Instance Specific Approximations for Submodular Maximization

Eric Balkanski [1] Sharon Qian [2] Yaron Singer [2]

Abstract

For many optimization problems in machine learning, finding an optimal solution is computationally intractable and we seek algorithms that perform well in practice. Since computational intractability often results from pathological instances, we look for methods to benchmark algorithms' performance against optimal solutions on real-world instances. The main challenge is that an optimal solution cannot be efficiently computed for intractable problems, and we therefore often do not know how far a solution is from being optimal. A major question is therefore how to measure the performance of an algorithm in comparison to an optimal solution on instances we encounter in practice.

In this paper, we address this question in the context of submodular optimization problems. For the canonical problem of submodular maximization under a cardinality constraint, it is intractable to compute a solution that is better than a $1 - 1/e \approx 0.63$ fraction of the optimum. Algorithms like the celebrated greedy algorithm are guaranteed to achieve this $1 - 1/e$ bound on any instance, and are used in practice.

Our main contribution is not a new algorithm for submodular maximization but an analytical method that measures how close an algorithm for submodular maximization is to optimal on a given problem instance. We use this method to show that on a wide variety of real-world datasets and objectives, the approximation of the solution found by greedy goes well beyond $1 - 1/e$ and is often at least 0.95. We develop this method using a novel technique that lower bounds the objective of a dual minimization problem to obtain an upper bound on the value of an optimal solution to the primal maximization problem.

1. Introduction

A central challenge in machine learning is that many of the optimization problems we deal with are computationally intractable. For problems like clustering, sparse recovery, and maximum likelihood estimation for example, finding an optimal solution is computationally intractable and we seek heuristics that perform well in practice. Computational intractability implies that under worst case analysis any efficient algorithm is suboptimal; however, worst-case approximation guarantees are often due to pathological instances that are not representative of instances we encounter in practice. Thus, we would like to be assured that the algorithms we use, despite poor performance on pathological instances, perform provably well on real-world instances.

In order to evaluate the performance of an algorithm on real-world instances, we would like to measure its performance in comparison to an optimal solution. The main challenge however, is that we cannot evaluate the performance of an algorithm against an optimal solution since finding an optimal solution for a computationally intractable problem is, by definition, intractable. Thus, we often do not know how far an algorithm solution is from optimal, and whether there is a substantially better algorithm. Therefore, for computationally intractable problems, our main challenge is not necessarily how to design better algorithms, but rather how to measure the performance of an algorithm in comparison to a theoretically optimal solution on real-world instances.

How do we measure the performance of an algorithm on specific instances for problems that are intractable?

In this paper, we develop a method to measure how close to optimal the performance of an algorithm is on specific instances for the broad class of submodular maximization problems. Since many objectives that we aim to optimize, such as coverage, diversity, entropy, and graph cuts are submodular, submodular maximization algorithms are heavily employed in applications such as speech and document summarization (Lin & Bilmes, 2011), recommender systems (Mirzasoleiman et al., 2016b), feature selection (Das & Kempe, 2011), sensor placement (Guestrin et al., 2005), and network analysis (Kempe et al., 2003). Submodular maximization provides an ideal framework to address this problem since it is intractable to compute a solution that

[1] Columbia University, New York, NY, USA [2] Harvard University, Cambridge, MA, USA. Correspondence to: Eric Balkanski <eb3224@columbia.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

is better than a $1 - 1/e$ approximation for the canonical problem of maximizing a monotone submodular function under a cardinality constraint (Nemhauser & Wolsey, 1978). In addition, multiple algorithms are known to enjoy constant factor approximation guarantees, such as the greedy algorithm that achieves this $1 - 1/e$ approximation on any instance (Nemhauser et al., 1978).

Our contribution. We develop a novel and efficient method, called DUAL, to measure how close to optimal the performance of an algorithm is on an instance of maximizing a monotone submodular function under a cardinality constraint. These instance specific approximations are obtained by upper bounding the optimal value of an instance. We use this method to show that greedy, as well as other submodular maximization algorithms, perform significantly better than $1 - 1/e$ in practice. On a wide variety of large real-world datasets and objectives, we find that approximation of the solution found by greedy almost always exceeds 0.85 and often exceeds 0.95, a 50 percent improvement over $1 - 1/e \approx 0.63$. Using this method, we also find that greedy and another algorithm with the same $1-1/e$ theoretical guarantee have significantly different approximations in practice. Additionally, we show that DUAL significantly outperforms multiple benchmarks for measuring instance specific approximations.

Technical overview. DUAL derives an approximation obtained by a solution S to an instance of $\max_{|S|\leq k} f(S)$, where $f : 2^N \to \mathbb{R}$ is a monotone submodular function, by upper bounding the optimal value $\texttt{OPT}$. To upper bound $\texttt{OPT}$, we consider a dual problem $g(v) = \min_{S:f(S)\geq v} |S|$ of finding the solution S of minimum size that has value at least v. We then construct a function $\underline{g}(v)$ that lower bounds $g(v)$ and is such that the maximum value v such that $\underline{g}(v) \leq k$ can be found in $O(n \log n)$ running time. By exploiting the fact that $\underline{g}$ lower bounds g, we show that this maximum value v is an upper bound on $\texttt{OPT}$.

The construction of $\underline{g}$ is motivated by coverage functions, a special case of submodular functions, where the goal is to maximize the coverage of a universe U. We show that the dual objective $g : 2^U \to \mathbb{R}$, corresponding to the minimum size of a set $S \subseteq N$ which covers $T \subseteq U$, can be lower bounded by an *additive* function $\ell : 2^U \to \mathbb{R}$, which is close to g in practice. Since ℓ is additive, it can be minimized efficiently to give a lower bound on the dual problem, which implies an upper bound on $\texttt{OPT}$.

