Constrained Robust Submodular Optimization

Thomas Powers[1], **Jeff Bilmes**[1], **Scott Wisdom**[1], **David W. Krout**[2], **and Les Atlas**[1]
[1] Department of Electrical Engineering, University of Washington
{tcpowers, bilmes, swisdom, atlas}@uw.edu
[2] Applied Physics Laboratory, University of Washington
dkrout@apl.washington.edu

Abstract

In this paper, we consider the problem of constrained maximization of the minimum of a set of submodular functions, in which the goal is to find solutions that are robust to worst-case values of the objective functions. Unfortunately, this problem is both non-submodular and inapproximable. In the case where the submodular functions are monotone, an approximate solution can be found by relaxing the problem. We propose an algorithm called GENERALIZED SATURATE (GENSAT) that exploits the submodular structure of the problem and, as a result, returns a near-optimal solution with a constant-factor approximation guarantee on the relaxed problem. GENSAT can handle any submodular constraint, e.g. matroids, cover, and knapsack, and is compatible with any submodular maximization algorithm.

1 Introduction

Submodular function optimization problems have been well studied. Recent advancements have looked at even harder optimization problems that, while not submodular, have submodular structure. One particular problem is constrained maximization of the minimum of submodular functions, i.e., finding a solution that is robust to worst-case values of the objective functions. While this problem is NP-hard, inapproximable, and non-submodular, a relaxed version can be approximately solved if either the constraint or the problem is relaxed [7, 9]. Both of these methods rely on the saturation trick, in which a submodular surrogate problem is substituted for the original and maximized. The solution to the surrogate problem can then be used to find an approximation to the relaxed problem. Maximizing the minimum of a set of submodular functions is useful in problems like robust sensor selection, robust observation selection, submodular fair allocation, and other worst-case optimization problems [9, 7, 1].

Our main contribution is the proposed optimization algorithm, GENERALIZED SATURATE (GENSAT), that is related to SATURATE [7]. GENSAT maximizes the minimum of a set of submodular functions subject to any submodular constraint, e.g. matroids, cover, and knapsack, rather than just cardinality constraints. As a result, GENSAT offers a new strategy for robust submodular optimization over constraint sets (such as matroids) that are not easily expanded as in the case of a cardinality constraints. In section 2, we review submodularity and matroids. In section 3, we describe GENSAT. In section 4 we provide experimental results that demonstrate the efficacy of GENSAT. We conclude in section 5.

2 Submodularity

Submodularity is a property that describes set functions similar to how convexity describes functions in a continuous space. Rather than exhaustively searching over all combinations of subsets, submodular functions provide a fast and tractable framework to compute a solution [8, 4, 6].

Let the set of available objects, known as the ground set, be denoted as V. A submodular function f maps a set of objects denoted by a binary indicator vector of length V to a real number. The

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

binary indicator vector is represented by the expression 2^V since the binary elements can take one of two values and are indexed by elements of set V. Since a value of 1 or 0 for the i^{th} element of the indicator vector denotes the inclusion or exclusion of the i^{th} element of the ground set V, any subset $A \subseteq V$ can be placed in one-to-one correspondence with incidence vectors.

Submodularity can be expressed via the notion of diminishing returns, i.e., the incremental gain of the objective diminishes as the context grows. If we define the incremental gain of adding v to A as $f(v|A) = f(A \cup \{v\}) - f(A)$, then submodularity is defined as any function with $f(v|A) \geq f(v|B)$ for all $A \subseteq B \subset V$ and a $v \notin B$.

Submodularity is very closely tied to structures known as matroids, which generalize the notion of linear independence in vector spaces [2]. One can think of matroids as a generalization of matrices, which extend the definition of rank beyond column vectors of a matrix to more general independent subsets over a finite ground set. More importantly, submodular function optimization allows for matroid independence constraints to be placed on the problem, which means complicated variable dependence patterns can be encoded into the problem. Though matroids are combinatorial objects that grow exponentially with ground set size, their submodular nature allows approximate solutions in polynomial time. Given a finite set V and a finite set of subsets $\mathcal{I} = \{I_1, I_2, \dots\}$, the pair $(V, \mathcal{I})$ is said to be a matroid when the family of sets $\mathcal{I}$ satisfies the following three properties:

1. $\emptyset \in \mathcal{I}$,
2. $I_1 \subseteq I_2 \in \mathcal{I}$,
3. $I_1, I_2 \in \mathcal{I}, |I_1| < |I_2| \implies \exists v \in I_2 \backslash I_1 : I_1 \cup v \in \mathcal{I}$.

