A Distributed Algorithm for Partitioned Robust Submodular Maximization

Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher
École Polytechnique Fédérale de Lausanne (EPFL)
{ilija.bogunovic, slobodan.mitrovic, jonathan.scarlett, volkan.cevher}@epfl.ch

***Abstract*—In this paper, we consider the problem of maximizing a monotone submodular function subject to a cardinality constraint, with two added twists: The computation is distributed across a number of machines, and we require the solution to be robust against adversarial removals. We provide two versions of a partitioned robust algorithm for this problem, with the difference amounting to whether or not the centralized machine is informed (only in the final stage of the algorithm) which elements will be removed. In both of these cases, we provide a novel constant-factor approximation guarantee with respect to the optimal algorithm. Finally, we validate our algorithms via numerical experiments on real-world data sets in influence maximization and data summarization.**

I. Introduction

Discrete optimization problems arise frequently in machine learning, and are often NP-hard even to approximate. For set functions exhibiting *submodularity*, one can efficiently perform cardinality-constrained maximization with a $\left(1-\frac{1}{e}\right)$-factor approximation guarantee. Applications of submodular function maximization include influence maximization [1], document summarization [2], sensor placement [3], active learning [4] and Bayesian optimization [5], just to name a few.

In this paper, we combine two important aspects of submodular maximization that have recently gained attention: *distributed computation* [6], [7] and *robustness* [8], [9], [10]. Distributed algorithms are of interest in large data sets where the data does not fit on a single machine, or where processing all of the data on a single machine is computationally prohibitive. Robustness is important in applications where some of the returned elements may "fail" (e.g., users refusing to spread the word in influence maximization), and one wishes to ensure that this degrades the objective function as little as possible.

The algorithm proposed in this paper combines our robustness techniques from [10] and the distributed techniques from [6] in a non-trivial manner. We provide a rigorous constant-factor approximation guarantee on the final solution returned, being the first such result for the robust distributed setting. In addition, we experimentally validate the performance of our algorithm on real-world data sets in influence maximization and data summarization.

A. Problem Statement

We proceed by formally specifying the problem statement. Let V be a ground set with cardinality $|V| = n$, and let $f : 2^V \to \mathbb{R}_{\geq 0}$ be a set function defined on V. The function f is said to be *submodular* if for any sets $X \subseteq Y \subseteq V$ and any element $e \in V \setminus Y$, it holds that

$$f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y).$$

We use the following notation to denote the marginal gain in f due to adding the elements of a set Y to the set X:

$$f(Y|X) := f(X \cup Y) - f(X).$$

If Y is a singleton of the form $\{e\}$, we adopt the shorthand $f(e|X)$. We say that f is *monotone* if for any sets $X \subseteq Y \subseteq V$ we have $f(X) \leq f(Y)$, and *normalized* if $f(\emptyset) = 0$.

Our goal is to output a set W of a given cardinality k in a robust manner. Specifically, we assume that there exists a set E of cardinality at most τ whose elements are removed from the output W (if they were included in the first place). We seek to design W in a manner such that $f(W \backslash E)$ is as high as possible, rather than considering $f(W)$ alone.

We give a distributed solution to this problem, in which each element from the ground set V is sent to one of m machines uniformly at random. This results in a random partition of V into $V_1, \dots, V_m$. We then proceed in two stages. In the first stage, each machine $i = 1, \dots, m$ returns a set $S_i \subseteq V_i$ of cardinality k to the central machine. In the second stage, the central machine processes $S_1, \dots, S_m$ to produce the final output W, and we consider two variants:

- *(E is given)* The set E is given to the central machine in the second stage only, and hence W is a function of both $S_1, \dots, S_m$ and E.
- *(E is unknown)* The set E remains unknown, and W can only be a function of $S_1, \dots, S_m$.