Related work. A line of work has investigated different properties of submodular functions that enable improved approximation guarantees for the greedy algorithm. The curvature $c \in [0, 1]$ of a function f measures how close f is to additive (Conforti & Cornuéjols, 1984; Soma & Yoshida, 2017). Submodular sharpness, an analog of sharpness from continuous optimization (Lojasiewicz, 1963), measures the behavior of a function f around the set of optimal solutions (Pokutta et al., 2020). For stability, a function f is perturbation-stable if the optimal solution for maximizing f does not change under small perturbations (Chatziafratis et al., 2017). Finally, drawing from smoothed analysis, Rubinstein and Zhao (2021) show improved approximations using randomized perturbations of the budget k. For curvature, sharpness and stability, the main issue with using these parameterized properties to measure how close to optimal the greedy algorithm is on specific instances is that their parameters cannot be computed efficiently. Since they require brute-force computation, they have only been computed on small instances with at most $n = 20$ elements (Pokutta et al., 2020) and cannot be used on real-world instances. In addition, on these small instances, these three properties and budget-smooth analysis all yield approximations that are not as strong as those obtained by DUAL. The main goal of this line of work is to provide an explanation for greedy's strong performance in practice. In contrast, our paper focuses on the problem of measuring the performance of greedy, and other algorithms, in practice.

Sakaue and Ishihata (2018) develop a method to upper bound the optimal value of a submodular maximization instance, but with a different goal of using this upper bound to accelerate submodular maximization algorithms. This method is used as a benchmark in our experiments. Baeza-Yates, Boldi, and Chierichetti (2015) construct a method to obtain instance-specific approximations for the special case of coverage functions by exploiting the linear program formulation of coverage functions and using the dual formulation of its relaxation as an upper bound of the optimal value. This method does not extend to general submodular problems, which cannot be formulated as a polynomial size LP. We instead propose a different dual problem for instance-specific approximations for general submodular objectives.

1.1. Preliminaries

A function $f : 2^N \to \mathbb{R}$ is submodular if $f_S(a) \geq f_T(a)$ for all $S \subseteq T \subseteq N$ and $a \in N \setminus T$, where $f_S(a) = f(S \cup \{a\}) - f(S)$ is the marginal contribution of a to S. It is monotone if $f(S) \leq f(T)$ for all $S \subseteq T \subseteq N$. A function $f : 2^P \to \mathbb{R}$ is a coverage function if there exists a bipartite graph $G = (P, D, E)$ over primal and dual elements $P \cup D$ such that $f(S) = |N_G(S)|$ where $N_G(S) = \cup_{a\in S} N_G(a) \subseteq D$ denotes the neighbors of $S \subseteq P$ in G. We say that set S covers T, or equivalently that T is covered by S, if $T \subseteq N_G(S)$. When G is clear from the context, we write $N(S)$ instead of $N_G(S)$ to denote the neighbors of S in graph G.

Given an instance of the problem $\max_{|S|\leq k} f(S)$ and a solution $S^\star$ to this problem, we aim to compute an approximation for $S^\star$, i.e. a lower bound on $f(S^\star)/\max_{|S|\leq k} f(S)$ or, equivalently, an upper bound on $\max_{|S|\leq k} f(S)$.

2. Instance Specific Approximations for Max-Coverage

In this section, we present a method that measures how close to optimal a solution is for maximum coverage problems. As a warm-up, we focus on this special case to motivate and provide intuition for the method for submodular functions presented in Section 3. We emphasize that our main contribution is for general submodular maximization problems, which, unlike coverage problems, cannot be formulated as a polynomial size linear program.

In Section 2.1, we introduce the problem of minimum cover under a cardinality constraint, which is a generalization of the classical set cover problem. We show that a lower bound on this minimum cover problem implies an upper bound on the optimal value OPT for the max-coverage problem. In Section 2.2, we present an additive lower bound on the dual problem, which is used to efficiently compute a lower bound of the optimal value to the dual problem. Then, in Section 2.3, we develop a more sophisticated lower bound on the dual problem. Due to space constraints, we defer missing proofs to Appendix A.

2.1. The dual problem

We introduce the minimum cover under a cardinality constraint problem. Recall that given a bipartite graph G over nodes P and nodes D, the problem of maximum coverage under a cardinality constraint problem is to find the k elements $S \subseteq P$ that maximize the number of elements $N_G(S) \subseteq D$ covered by S. In contrast, the minimum cover under a cardinality constraint problem is to find the v elements $T \subseteq D$ that minimize the number of elements $S \subseteq P$ needed to cover $T \subseteq N_G(S)$.

Definition 1. *The minimum cover under a cardinality constraint problem is defined as* $\min_{T\subseteq D:|T|\geq v} g(T)$*, where* $g(T) = \min_{S\subseteq P: T\subseteq N(S)} |S|$ *is the size of the minimum cover of* T *and where* $v \in [|D|]$ *is the cardinality constraint.*

This problem is a generalization of the classical set cover problem, which finds the minimum number of elements S to cover *all* elements D. We obtain the following duality property: a lower bound on minimum cover implies an upper bound on maximum coverage, and vice-versa.

Lemma 1. *Let* $f : 2^P \to \mathbb{N}$ *be a coverage function defined over a biparite graph between elements* P *and* D*. For any* $k \in [|P|]$ *and* $v \in [|D|]$*,* $\max_{S\subseteq P:|S|\leq k} f(S) < v$ *if and only if* $\min_{T\subseteq D:|T|\geq v} g(T) > k$ *where* $g : 2^D \to \mathbb{N}$ *is the size of the minimum cover of* T *as defined in Definition 1.*

We refer to $\max_{S\subseteq P:|S|\leq k} f(S)$ and $\min_{T\subseteq D:|T|\geq v} g(T)$ as the primal and dual problems. We also refer to P and D as the primal and dual elements.

2.2. Warm-up: Approximations via an additive lower bound on the dual

This dual problem admits an additive[1] lower bound that is, as we will show empirically in Section 5, close to the dual objective in practice. This is in contrast to the primal maximum coverage problem, which is far from additive on real instances. We define the individual value v_b of each dual element b as $v_b = \min_{a\in N(b)} \frac{1}{|N(a)|}$. The additive function $\ell : 2^D \to \mathbb{R}$ is defined as follows:

$$\ell(T) = \sum_{b\in T} v_b = \sum_{b\in T} \min_{a\in N(b)} \frac{1}{|N(a)|}.$$

Note that if $b \in D$ is covered by primal element $a \in P$, then a covers at most $1/v_b$ dual elements. In other words, $1/v_b$ is an upper bound on the value obtained from an element a that covers b.

