SAMPLING as described 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 samples points 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
We select a representative summary by maximizing

\[
\text{We then learn the parameters}
\]

\[
\text{we refer the reader to Kulesza and Taskar [2012].}
\]

\[
\text{frames showing face } i \text{. In our experiments, the parameters } k_i \text{ are always set to a fixed constant for all videos. Hence, the only variable that affects } p \text{ is the total number of distinct faces in each movie.}
\]

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

\[
\text{The results are displayed in Figure 1, where we describe the parameter choice for each algorithm. For each nondeterministic algorithm, results are the sample mean of 100 independent runs. We observe that, for different parameter choice, our algorithm outperforms FANTOM and the FASTSGS, and it has better adaptivity than the greedy algorithms. The solution quality for the SAMPLEGREEDY and REP-SAMPLING, with parameters as in Lemma 3 is worse on these instances.}
\]

\[
\text{9 Bayesian D-Optimality}
\]

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 \( \theta \) 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.
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 \( \mathcal{H}(S) = \frac{1}{2} \ln |\Sigma| + \frac{1}{2} \ln |\det \Sigma(S)| \), with \( S \) a subset of stations, and \( \Sigma \) 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.
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).

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


Adaptive Sampling for Fast Constrained Maximization of Submodular Functions