The case that E is given might be of interest, for example, when we wish to summarize a set of images, and we can receive feedback from the user on which images they wish to remove from $S_1, \dots, S_m$ before forming the final set W, but cannot re-process the initial large data set.

Our distributed algorithm for both variants is given in Section II, and Section III gives their theoretical guarantees.

B. Related Work

The two most related works in the literature are the distributed submodular maximization framework of [6], and the robust framework of [10]. The technique of [10] built on the recent work [9], which in turn studied the robust formulation originally proposed in [8]. Other works related to the distributed setting include [11], [7], but these are less directly relevant to the present paper.

II. D-PRO ALGORITHM

Our distributed algorithm, which we refer to as D-PRO, makes use of the Partitioned Robust (PRO) algorithm recently proposed in [10], shown in Algorithm 1. Here GREEDY(k, V) denotes the size-k output the greedy algorithm applied to the set V; while alternative subroutines were also considered in [10], in this paper we focus on the greedy subroutine. Note that η is an integer parameter that increases with k in the theoretical guarantees, but can be set to one in practice.

PRO was shown in [10] to provide rigorous robustness guarantees. However, its computational complexity may be prohibitive for large ground sets, or the ground set may even be too large to store on a single machine. The distributed approach provides a powerful means to circumvent these limitations.

The first stage of D-PRO (performed across the m machines) is shown in Algorithm 2. Following [6], each machine takes the random subset $V_i \subseteq V$ and runs the PRO subroutine. Provided that m is sufficiently large, this is considerably more tractable than running PRO on the entire ground set.

The second stage of D-PRO (done on the central machine) is shown in Algorithm 3. We again use the PRO subroutine, this time restricted to the subset $\cup_{i=1}^m S_i$, and denote the output by T. If the set E is unknown then we simply return T, whereas if E is known then we return the best set among $(S_1 \setminus E, \dots, S_m \setminus E, T \setminus E)$.

Algorithm 1 Partitioned Robust Submodular optimization algorithm (PRO)

Input: Set V, parameters k, τ, $\eta \in \mathbb{N}_+$
Output: Set $S \subseteq V$ such that $|S| \leq k$
$S_0, S_1 \leftarrow \emptyset$
for $i \leftarrow 0$ **to** $\lceil \log \tau \rceil$ **do**
  for $j \leftarrow 1$ **to** $\lceil \tau / 2^i \rceil$ **do**
    $B_j \leftarrow$ GREEDY$(2^i \eta,\ V \setminus S_0)$
    $S_0 \leftarrow S_0 \cup B_j$
$S_1 \leftarrow$ GREEDY$(k - |S_0|,\ V \setminus S_0)$
$S \leftarrow S_0 \cup S_1$
return S

Algorithm 2 First stage of the distributed algorithm D-PRO

Input: Set V, parameters k, τ, η, $m \in \mathbb{N}_+$
Partition V randomly into $\{V_i\}_{i=1}^m$
for $i \leftarrow 1$ **to** m **in parallel do**
  $S_i \leftarrow$ PRO(V_i, k, τ, η)
return $(S_1, \dots, S_m)$

Algorithm 3 Second stage of D-PRO on the central machine

Input: Sets $(S_1, \cdots, S_m)$, parameters k, τ, $\eta \in \mathbb{N}_+$
Output: Set $W \subseteq \cup_{i=1}^m S_i$ such that $|W| \leq k$
$T \leftarrow$ PRO$(\cup_{i=1}^m S_i, k, \tau, \eta)$
$Z = \{S_1, \cdots, S_m, T\}$
if E is known **then**
  $W \leftarrow \mathrm{argmax}_{S \in Z} f(S \setminus E)$
else
  $W \leftarrow T$
return W

Remark 2.1: In the case that E is unknown, an alternative approach would be to return the best set among $(S_1, \dots, S_m, T)$ after the *worst-case removal* of τ elements. Such an approach would in fact improve the theoretical guarantees (*cf.*, Section III) for this case. However, finding such a worst-case set is an instance of a constrained submodular minimization algorithm, which is NP hard in general. We thus focus on the simpler tractable algorithm that immediately returns T.