We use this additive lower bound $\ell(\cdot)$ on the dual objective to design a method that returns an upper bound on the optimal value for the primal problem. Method 1 first orders the dual elements b by increasing value v_b. It then finds the prefix $\{b_1, \ldots, b_{i^\star}\}$ of this ordering where $i^\star$ is the minimum size i such that $\ell(\{b_1, \ldots, b_i\}) = \sum_{j=1}^{i} v_{b_j} > k$, and then returns $i^\star$. In other words, it finds the smallest size i such that $\ell(T) > k$ for all sets T of size i.

Method 1 Linear bound on dual objective for coverage

```
input bipartite graph $G = (P, D, E)$, constraint $k$
  $v_b \leftarrow \min_{a\in N(b)} \frac{1}{|N(a)|}$, for each $b \in D$
  $(b_1, \ldots, b_{|D|}) \leftarrow$ elements $D$ ordered by increasing $v_{b_i}$
  $i^\star \leftarrow \min\{i : \sum_{j=1}^{i} v_{b_j} > k\}$
  return $i^\star$
```

The analysis. We first show that $\ell(\cdot)$ is a lower bound on the dual objective $g(\cdot)$ (Lemma 2). We then show that $\{b_1, \ldots, b_i\}$ minimizes ℓ over all sets of size at least i (Lemma 3). Together, these imply that there are no sets of primal elements of size k which cover $i^\star$ dual elements and we obtain $i^\star > \max_{S\subseteq P:|S|\leq k} f(S)$ (Theorem 1).

Lemma 2. *For any coverage function defined by a bipartite graph* G *between* P *and* D*, we have that for any set* $T \subseteq D$*,* $\ell(T) \leq g(T) = \min_{S\subseteq P:T\subseteq N(S)} |S|$.

[1] A function g is additive if there exist values $v_1, \ldots, v_m$ such that $g(T) = \sum_{b\in T} v_b$.

Proof. For any $S \subseteq P$ such that $T \subseteq N(S)$, we have

$$\begin{aligned} |S| &= \sum_{a \in S} \sum_{b \in N(a)} \frac{1}{|N(a)|} \\ &\geq \sum_{a \in S} \sum_{b \in N(a)} \min_{a' \in N(b)} \frac{1}{|N(a')|} \\ &\geq \sum_{b \in T} \min_{a \in N(b)} \frac{1}{|N(a)|} = \ell(T). \qquad \square \end{aligned}$$

Lemma 3. *Consider the ordering of dual elements D by increasing singleton values, i.e., $(b_1, \dots, b_{|D|})$ where $\ell(b_i) \leq \ell(b_j)$ for all $i < j$, then, for all $v \leq |D|$, $\ell(D_v) = \min_{T \subseteq D : |T| \geq v} \ell(T)$ where $D_v = \{b_1, \dots, b_v\}$.*

Proof. Since ℓ is an additive function, the set T of size at least v with minimum value is the set consisting of the v dual elements of minimum singleton value, which is D_v. □

We are now ready to formally prove that Method 1 returns an upper bound on the optimal value to the primal problem.

Theorem 1. *For any k, let $i^\star$ be the value returned by Method 1, then $i^\star > \max_{S \subseteq P : |S| \leq k} f(S)$.*

Proof. By definition of $i^\star$, Lemma 3 and Lemma 2,

$$k < \ell(D_{i^\star}) = \min_{T \subseteq D : |T| \geq i^\star} \ell(T) \leq \min_{T \subseteq D : |T| \geq i^\star} \min_{\substack{S \subseteq P: \\ T \subseteq N(S)}} |S|.$$

By Lemma 1, we get $i^\star > \max_{S \subseteq P : |S| \leq k} f(S)$. □

2.3. Improved method for coverage functions

We improve Method 1 by constructing a lower bound $\underline{g}(\cdot)$ on the dual objective $g(\cdot)$ that is tighter than $\ell(\cdot)$. The function $\underline{g}(T)$ is obtained by partitioning the collection of dual elements T into j parts $\cup_{i=1}^{j} P_i = T$. We define the weight $w(P_i)$ of part P_i to be

$$w(P_i) = |P_i| \cdot \max_{b \in P_i} v_b = |P_i| \cdot \max_{b \in P_i} \min_{a \in N(b)} \frac{1}{|N(a)|}.$$

We note that if $w(P_i) > 1$, then dual elements P_i cannot be covered by a single primal element since there must exist $b \in P_i$ such that $\max_{a \in N(b)} |N(a)| < |P_i|$. This motivates the following definition of a valid partition.

Definition 2. *A partition $P_1, \dots, P_j$ of T is* valid *if $w(P_i) \leq 1$ for all $i \leq j$.*

This definition is such that if a partition $P_1, \dots, P_j$ is *not* valid, then there must exist a part P_i which cannot be covered by a single primal element. We exploit this property to define the following improved lower bound $\underline{g} : 2^D \to \mathbb{R}$ on the dual objective:

$$\underline{g}(T) = \min\{j : \exists \text{ a valid partition } P_1, \dots, P_j \text{ of } T\}.$$

This lower bound $\underline{g}(T)$ is always tighter than the additive lower bound $\ell(\cdot)$.

Proposition 1. *For any coverage function $f : 2^P \to \mathbb{N}$ defined by bipartite graph (P, D, E), we have $\underline{g}(T) \geq \ell(T)$ for all $T \subseteq D$.*

Similarly as with Method 1, we use this lower bound $\underline{g}(\cdot)$ on the dual objective to design a method that returns an upper bound on the optimal value for the primal problem. Method 2, which will be generalized to Method 3 for submodular functions, iteratively constructs a valid partition $\cup_{\kappa=1}^{k} P_\kappa$ of a collection of dual elements T such that $|T|$ is maximized. At iteration κ, part P_κ of the partition is defined as the collection of dual elements $\{b_{i_{\kappa-1}+1}, \dots, b_{i_\kappa}\}$, which are the dual elements with minimum value v_b that are not in the previous parts $P_1, \dots, P_{i-1}$, where i_κ is the maximum index such that part P_i is valid. The method returns value i_k, which is the total size $|\cup_{\kappa=1}^{k} P_\kappa|$ of the partition.