3 Proposed algorithm: GENERALIZED SATURATE (GENSAT)

We consider the constrained robust maximization problem given by

$$A^* \in \underset{A \in \mathcal{C}}{\operatorname{argmax}} \min_i f_i(A), \tag{1}$$

where $\mathcal{C}$ denotes a family a feasible sets that are compatible with submodular optimization, e.g. matroid, a knapsack, cover, or other submodular constraint.

To optimize the Equation (1), we develop a novel algorithm GENSAT, which generalizes the SATURATE algorithm created by Krause et al. [7]. Krause et al. use SATURATE to optimize an objective function of the form

$$A^* \in \underset{|A| \leq k}{\operatorname{argmax}} \min_i f_i(A) \tag{2}$$

where $f_i(A)$ is a set of monotone submodular functions. SATURATE solves this worst-case optimization problem by proposing an alternative formulation and relaxing the cardinality constraint from $|A| \leq k$ to $|A| \leq \alpha k$. As long as α is large enough, the solution $\hat{A}$ from the SATURATE algorithm guarantees that

$$\min_i f_i(\hat{A}) \geq \max_{|A| \leq k} \min_i f_i(A) \quad \text{and} \quad |\hat{A}| \leq \alpha k.$$

Krause et al. claim that the only way to achieve a non-trivial guarantee is to relax the constraint, which limits both the types of constraints that can be applied to the problem as well as the values the objective functions can take, i.e. integral or rational valued objective functions. Matroid constraints, for instance, have no immediately obvious relaxation. One way to relax the matroid constraint, however, might be to expand the bases of the matroid, which may lead to undesirable solutions. However, there is another way to achieve non-trivial guarantees, which is to relax the objective itself, leaving the constraints intact, and produce a fractional bound on the objective function, something that is made possible thanks to the use of submodularity. The proposed algorithm, GENSAT, uses this alternative approach to find a solution such that a fraction γ of the submodular functions are above a minimum value β. Moreover, the user can set particular values of β or γ, as long as β is less than the submodular guarantee α and $\gamma < 1$. The derivation for the lower bound is given below.

For a fixed value of c, which can be thought of as the saturation level, we can determine if $f_i(A) \geq c$ from equation (1) via submodular maximization of the following surrogate function:

$$f^c(A) = \frac{1}{M} \sum_{i=1}^{M} \min\{f_i(A), c\}. \tag{3}$$

$f^c(A)$ is a submodular function, because it is a non-negatively weighted sum of functions $\min\{f_i(A), c\}$ which are submodular [5]. At each iteration of GENSAT, we run a submodular maximization algorithm like the one described in Algorithm 2, which selects the element which provides the best incremental gain in the objective and does not violate the constraint.

GENSAT is outlined in Algorithm 1. Given monotone submodular functions $(f_1, ..., f_M)$, approximation guarantee α for the matroid constrained submodular maximization problem, and tolerance threshold ϵ, we first set c_{min} and c_{max} to values that ensure the true optimal value lies in the interval. While performing a binary search over c, we test the value of the approximate solution $f^c(\hat{A})$ against the lower bound αc. If the approximate solution is less than the lower bound, we know that the true optimal is less than c, so we limit the search to the lower half of the interval. Likewise, if the lower bound is met, we store the solution (which, as we describe below, is fractionally good w.r.t. the current c) and then continue attempting to find a better one (higher c) by searching over the upper half of the interval. We stop when the range falls within the tolerance.

Algorithm 1 GENSAT $(f_1, ..., f_M, \alpha, \epsilon)$

1: $c_{min} \leftarrow 0, c_{max} \leftarrow \min_i f_i(V)$
2: **while** $(c_{max} - c_{min}) > \epsilon$ **do**
3: $\quad c \leftarrow (c_{max} - c_{min})/2$
4: $\quad f^c(A) \leftarrow \frac{1}{M}\sum_{i=1}^{M} \min\{f_i(A), c\}$
5: $\quad \hat{A} \leftarrow$ GREEDY (f^c, c)
6: $\quad$ **if** $f^c(\hat{A}) < \alpha c$ **then**
7: $\quad\quad c_{max} \leftarrow c$
8: $\quad$ **else**
9: $\quad\quad c_{min} \leftarrow c, \quad A_{best} \leftarrow \hat{A}$
10: $\quad$ **end if**
11: **end while**
12: **return** A_{best}