III. THEORETICAL GUARANTEES

In this section, we formally state the robustness guarantees of D-PRO (*cf.*, Algorithms 2 and 3). We note that all of our results are stated and proved only for arbitrary sets E (of cardinality at most τ) that are fixed in advance, and do not depend on the randomness in producing $V_1, \dots, V_m$ (e.g., this would be the case if E was a function of W). However, we have observed in the numerical experiments that our algorithm still works well in such cases (*cf.*, Section IV).

A. Preliminaries

PRO guarantees. Fix $E \subseteq V$ with $|E| \leq \tau$, and let

$$\mathrm{OPT}(k - \tau, V \setminus E) = \underset{S \subseteq V \setminus E, |S| = k - \tau}{\mathrm{argmax}} f(S).$$

With this definition, the main theoretical result from [10] can be restated as follows.[1]

Theorem 3.1: For a given budget k and parameters $2 \leq \tau \leq \frac{k}{5\eta(\log k + 2)}$ and $\eta \geq 4(\log k + 1)$, PRO in Algorithm 1 returns a set S of size k such that

$$f(S \setminus E) \geq \alpha f(OPT(k - \tau, V \setminus E)),$$

where $E \subseteq V$ is any set such that $|E| \leq \tau$, and

$$\alpha = \frac{\frac{\eta}{5\lceil \log \tau \rceil + \eta}\left(1 - e^{-\frac{k - |S_0|}{k - \tau}}\right)}{1 + \frac{\eta}{5\lceil \log \tau \rceil + \eta}\left(1 - e^{-\frac{k - |S_0|}{k - \tau}}\right)}. \tag{1}$$

In addition, if $\tau = o\left(\frac{k}{\eta \log k}\right)$ and $\eta \geq \log^2 k$, then we have the following as $k \to \infty$:

$$f(S \setminus E) \geq (0.387 + o(1)) f(OPT(k - \tau, V \setminus E)).$$

Lovász extension. The Lovász extension $f^L : [0,1]^V \to \mathbb{R}_{\geq 0}$ of f is given by

$$f^L(x) = \mathbb{E}_{\lambda \in \mathcal{U}(0,1)}[f(\{i \mid x_i \geq \lambda\})],$$

where $\mathcal{U}(0,1)$ denotes the uniform distribution on $[0,1]$. Due to the submodularity of f, the Lovász extension $f^L(x)$ satisfies the following two properties [6]:

1) f^L is convex, and $f^L(c\mathbf{x}) \geq c f^L(\mathbf{x})$ for $c \in [0,1]$;
2) For all $S \subseteq V$, $f^L(\mathbb{1}_S) = f(S)$, where $\mathbb{1}_S$ is a vector of size $|V|$ equaling one at all entries i such that $v_i \in S$, and zero at all other entries.

[1]This result is slightly stronger than that stated in [10], but the proof therein can be viewed as first deriving this stronger result and then weakening it.

B. Auxiliary lemmas

Let $E \subseteq V$ be any fixed set such that $|E| \leq \tau$, and let $\mathrm{OPT}(k-\tau, V \setminus E)$ denote the optimal set of size $k-\tau$ on the ground set $V \setminus E$. Recall that each item is sent to a single machine i chosen uniformly from $\{1, \dots, m\}$, and that V_i denotes the resulting random subset of V for machine i. We define p_v to be the probability that an item $v \in \mathrm{OPT}(k-\tau, V \setminus E)$ is selected by PRO if it is added to the set V_i (by symmetry, i can be any value in $\{1, \dots, m\}$ here):

$$p_v = \Pr\big(v \in \mathrm{PRO}(V_i \cup \{v\})\big). \tag{2}$$

Moreover, let $\mathbf{p}$ denote a vector of size $|V|$ equaling p_v and the entries corresponding to $v \in \mathrm{OPT}(k-\tau, V \setminus E)$, and zero at all other entries.