Method 2 Dual bound via partitioning for coverage

```
input bipartite graph $G = (P, D, E)$, constraint $k$
  $v_b \leftarrow \min_{a \in N(b)} \frac{1}{|N(a)|}$, for each $b \in D$
  $(b_1, \dots, b_{|D|}) \leftarrow$ elements $D$ ordered by increasing $v_{b_i}$
  $i_0 \leftarrow 0$
  for $\kappa = 1$ to $k$ do
      $i_\kappa \leftarrow \max\{i : (i - i_{\kappa-1}) \cdot v_{b_i} \leq 1\}$
      $P_\kappa \leftarrow \{b_{i_{\kappa-1}+1}, \dots, b_{i_\kappa}\}$
  return $i_k$
```

Since Method 2 is a special case of Method 3, the analysis of Method 3 in the next section also applies to Method 2.

3. Instance Specific Approximations for Submodular Maximization

In this section, we present our main method, DUAL, which generalizes Method 2 to submodular functions. For coverage functions, the value achieved by a solution S corresponds to the number of dual elements covered by S and the optimal value can be upper bounded by analyzing dual elements. However, for general submodular functions, there are no dual elements corresponding to the solution value. We first introduce a similar dual minimization problem as for coverage, but defined over values $v \in \mathbb{R}$ instead of dual elements $T \subseteq D$. We then construct a lower bound on the dual objective and use it to design a method that upper bounds OPT in $O(n \log n)$ running time. Missing proofs are in Appendix B.1.

We introduce the minimum submodular cover under a cardinality constraint problem, where the goal is to find the smallest set S of value at least v:

$$g(v) = \min_{S \subseteq N : f(S) \geq v} |S|.$$

This problem is a generalization of the submodular cover problem (Wolsey, 1982), which is to find the smallest set S of value at least $f(N)$. We note that, unlike the dual objective for coverage functions, there are no dual elements.

Next, we define a function $\underline{g}(v)$ which lower bounds this dual objective. Similarly as for coverage functions, we consider partitions of the dual space and we define a collection of valid partitions that is used to then define $\underline{g}(v)$. We assume that the ground set of elements $N = \{a_1, \dots, a_n\}$ is indexed by decreasing singleton value, i.e., $f(a_i) \geq f(a_j)$ for all $i < j$. We define $A_i := \{a_1, \dots, a_i\}$.

Definition 3. *Values $v_1, \dots, v_k \in \mathbb{R}$ form a valid partitioning of $v \in \mathbb{R}$ if $\sum_{j \in [k]} v_j = v$ and there exists a witness $W \subseteq N$ such that, for all $j \in [k]$, $f(a_{i_j}) \geq v_j$ where*

$$i_j = \min \left\{ i : a_i \in W, f(W \cap A_i) \geq \sum_{\ell=1}^{j} v_\ell \right\}.$$

As we will show in Lemma 4, for any solution set S, $v_1 = f_{S_0}(S_1), v_2 = f_{S_1}(S_2), \dots, v_{|S|} = f_{S_{|S|-1}}(S)$ forms a valid partioning of $v = f(S)$, where S_j is the set of j elements in S with the largest singleton value. This implies that if a partition $v_1, \dots, v_k$ is not valid, then there is no solution S of size k such that $f_{S_{j-1}}(S_j) \geq v_j$ for all $j \in [k]$. Thus, if a value v does not have a valid partitioning $v_1, \dots, v_k$, then there are no solution S of size k such that $f(S) \geq v$. This implies that the following function $\underline{g} : \mathbb{R} \to \mathbb{N}$ lower bounds the dual objective (Lemma 4):

$$\underline{g}(v) = \min\{k : \exists \text{ a valid partition } v_1, \dots, v_k \text{ of } v\}.$$

We use the lower bound $\underline{g}(v)$ on the dual objective to construct a method that returns an upper bound on $\texttt{OPT}$. Method 3 iteratively constructs a valid partition $v_1, \dots, v_k$ of the dual space such that $\sum_{j \in [k]} v_j$ is maximized. It first orders the elements $a_i \in N$ by decreasing singleton value $f(a_i)$. Then, at each iteration j, it defines value v_j to be the maximum value v such that partition $v_1, \dots, v_j$ is a valid partition with witness $W = A_{i_j}$.

Method 3 Method for submodular functions

```
input function $f$, cardinality constraint $k$
  $(a_1, \dots, a_n) \leftarrow$ elements ordered by decreasing $f(a_i)$
  For $j = 1$ to $k$ do
      $v_j \leftarrow \max\{v : f(a_{i_j}) \geq v$ where
                  $i_j = \min\{i : f(A_i) - \sum_{\ell=1}^{j-1} v_\ell \geq v\}\}$
  return $\sum_{j=1}^{k} v_j$
```

Value v_j at iteration j can be found by iterating through elements indexed by $i \in \{i_{j-1}+1, i_{j-1}+2, \dots\}$ until $i^\star$, where $i^\star$ is the minimum index such that $f(A_{i^\star}) - \sum_{\ell=1}^{j-1} v_\ell \geq f(a_{i^\star})$. Since an element a_i is considered at most once over all iterations, the total running time of the for loop is $O(n)$. Thus, the running time of Method 3 is $O(n \log n)$ due to the sorting of the elements by singleton value. More details on finding value v_j in Appendix B.2.

The analysis. We first show that $\underline{g}(v)$ lower bounds $g(v)$ in Lemma 4. The proof uses the fact that $\underline{g}(v)$ is monotone.

Lemma 4. *For any submodular function f and $v \in \mathbb{R}$, $\underline{g}(v) \leq g(v) = \min_{S \subseteq N : f(S) \geq v} |S|$.*

We now show that the value $\sum_{j=1}^{k} v_j$ returned by the method is the maximum value v such that $\underline{g}(v) \leq k$.

Lemma 5. *Let $\overline{\texttt{OPT}} = \sum_{j=1}^{k} v_j$ be the solution returned by Method 3, then $\underline{g}(\overline{\texttt{OPT}} + \epsilon) > k$ for all $\epsilon > 0$.*

Finally, we combine Lemma 4 and Lemma 5 to show that Method 3 returns an upper bound of $\texttt{OPT}$.