Algorithm 2 GREEDY (f^c, c)

$A \leftarrow \emptyset$
while $\exists a \in V \backslash A$ s.t. $A \cup \{a\} \in \mathcal{C}$
and $f^c(a|A) > 0$ **do**
$\quad S \leftarrow \{a \in V \backslash A : A \cup \{a\} \in \mathcal{C}\}$
$\quad s^* \leftarrow \text{argmax}_{s \in S} f^c(a|A)$
$\quad A \leftarrow A \cup \{s^*\}$
end while
return A

Theorem 1. *Given a value $\beta < \alpha$,* GENSAT *finds a solution $\hat{A}$ that guarantees the following fraction γ of the M functions min$\{f_i(\hat{A}), c\} \geq \beta c$:*

$$\gamma \geq \frac{\alpha - \beta}{1 - \beta}$$

where α is the approximation guarantee for matroid independence set constrained submodular maximization problem.

Proof: $f^c(A) \geq c$ only if $\min_i f_i(A) \geq c$. When all $f_i(A) \geq c$, then $f^c(A) = c$. Likewise, when any $f_i(A) < c$ then the $f^c(A) < c$, since the maximum value of $f^c(A)$ is c. The greedy solution $\hat{A}$ for maximizing a monotone submodular function subject to a constraint is $f^c(\hat{A}) \geq \alpha f^c(A^*)$, where α depends on the algorithm chosen and the constraint. If line 6 of Algorithm 1 is true, then $f^c(\hat{A}) < \alpha c$ which implies that $\min_i \ f_i(A^*) < c$. Line 6 being true also implies that c is too large, so $c_{max} \leftarrow c$. At line 12 of Algorithm 1, $(c_{max} - c_{min}) \leq \epsilon$ and the true optimal value $\min_i \ f_i(A^*)$ is in the interval $[c_{min}, c_{max}]$. The submodular approximation guarantee ensures that $f^c(\hat{A}) \geq \alpha f^c(A^*)$ and $f^c(\hat{A}) \geq \alpha c$. Given a value for β, the terms of $f^c(\hat{A})$ can be split into two groups, one that have value less than βc and the other greater than or equal to βc:

$$f^c(\hat{A}) = \frac{1}{M} \sum_{i:\min\{f_i(A),c\} < \beta c} \min\{f_i(A), c\} + \frac{1}{M} \sum_{i:\min\{f_i(A),c\} \geq \beta c} \min\{f_i(A), c\}.$$

Let γ be the fraction of terms that meet the βc threshold. Then, the two summation terms become $f^c(\hat{A}) = (1-\gamma)\beta c + \gamma c \geq \alpha c$. Rearranged, the expression becomes $\gamma \geq \frac{\alpha - \beta}{1 - \beta}$. $\square$

A visualization of the lower bound βc over values of the saturation level c and fractional number of functions $\lfloor \gamma M \rfloor$ for $M = 8$ is shown in figure 1. In this case, $\alpha = 1/2$, which is the submodular guarantee for a monotone submodular function constrained by a single matroid via the greedy algorithm [3]. Notice that the bound becomes trivial, $(\beta c = 0)$, when $\gamma \geq \alpha$, i.e. where $\lfloor \gamma M \rfloor \geq 5$.

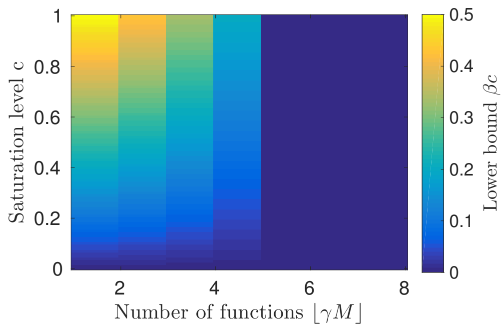

Figure 1: Visualization of lower bound βc as the saturation level c and the fractional guarantee $\lfloor \gamma M \rfloor$ vary for $M = 8$ and $\alpha = 1/2$. Note that the bound worsens as we demand that a larger fraction γ of the functions have non-trivial minimum values and if the algorithm returns low values of the saturation level c. Also note that the bound becomes trivial, ($\beta c = 0$), when $\gamma \geq \alpha$, i.e. where $\lfloor \gamma M \rfloor \geq 5$.