The following two lemmas are key to our analysis, lower bounding the expectations $\mathbb{E}[f(S_i \setminus E)]$ (for every $i \in \{1, \dots, m\}$) and $\mathbb{E}[f(T \setminus E)]$ respectively.

Lemma 3.2: $\mathbb{E}[f(S_i \setminus E)] \geq \alpha f^L(\mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)} - \mathbf{p})$.

Proof: Let

$$O_i = \{v \in \mathrm{OPT}(k-\eta, V \setminus E) : v \notin \mathrm{PRO}(V_i \cup \{v\})\},$$

be the set of omitted items for machine i. We claim that

$$\mathrm{PRO}(V_i) = \mathrm{PRO}(V_i \cup O_i), \tag{3}$$

and that more generally, if several elements remain unchosen by PRO when individually added to the ground set, then adding all such elements to the ground set does not affect the output. This property is inherited directly from the analogous property of GREEDY, as used in [6]. We remark that it may not hold for variations of PRO that use a non-greedy subroutine [10].

Using the above, we write

$$\begin{aligned} f(S_i \setminus E) &\geq \alpha f(\mathrm{OPT}(k-\tau, (V_i \cup O_i) \setminus E)) \\ &\geq \alpha f(\mathrm{OPT}(k-\tau, O_i \setminus E)) \\ &= \alpha f(\mathrm{OPT}(k-\tau, O_i)) \\ &= \alpha f(O_i), \end{aligned} \tag{4}$$

where the first line follows from Theorem 3.1 and (3), the second line uses monotonicity, the third line follows since O_i contains no items from E by definition, and the last line from the fact that $|O_i| \leq k - \tau$.

For each element $v \in \mathrm{OPT}(k-\tau, V \setminus E)$, we have $\Pr[v \in O_i] = 1 - p_v$, and hence

$$\mathbb{E}[\mathbb{1}_{O_i}] = \mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)} - \mathbf{p}. \tag{5}$$

Finally, we weaken (4) as follows:

$$\begin{aligned} \mathbb{E}[f(S_i \setminus E)] &\geq \alpha\, \mathbb{E}[f(O_i)] \\ &= \alpha\, \mathbb{E}[f^L(\mathbb{1}_{O_i})] \\ &\geq \alpha f^L(\mathbb{E}[\mathbb{1}_{O_i}]) \\ &= \alpha f^L(\mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)} - \mathbf{p}), \end{aligned} \tag{6}$$

where the second line follows from the second property of Lovász extension, the third line follows from Jensen's inequality, and the last line follows from (5). ∎

Lemma 3.3: $\mathbb{E}\left[f\left(T \setminus E\right)\right] \geq \alpha f^L(\mathbf{p})$.

Proof: Let $D = \cup_{i=1}^m S_i$. From Theorem 3.1, we have

$$\begin{aligned} f\big(T \setminus E\big) &\geq \alpha f\big(\,\mathrm{OPT}(k-\tau, D \setminus E)\big) \\ &\geq \alpha f\big(D \cap \mathrm{OPT}(k-\tau, V \setminus E)\big), \end{aligned} \tag{7}$$

where the last line follows since $D \cap \mathrm{OPT}(k-\tau, V \setminus E)$ is a subset of $D \backslash E$ of size at most $k - \tau$.