Theorem 2. *For any k, Let $\overline{\texttt{OPT}} = \sum_{j=1}^{k} v_j$ be the solution returned by Method 3, then $\overline{\texttt{OPT}} \geq \max_{S \subseteq N : |S| \leq k} f(S)$.*

Proof. We first show by contrapositive that a lower bound $k < g(v) = \min_{S \subseteq N : f(S) \geq v} |S|$ on the dual problem implies an upper bound $v > \max_{S \subseteq N : |S| \leq k} f(S)$ on the primal problem. Assume $\max_{S \subseteq N : |S| \leq k} f(S) \geq v$. Then, there exists $S^\star$ such that $f(S^\star) \geq v$ and $|S^\star| \leq k$ and we get $g(v) = \min_{S \subseteq N : f(S) \geq v} |S| \leq |S^\star| \leq k$.

Then, by Lemma 5 and Lemma 4, we have

$$k < \underline{g}(\overline{\texttt{OPT}} + \epsilon) \leq g(\overline{\texttt{OPT}} + \epsilon) = \min_{S \subseteq N : f(S) \geq \overline{\texttt{OPT}} + \epsilon} |S|$$

which implies $\overline{\texttt{OPT}} + \epsilon > \max_{|S| \leq k} f(S)$ for all $\epsilon > 0$. □

3.1. DUAL

We describe our main method, DUAL, which uses Method 3 as a subroutine. In the case where a small number of elements have very large singleton values, Method 3 can return an arbitrarily bad approximation to $\texttt{OPT}$ (See example in Appendix B.3). To circumvent this issue, DUAL calls Method 3 on the marginal contribution function $f_S(T) = f(S \cup T) - f(S)$ for each S in a collection of sets $\mathcal{S}$ given as input. If $\{\} \in \mathcal{S}$, then DUAL is no worse than Method 3. If there is $S \in \mathcal{S}$ such that there are no elements with large singleton value according to f_S, then DUAL circumvents the issue previously mentioned. We note that adding more sets to $\mathcal{S}$ can only improve the approximation given by DUAL.

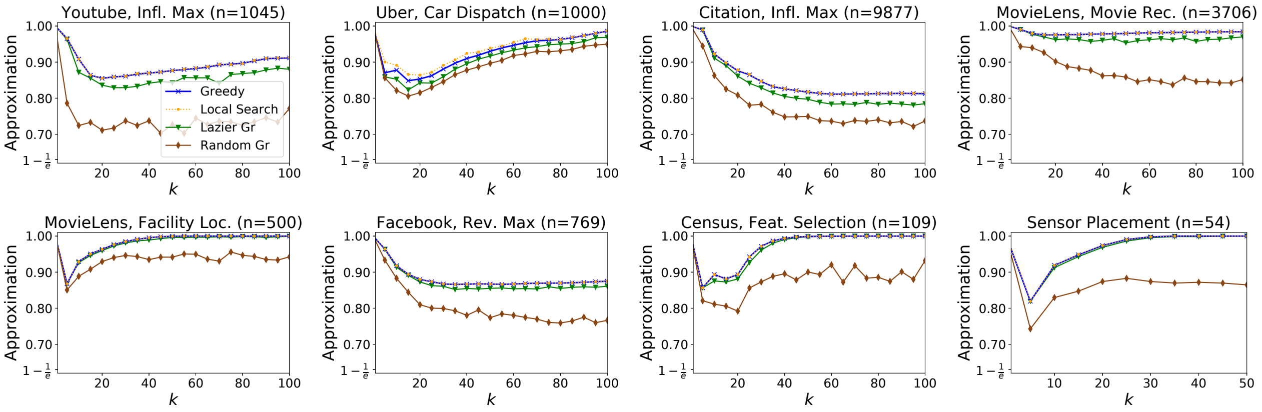

Figure 1. Approximations computed by DUAL for the performance of four different submodular maximization algorithms.

Method 4 DUAL

```
input  function $f$, constraint $k$, collection of sets $\mathcal{S}$
  $\overline{\texttt{OPT}} \leftarrow f(N)$
  for $S$ in $\mathcal{S}$ do
    $\overline{\texttt{OPT}}' \leftarrow$ Method3$(f_S, k)$
    $\overline{\texttt{OPT}} \leftarrow \min(f(S) + \overline{\texttt{OPT}}', \overline{\texttt{OPT}})$
  return $\overline{\texttt{OPT}}$
```

Theorem 3. *For any set collection of sets $\mathcal{S}$, Method 4 returns $\overline{OPT}$ such that $\overline{OPT} \geq OPT$.*

4. Guarantees for DUAL

In this section, we show guarantees on the performance of DUAL, i.e., guarantees on how far the upper bound, $\overline{\texttt{OPT}}$, returned by DUAL is to $\texttt{OPT}$. We note that our analysis holds for the more demanding $f(S_g)$ benchmark, which is the value of the greedy solution S_g. The analysis is deferred to Appendix C.

We first show that, if $S_g \in \mathcal{S}$, then the solution $\overline{\texttt{OPT}}$ returned by DUAL is such that $\overline{\texttt{OPT}} \leq 2 \cdot \texttt{OPT}$. Of course, this 2-approximation can be improved to $e/(e-1)$ with the upper bound $(e/(e-1)) \cdot f(S_g)$. However, for instance specific approximations, this upper bound only gives a $1 - 1/e$ approximation for greedy on each instance while DUAL can give a much stronger approximation.

Proposition 2. *Let S_g be the solution retuned by the greedy algorithm to the problem $\max_{|S| \leq k} f(S)$. Then, if f is a monotone submodular function and $S_g \in \mathcal{S}$,* DUAL *returns $\overline{OPT}$ such that $\overline{OPT} \leq 2f(S_g) \leq 2OPT$.*

We also show that, for functions with curvature c, the solution $\overline{\texttt{OPT}}$ returned by DUAL is such that $\overline{\texttt{OPT}} \leq \frac{1}{1-c}\texttt{OPT}$. Since the curvature parameter c cannot be computed efficiently, the known $(1 - e^{-c})/c$ approximation obtained by greedy (Conforti & Cornuéjols, 1984) is an approximation that cannot be computed efficiently (unless c is known). Unlike $(1 - e^{-c})/c$, $\overline{\texttt{OPT}}$ provides an approximation for functions with curvature that can be efficiently computed. This approximation is guaranteed to improve over $1 - 1/e$ when $c < 1/e$.