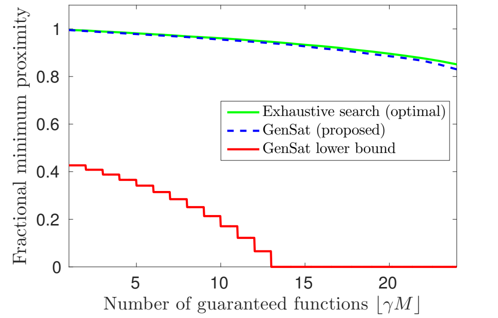

Figure 2: Fractional minimum proximity results averaged across 100 realizations of a $M = 24$ sensor placement problem, comparing the proposed GENSAT (dashed blue line) method to exhaustive search (solid green line), and submodular guarantee lower bound on GENSAT using the forward greedy algorithm (solid red line). GENSAT achieves near-optimal performance, even when the bound is trivial.

4 Experiment

In this section, we test GENSAT on a synthetic sensor placement problem in which the goal is to select a set $\hat{A}$ of sensor locations to maximize the minimum proximity between all locations and sensors, while meeting restrictions on how many sensors can be placed in particular regions. The ground set V consists of all possible locations for the sensors. The objective functions in the min are facility location functions given as $f_i = \max_{a \in A}\{w_{ia}\}$, where w_{ia} is the proximity between location $i \in V$ and sensor $a \in A$. The randomly generated symmetric proximity matrix describes the closeness between locations and sensors and has values uniformly sampled from $[0, 1]$. The constraint on the number of sensors that can be placed in a given region is encoded in a randomly generated partition matroid, with four partitions and partition cardinality constraints of three. We use the forward greedy algorithm described in algorithm 2, so $\alpha = 1/2$. We run 100 trials, each with a different similarity matrix and partition matroid, and compare the estimated solution against the true optimal solution $A^\star$ found via exhaustive search over the maximal independent sets of the partition matroid.

Results are shown in figure 2. The optimal (solid green line) and estimated (dashed blue line) fractional minimum proximity are computed over all fractions γ of the M functions. The fractional minimum proximity is computed as $\min_{j \in \mathcal{J}_\gamma} f_j(A)$, where $\mathcal{J}_\gamma$ is the set of indices of the $\lfloor \gamma M \rfloor$ functions f_i with the greatest value. The lower bound βc (solid red line) from theorem 1 is computed for every fraction γ of the M functions as well. Since this lower bound is bicriterion, the minimum proximity β could be fixed to find the fraction of functions γ that meet the threshold βc. Note that the proposed GENSAT algorithm nearly matches the optimal exhaustive search approach, even in the region where the bound becomes trivial (i.e., where $\gamma > \alpha \Rightarrow \lfloor \gamma M \rfloor \geq 13 \Rightarrow \beta = 0$).

5 Conclusion

Constrained maximization of the minimum of a set of monotone submodular functions has several applications in machine learning, including robust sensor selection, robust observation selection, and submodular fair allocation. While this problem is non-submodular, NP-hard, and inapproximable, it is not entirely intractable. In this paper, we show that through relaxation of the minimum, a fraction of the functions in the minimum are lower-bounded. We present the algorithm GENSAT that achieves the lower bound and derive the formula theoretical lower bound. We then demonstrate that GENSAT can obtain near optimal performance on a matroid-constrained sensor placement problem.

References

[1] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010.

[2] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[3] M.L. Fisher, G.L. Nemhauser, and L.A. Wolsey. An analysis of approximations for maximizing submodular set functions-ii. In M.L. Balinski and A.J. Hoffman, editors, *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 73–87. Springer Berlin Heidelberg, 1978.

[4] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

[5] Toshihiro Fujito. Approximation algorithms for submodular set cover with applications. *IEICE Transactions on Information and Systems*, 83(3):480–487, 2000.

[6] A. Krause. SFO: A toolbox for submodular function optimization. *J. Mach. Learn. Res.*, 11:1141–1144, March 2010.

[7] A. Krause, B. McMahan, Guestrin C., and Gupta A. Robust submodular observation selection. *Journal of Machine Learning Research (JMLR)*, 9:2761–2801, December 2008.

[8] L. Lovász. Submodular functions and convexity. In Achim Bachem, Bernhard Korte, and Martin Grötschel, editors, *Mathematical Programming The State of the Art*, pages 235–257. Springer Berlin Heidelberg, 1983.

[9] T. Powers, J. Bilmes, D. W. Krout, and L. Atlas. Constrained robust submodular sensor selection with applications to multistatic sonar arrays. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 2179–2185, July 2016.