Next, we have for any $v \in \mathrm{OPT}(k-\tau, V \setminus E)$ that $\Pr\left(v \in D \cap \mathrm{OPT}(k-\tau, V \setminus E)\right) = p_v$, and consequently

$$\mathbb{E}\left[\mathbb{1}_{D\cap \mathrm{OPT}(k-\tau, V\setminus E)}\right] = \mathbf{p}. \tag{8}$$

Finally, continuing with (7), we obtain

$$\begin{aligned} &\mathbb{E}\left[f\left(T \setminus E\right)\right] \\ &\quad\geq \alpha\, \mathbb{E}\left[f\big(D \cap \mathrm{OPT}(k-\tau, V \setminus E)\big)\right] \\ &\quad= \alpha\, \mathbb{E}\left[f^L\left(\mathbb{1}_{D\cap \mathrm{OPT}(k-\tau, V\setminus E)}\right)\right] \\ &\quad\geq \alpha f^L\left(\mathbb{E}\left[\mathbb{1}_{D\cap \mathrm{OPT}(k-\tau, V\setminus E)}\right]\right) \\ &\quad= \alpha f^L(\mathbf{p}), \end{aligned} \tag{9}$$

where he third line follows from the second property of the Lovász extension, the fourth line follows from the convexity of f^L, and the last line follows from (8). ∎

C. Statement of main result

With the above preliminaries and lemmas in place, we are now in a position to state our main result.

Theorem 3.4: For any monotone submodular function f with $f(\emptyset) = 0$, and any parameters k, $\eta \geq 4(\log k + 1)$ and $2 \leq \tau \leq \frac{k}{5\eta(\log k+2)}$, the algorithm D-PRO returns a set W satisfying the following:

(i) In the case that E is given, we have

$$\mathbb{E}[f(W \setminus E)] \geq \frac{1}{2}\alpha f\left(\mathrm{OPT}(k-\tau, V \setminus E)\right);$$

(ii) In the case E is unknown, we have

$$\mathbb{E}[f(W \setminus E)] \geq \frac{\alpha^2\big(1 - \frac{\tau}{k}\big)}{2} f\left(\mathrm{OPT}(k-\tau, V \setminus E)\right),$$

where α is given in (1). In addition, if $\tau \in o\big(\frac{k}{\eta \log k}\big)$ and $\eta \geq \log^2 k$, then as $k \to \infty$, the asymptotic approximation factor becomes $\frac{\alpha_\infty}{2}$ and $\frac{\alpha_\infty^2}{2}$ for the first and second case respectively, where $\alpha_\infty \geq 0.387$.

Proof: **Case 1:** From Lemmas 3.2 and 3.3, we have $\mathbb{E}[f(W \setminus E)] \geq \alpha f^L(\mathbf{p})$ and $\mathbb{E}[f(W \setminus E)] \geq \alpha f^L\left(\mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)} - \mathbf{p}\right)$. Combining these, we obtain

$$\begin{aligned} \mathbb{E}[f(W \setminus E)] &\geq \frac{1}{2}\alpha\left(f^L(\mathbf{p}) + f^L\left(\mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)} - \mathbf{p}\right)\right) \\ &\geq \frac{1}{2}\alpha f^L\left(\mathbb{1}_{\mathrm{OPT}(k-\tau, V\setminus E)}\right) \\ &= \frac{1}{2}\alpha f\left(\mathrm{OPT}(k-\tau, V \setminus E)\right), \end{aligned}$$

where the second line follows from Jensen's inequality, and the third line uses the second Lovász property.

Case 2: From Lemmas 3.2 and 3.3, we have $\mathbb{E}[f(T \setminus E)] \geq \alpha f^L(\mathbf{p})$ and $\mathbb{E}[f(S_i \setminus E)] \geq \alpha f^L\left(\mathbb{1}_{\mathrm{OPT}(k-\eta, V\setminus E)} - \mathbf{p}\right)$. We