Proposition 3. *Let f be a monotone submodular function with curvature c. Then,* DUAL *returns $\overline{OPT}$ such that $\overline{OPT} \leq \frac{1}{1-c} f(S_g) \leq \frac{1}{1-c} OPT$.*

5. Experiments

We utilize DUAL to obtain bounds on the approximation achieved by submodular maximization algorithms in practice. We show that GREEDY and other algorithms find solutions that approximate the optimal solution significantly better than $1 - 1/e$ on a wide variety of real-world datasets and objectives. We also show that DUAL outperforms multiple benchmarks for deriving approximations for the solution found by GREEDY. For all instances, we use $\mathcal{S} = \{S_1, S_2, \ldots, S_{20}\} \cup \{S_{25}, S_{30}, \ldots, S_{50}\}$ as an input to DUAL, where S_i is the greedy solution of size i.

5.1. Approximations for submodular maximization algorithms using DUAL

We begin by evaluating the bounds derived by DUAL on the approximation achieved by four different submodular maximization algorithms. The goal here is not to provide a comprehensive comparison of submodular maximization algorithms, but instead to analyze the approximations computed by DUAL.

Algorithms for submodular maximization. The **GREEDY** algorithm obtains the optimal $1 - 1/e$ approximation (Nemhauser et al., 1978) and is widely considered as the standard algorithm for monotone submodular maximization under a cardinality constraint. **LOCAL SEARCH** obtains a $1/2$ approximation guarantee (Nemhauser et al., 1978) and is another widely used algorithm. **LAZIER-THAN-LAZY**

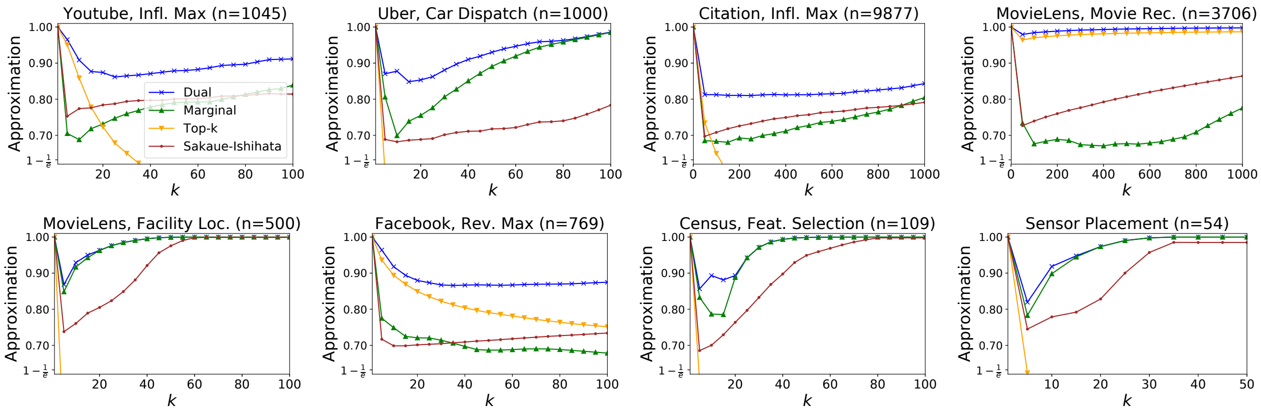

Figure 2. GREEDY approximations computed by DUAL compared to multiple benchmarks on various submodular objectives.

GREEDY, also called sample greedy, improves the running time of greedy by sampling a small subset of elements at each iteration (Mirzasoleiman et al., 2015; Buchbinder et al., 2015). **RANDOM GREEDY** handles submodular functions that are not necessarily monotonic by introducing randomization into the element selection step and obtains a $1 - 1/e$ approximation guarantee for monotone submoduar functions (Buchbinder et al., 2014). We provide further details on these algorithms in Appendix D. For randomized algorithms, we average the results over 5 runs.

Settings. We examine the approximations computed by DUAL for these algorithms on 8 different datasets and objectives. Additional details can be found in Appendix E.1.

- **Influence maximization**: As in Mirzasoleiman, Badanidiyuru, and Karbasi (2016a), we use **Youtube social network** data (Yang & Leskovec, 2015) and sample $n = 1,045$ users from 50 large communities. We select k users by maximizing influence: $f(S) = |N_G(S)|$.

- **Car dispatch**: Our goal is to select the k best locations to deploy drivers to cover the maximum number of pickups. As in Balkanski and Singer (2018), we analyze $n = 1,000$ locations of **Uber pickups** (FiveThirtyEight, 2019) and assign a weight w_i to each neighborhood n_i that is proportional to the number of trips in the neighborhood.

- **Influence maximization**: We use a **citation network** of Physics collaborations (Leskovec et al., 2007) with $n = 9,877$ authors (nodes) and 25,998 co-authorships (edges), and maximize $f(S) = |N_G(S)|$.

- **Movie recommendation**: As in Mirzasoleiman, Badanidiyuru, and Karbasi (2016a), we use the **MovieLens dataset** (Harper & Konstan, 2015) of $n = 3,706$ movies to recommend k movies that have both good overall ratings and are highly rated by the most users.

- **Facility Location**: As in Lindgren, Wu, and Dimakis (2016), we use facility location objective and the movie ranking matrix $[r_{ij}]$ from the **MovieLens dataset** where r_{ij} is user j's ranking on movie i to select k movies to recommend from N using $f(S) = \frac{1}{|N|}\sum_{i \in S} \max_{j \in U} r_{ij}$.

- **Revenue maximization**: We use a revenue maximization objective from Breuer, Balkanski, and Singer (2020) on the **CalTech Facebook Network dataset** (Traud et al., 2012) of 769 Facebook users N.

- **Feature selection**: We use the **Adult Income dataset** (Blake & Merz, 1998) and select a subset of features to predict income level Y. We extract 109 binary features as in Kazemi, Zadimoghaddam, and Karbasi (2018) and use a joint entropy objective to select features.

- **Sensor placement**: As in Krause, Singh, and Guestrin (2008), we use the **Berkeley Intel Lab dataset** which comprises of 54 sensors that collect temperature information. We select k sensors that maximize entropy.

Results. In Figure 1, we see that DUAL computes bounds on the approximations achieved by all four algorithms that are significantly better than $1 - 1/e$. For GREEDY and LOCAL SEARCH, DUAL derives nearly identical approximations that are almost always over 0.85. In many instances, such as movie recommendation, facility location, feature selection, and sensor placement, approximations are over 0.95. The approximations given by DUAL for LAZIER-THAN-LAZY GREEDY are either identical to GREEDY and LOCAL SEARCH, or 0.02 to 0.05 worse. Even though RANDOM GREEDY and GREEDY have the same theoretical guarantee of $1 - 1/e$ for monotone submodular functions, the gap in their approximations on these instances is significant.

In most cases, the approximations obtained by DUAL follow a "Nike-swoosh" shape as a function of constraint k. For $k = 1$, the algorithms are either exactly or near-optimal;

the lowest approximations are obtained for small values of k, and then approximations rebound and slowly increase as k increases. For experiments with $n > 3000$ and the Facebook revenue maximization setting, the values of k (up to 100) are too small to observe this increase.

5.2. DUAL vs benchmarks for GREEDY approximations

For the next set of experiments, we compare DUAL to multiple benchmarks, which also compute approximations for the performance of submodular maximization algorithms. We fix a single algorithm, GREEDY, and compare the approximations found by DUAL and the benchmarks. We consider large instances, as well as small instances with $n \leq 20$ elements, where we can compute the curvature and sharpness benchmarks that require brute-force computation.

5.2.1. EXPERIMENTS ON LARGE INSTANCES

Benchmarks. We consider the following benchmarks that upper bound the optimal value. These benchmarks can be efficiently computed on large instances.

- **Top-k.** For a simple baseline, we upper bound $\texttt{OPT}$ using the k elements, A, with maximum singleton value $f(a)$: $\overline{\texttt{OPT}}_k = \sum_{a \in A} f(a)$.
- **Marginal.** By using the value of GREEDY solutions S_i of size i as well as GREEDY analysis techniques, we derive the following more sophisticated bound:
$$\texttt{OPT}_k \leq \frac{f(S_j) - (1 - 1/k)^{j-i} f(S_i)}{1 - (1 - 1/k)^{j-i}}$$
for all $i < j$ (See Appendix F.1 for proof). We compute the minimum upper bound over all $i < j \leq n$ pairs.
- **Sakaue-Ishihata.** Sakaue and Ishihata (2018) propose a different upper bound of $\texttt{OPT}$ using GREEDY proof techniques and derive the following bound:
$$\texttt{OPT}_k \leq \frac{f(S)}{1 - \prod_{i=1}^{k} \beta_i},$$
where $\beta_i = 1 - \frac{f_{S_{i-1}}(a_i)}{\sum_{a \in A_i} f_{S_{i-1}}(a)}$ and A_i is the set of k elements with maximal singleton values for $f_{S_{i-1}}$.

Results. We compare the approximations of GREEDY found by DUAL to those found by TOPK, MARGINAL and SAKAUE-ISHIHATA. Figure 2 shows that DUAL consistently outperforms or matches the baselines. TOP-K benchmark performs poorly in most cases, which implies that most objectives are not close to additive. MARGINAL and SAKAUE-ISHIHATA are the strongest benchmarks, but are still significantly outperformed by DUAL on most instances. For MovieLens movie recommendation and Facebook revenue maximization, where objectives are close to additive, TOP-K outperforms MARGINAL and SAKAUE-ISHIHATA.

5.2.2. EXPERIMENTS ON SMALL INSTANCES

Benchmarks. We consider the following benchmarks on small datasets that are parameterized properties that yield improved approximation guarantees. Unlike the benchmarks that upper bound $\texttt{OPT}$, these benchmarks identify properties that guarantee a bound on the GREEDY approximation for any function that satisfies the properties. However, computing the parameters of these properties requires brute force computation and is computationally infeasible on large datasets.

- **Curvature.** The curvature $c = 1 - \min_{S,a} \frac{f_S(a)}{f(a)}$ of a function measures how close f is to additive (Conforti & Cornuéjols, 1984). It yields an improved $(1 - e^{-c})/c$ approximation for GREEDY.
- **Soma-Yoshida.** Soma and Yoshida (2017) propose an improved notion of curvature for monotone submodular functions f that can be decomposed as $g + h$, where g is monotone submodular and h is $M^\natural$-concave. The h-curvature $\gamma_h = 1 - \min_{S \subseteq N} \frac{h(S)}{f(S)}$ yields an improved approximation by improving upon the previous curvature parameter.
- **Sharpness.** The property of submodular sharpness was introduced by Pokutta, Singh, and Torrico (2020) as an explanation for the performance of GREEDY in practice. It is the analog of sharpness from continuous optimization (Lojasiewicz, 1963) and assumes that any solution which differs significantly from the optimal solution has a substantially lower value. On small instances, we compute the Dynamic Submodular Sharpness property of the function, which is the sharpness property that yields the best approximation. More details in Appendix F.2.

Another benchmark is stability, which guarantees that GREEDY finds the optimal solution if the instance is sufficiently stable, i.e., its optimal solution remains optimal under small perturbations of the function (Chatziafratis et al., 2017). However, our settings are not perturbation-stable because there are multiple near-optimal solutions.

Results. For small instances where we can exactly compute necessary parameters as well as $\texttt{OPT}$ by brute-force, we follow the experimental setup from (Pokutta et al., 2020). For $k \in [1, 10]$, we randomly choose $n = 2k$ elements to comprise the ground set and analyze the result of DUAL versus benchmarks on objectives, facility location and movie recommendation, from Pokutta, Singh, and Torrico (2020) on the MovieLens dataset. More details in Appendix E.2.