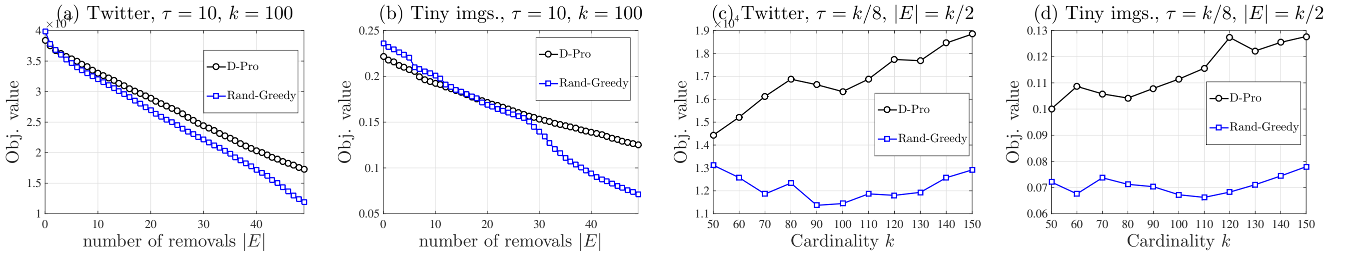

Fig. 1: Comparisons of the D-PRO algorithm, and the existing distributed algorithm of [6] designed for the non-robust setting.

also have $\mathbb{E}[f(T \setminus E)] \geq \alpha\big(1 - \frac{\tau}{k}\big)\, \mathbb{E}[f(S_i \setminus E)]$, which follows from Theorem 3.1 and the fact that the optimal size-$(k-\tau)$ set has an objective function within a factor $\frac{k-\tau}{k}$ of the optimal size-k set by submodularity and monotonicity (note that $|S_i \setminus E|$ can take any value in $\{k-\tau, \dots, k\}$).

Combining these observations, we have

$$\begin{aligned}
&\mathbb{E}[f(T \setminus E)] \\
&\quad \geq \frac{\alpha\big(1 - \frac{\tau}{k}\big)}{2}\left(f^L(\mathbf{p}) + \alpha f^L\left(\mathbb{1}_{\mathrm{OPT}(k-\tau, V \setminus E)} - \mathbf{p}\right)\right) \\
&\quad = \frac{\alpha^2\big(1 - \frac{\tau}{k}\big)}{2} f\left(\mathrm{OPT}(k-\tau, V \setminus E)\right)
\end{aligned}$$

by analogous arguments to Case 1. ■

IV. Experiments

In this section, we validate our algorithm in two contexts consisting of a non-negative, monotone, and submodular objective function. We focus on the case of unknown E. We first introduce the problems, objective functions, and data sets.

Dominating set problem. The dominating set problem arises as a simple example of influence maximization, where a graph specifies influence relations among a set nodes, and the goal is to select a subset of nodes such that the total influence is as high as possible. More formally, we fix a graph $G = (V, F)$, where V denotes the set of nodes and F denotes the set of edges. Letting $\mathcal{N}(S)$ denote the neighbors of the nodes in $S \subset V$, the goal is to find a set of nodes S of size k that maximizes the following objective function:

$$\min_{|E| \leq \tau} |(S \setminus E) \cup \mathcal{N}(S \setminus E)|. \tag{10}$$

We consider the EGO-TWITTER dataset that consists of 973 social circles from Twitter, used previously in [12]. These form a directed graph with 81306 nodes and 1768149 edges.

Exemplar based clustering. Exemplar based clustering provides a powerful means for finding a representative subset from a much larger set (e.g., image summarization). The goal is to find a set of items S of size k that maximize the following:

$$\min_{|E| \leq \tau} f(\{e_0\}) - f((S \setminus E) \cup \{e_0\}). \tag{11}$$

Here we use e_0 to denote a reference element, $f(S) = \frac{1}{|V|} \sum_{v \in V} \min_{s \in S} d(s, v)$ is the *k-medoid* loss function, and $d(s, v)$ measures the dissimilarity between images s and v.

For this problem, we consider TINY10K dataset of size $10k$ [13], consisting of images each represented as a 3072-dimensional vector. We use $d(s, v) = \|s - v\|^2$, and let the reference element e_0 be the zero vector. Moreover, we shift and scale each vector to have zero mean and unit variance.

Experimental setup. In the following, the distributed algorithm of [6] (which is not targeted at achieving robustness) is referred to as RAND-GREEDI. We compare the robustness of the solution produced by our algorithm against the one produced by RAND-GREEDI.

Given a solution set S of size k, we measure the performance in terms of the minimum objective value upon the removal of τ elements, i.e. $f(S \setminus E)$ with $|E| = \tau$. Unfortunately, for a given solution set S, finding the worst-case set E (i.e., the optimal adversary) is an instance of the submodular minimization problem with a cardinality constraint, which is known to be NP-hard with polynomial approximation factors [14]. Hence, implementing the optimal adversary appears to be computationally challenging.

However, as was observed in [10], the *greedy adversary*, which iteratively removes elements to reduce the objective value as much as possible, behaves in a nearly identical manner to the optimal adversary for the data sets under consideration. Hence, we adopt the greedy adversary in our experiments, as it is easy to implement efficiently.

In Figures 1 (a)–(b), we show how the objective value changes as we increase the number of removed elements, for both the EGO-TWITTER and TINY10K datasets. Here, k is fixed and set to 100. Our algorithm is run with $\tau = 10$, i.e. as if 10 elements were to be removed. However, in both experiments, we also consider the performance when an even larger number of elements is removed – up to 50. We can observe that for a small number of removals, RAND-GREEDI outperforms our algorithm; this is because D-PRO stores "redundant" items in order to ensure robustness. On the other hand, as the number of removals increases, D-PRO outperforms RAND-GREEDI.

In Figure 1 (c)–(d), we vary the size of the returned set k while the number of removed elements is set to $\tau = k/2$, for both the EGO-TWITTER and TINY10K datasets. In our algorithm, we set $\tau = k/8$, so that we under-estimate the true number of removals. In both experiments, D-PRO dominates RAND-GREEDI for all values of k considered, often with a significant gap between the two.

Acknowledgment

This work was supported in part by the European Commission under Grant ERC Future Proof, SNF 200021-146750 and SNF CRSII2-147633, and the 'EPFL Fellows' program (Horizon2020 grant 665667).

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, 2003.

[2] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Assoc. for Comp. Ling.: Human Language Technologies-Volume 1*, 2011.

[3] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *Conf. Art. Intell. (AAAI)*, 2007.

[4] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, no. 19, p. 8, 2012.

[5] I. Bogunovic, J. Scarlett, A. Krause, and V. Cevher, "Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation," in *Adv. Neur. Inf. Proc. Sys. (NIPS)*, 2016, pp. 1507–1515.

[6] R. Barbosa, A. Ene, H. L. Nguyen, and J. Ward, "The power of randomization: Distributed submodular maximization on massive datasets," in *Int. Conf. Mach. Learn. (ICML)*, 2015.

[7] M. Lucic, O. Bachem, M. Zadimoghaddam, and A. Krause, "Horizontally scalable submodular maximization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016.

[8] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, no. Dec, pp. 2761–2801, 2008.

[9] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," in *Int. Conf. on Integer Programming and Combinatorial Opt. (IPCO)*. Springer, 2016.

[10] I. Bogunovic, S. Mitrović, J. Scarlett, and V. Cevher, "Robust submodular maximization: A non-uniform partitioning approach," in *Int. Conf. Mach. Learn. (ICML)*, 2017.

[11] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," in *Adv. Neur. Inf. Proc. Sys. (NIPS)*, 2013, pp. 2049–2057.

[12] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discov. Data*, 2014.

[13] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Patt. Ana. Mach. Intel.*, vol. 30, no. 11, pp. 1958–1970, 2008.

[14] Z. Svitkina and L. Fleischer, "Submodular approximation: Sampling-based algorithms and lower bounds," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1715–1737, 2011.