In Figure 3, we observe that DUAL yields the best approximations. For facility location, DUAL shows a near-optimal approximation while other benchmark approximations are

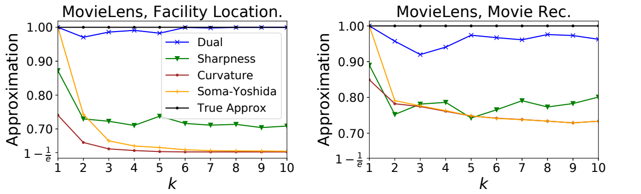

Figure 3. GREEDY approximations computed by DUAL and multiple benchmarks on small instances ($n \leq 20$).

k	DUAL	OPT	CURV.	SOMA.	SHARP.
6	0.0160	0.0317	0.365	0.729	0.830
8	0.0298	0.457	7.99	17.13	18.47
10	0.0716	7.17	156.1	372.4	359.1

Table 1. Average runtimes (in seconds) of benchmark methods on MovieLens facility location setting, where $n = 2k$.

near $1 - 1/e$ for larger k. For movie recommendation, the gap between the different benchmarks is smaller. In both settings and for all k, GREEDY finds a near-optimal solution.

We additionally report benchmark runtimes in Table 1 for the facility location objective and find that CURVATURE, SOMA-YOSHIDA, SHARPNESS and OPT, which all require brute-force computation, become exponentially slower as k increases. At $k = 10, n = 20$, the average time to compute curvature approximation is 156 seconds while sharpness and the improved curvature benchmark both have a computation time of over 300 seconds. These methods are much slower than even brute-force computing OPT which takes 7.2 seconds. While these benchmarks are not scalable, DUAL, which is at least 1000 times faster than these two methods for the facility location objective when $k = 10$, is scalable for larger datasets.

5.2.3. COMPARISON OF PROPOSED METHODS

We compare Method 1, Method 3, and DUAL on a coverage objective and compare Method 3 and DUAL on a non-coverage objective in Figure 4. We observe that even Method 1, which employs the additive lower bound on the dual objective, finds approximations that are above 0.8. This indicates that, unlike the primal objective, the dual objective is close to additive. By partitioning the dual space (Method 3), a small improvement over Method 1 is achieved. Finally, by considering the upper bound on the optimal solution for the functions f_S for $S \in \mathcal{S}$ (DUAL), the approximations further improve. This improvement is minor for MovieLens, but can be around 0.1 for some k on YouTube.

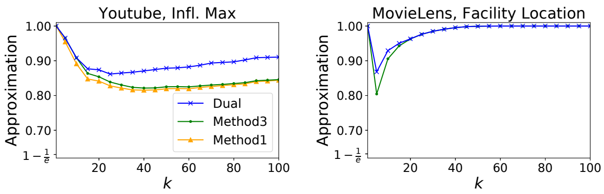

Figure 4. GREEDY approximations computed by Method 1, Method 3, and DUAL.

Acknowledgements

This research was supported by a IBM PhD Fellowship, BSF grant 2014389, NSF grant CAREER CCF-1452961, NSF USICCS proposal 1540428, Google research award, and a Facebook research award.

References

Baeza-Yates, R., Boldi, P., and Chierichetti, F. Essential web pages are easy to find. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 97–107, 2015.

Balkanski, E. and Singer, Y. Approximation guarantees for adaptive sampling. In *International Conference on Machine Learning*, pp. 384–393, 2018.

Blake, C. L. and Merz, C. J. UCI machine learning repository, 1998. URL `http://archive.ics.uci.edu/ml`.

Breuer, A., Balkanski, E., and Singer, Y. The FAST algorithm for submodular maximization. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1134–1143, 2020.

Buchbinder, N., Feldman, M., Naor, J., and Schwartz, R. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1433–1452. SIAM, 2014.

Buchbinder, N., Feldman, M., and Schwartz, R. Comparing apples and oranges: Query tradeoff in submodular maximization. In *SODA*, number CONF, pp. 1149–1168, 2015.

Chatziafratis, V., Roughgarden, T., and Vondrák, J. Stability and recovery for independence systems. *arXiv preprint arXiv:1705.00127*, 2017.

Conforti, M. and Cornuéjols, G. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

FiveThirtyEight. Kaggle, 2019. URL https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

Guestrin, C., Krause, A., and Singh, A. P. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pp. 265–272, 2005.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Kazemi, E., Zadimoghaddam, M., and Karbasi, A. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International conference on machine learning*, pp. 2544–2553, 2018.

Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *KDD*, 2003.

Krause, A., Singh, A., and Guestrin, C. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1 (1):2–es, 2007.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Human Language Technologies*, 2011.

Lindgren, E., Wu, S., and Dimakis, A. G. Leveraging sparsity for efficient submodular data summarization. *Advances in Neural Information Processing Systems*, 29: 3414–3422, 2016.

Lojasiewicz, S. Une propriété topologique des sous-ensembles analytiques réels, in ?les équations aux dérivées partielles (paris, 1962)? éditions du centre national de la recherche scientifique, 1963.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Mirzasoleiman, B., Badanidiyuru, A., and Karbasi, A. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pp. 1358–1367, 2016a.

Mirzasoleiman, B., Badanidiyuru, A., and Karbasi, A. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pp. 1358–1367, 2016b.

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

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

Pokutta, S., Singh, M., and Torrico, A. On the unreasonable effectiveness of the greedy algorithm: Greedy adapts to sharpness. In *International Conference on Machine Learning*, pp. 7772–7782. PMLR, 2020.

Rubinstein, A. and Zhao, J. Budget-smoothed analysis for submodular maximization. *arXiv preprint arXiv:2102.05782*, 2021.

Sakaue, S. and Ishihata, M. Accelerated best-first search with upper-bound computation for submodular function maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Soma, T. and Yoshida, Y. A new approximation guarantee for monotone submodular function maximization via discrete convexity. *arXiv preprint arXiv:1709.02910*, 2017.

Traud, A. L., Mucha, P. J., and Porter, M. A. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

Wolsey, L. A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4): 385–393, 1982.

Yang, J. and Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.