# Robust submodular maximization under matroid constraints


**Sebastian Pokutta, Mohit Singh & Alfredo Torrico**

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Email: atorrico3 at gatech dot edu, sebastian.pokutta, mohit.singh at isye dot gatech dot edu


## Problem Formulation

- Ground set $V = \{1, \dots, n\}$ and matroid $\mathcal{M} = (V, \mathcal{I})$.
- Collection of of non-negative, monotone, submodular[1] functions $f_i : 2^V \to \mathbb{R}_+$. We want to solve:

$$\max_{S\in\mathcal{I}} \min_{i\in[k]} f_i(S) \tag{1}$$

- Krause et al. [6] prove that problem (1) is NP-hard to approximate to any polynomial factor. This motivates the necessity of bi-criteria solutions.

**Our Approach:** relax the constraints. We construct a small family of feasible sets whose union $S^{\text{ALG}}$ has (nearly) optimal objective value.

**Example in partition constraints:**

- $V = P_1 \uplus \cdots \uplus P_q$ and $\mathcal{I} = \{S \subseteq V : |S \cap P_j| \le b_j, \ \forall j \in [q]\}$.
- $S^{\text{ALG}} = S_1 \cup \cdots \cup S_\ell \quad \to \quad |S^{\text{ALG}} \cap P_j| \le \ell \cdot b_j$.

## Our Results and Contributions

- Extended version of the *threshold greedy* algorithm introduced by Badanidiyuru and Vondrak [2].

**Algorithm 1** Extended Threshold-Greedy

**Input:** $\ell \ge 1$, ground set $V$ with $n := |V|$, monotone submodular function $g : 2^V \to \mathbb{R}_+$, matroid $\mathcal{M} = (V, \mathcal{I})$ and $\delta > 0$.
**Output:** feasible sets $S_1, \dots, S_\ell \in \mathcal{I}$.

```
for τ = 1, ..., ℓ do
    S_τ ← ∅
    d ← max_{e∈V} g(∪_{j=1}^{τ-1} S_j + e)
    for (w = d; w ≥ (δ/n)d; w ← (1 − δ)w) do
        for e ∈ V\S_τ do
            if S_τ + e ∈ I and g_{∪_{j=1}^{τ} S_j}(e) ≥ w then
                S_τ ← S_τ + e
```

**Proposition 1.** *Given $\ell \ge 1$, a monotone submodular function $g : 2^V \to \mathbb{R}_+$ with $g(\emptyset) = 0$, and parameter $\delta > 0$, Algorithm 1 returns feasible sets $S_1, \dots, S_\ell \in \mathcal{I}$ such that*

$$g(S_1 \cup \cdots \cup S_\ell) \ge \left(1 - \left(\frac{1}{2-\delta}\right)^\ell\right) \cdot \max_{S\in\mathcal{I}} g(S).$$

*This algorithm performs $O(\frac{n\ell}{\delta}\log\frac{n}{\delta})$ function calls, independent of the rank of the matroid.*

- Efficient bi-criteria algorithm for problem (1).
  - Outer loop: get an estimate $\gamma$ on the value $\text{OPT} := \max_{S\in\mathcal{I}} \min_{i\in[k]} f_i(S)$ via a binary search.
  - Inner loop: for a given $\delta > 0$, run Algorithm 1 on $g^\gamma(S) := \frac{1}{k}\sum_{i\in[k]} \min\{f_i(S), \gamma\}$ with $\ell = \lceil \log\frac{2k}{\epsilon} / \log(2-\delta) \rceil$.
  - We stop the binary search whenever we get a relative error of $1 - \epsilon/2$, namely, $(1 - \epsilon/2)\,\text{OPT} \le \gamma \le \text{OPT}$.

**Theorem 1.** *For problem (1), there is a polynomial time algorithm that returns a set $S^{\text{ALG}}$, such that for given $0 < \epsilon, \delta < 1$, for all $j \in [k]$ it holds*

$$f_j(S^{\text{ALG}}) \ge (1 - \epsilon) \cdot \max_{S\in\mathcal{I}} \min_{i\in[k]} f_i(S),$$

*where $S^{\text{ALG}} = S_1 \cup \cdots \cup S_\ell$ with $\ell = \lceil \log\frac{2k}{\epsilon} / \log(2-\delta) \rceil$, and $S_1, \dots, S_\ell$ are feasible.*

- Implementation improvement: early stopping criteria.
  - If in iteration $\tau \in [\ell]$ of Algorithm 1 on $g^\gamma$ we obtain a set $S_\tau$ such that $g^\gamma(\cup_{t=1}^{\tau} S_t) < (1 - 1/(2-\delta)^\tau) \cdot \gamma$, then we stop and update the upper bound on the optimum to be $\gamma$.

## Computational Experiments

In all experiments we consider partition constraints $\mathcal{I} = \{S : |S \cap P_j| \le b, \ \forall j \in [q]\}$, note same budget $b$ for each part.

**Tested Algorithms and Baselines.**

- (`prevE-G`) the extended greedy of [1] with no improvements, $\ell = \lceil \log\frac{2k}{\epsilon} \rceil$.
- (`E-G`) the extended greedy of [1] with improvements, $\ell = \lceil \log\frac{2k}{\epsilon} \rceil$.
- (`E-ThG`) the extended threshold greedy (this work), $\ell = \lceil \log\frac{2k}{\epsilon} / \log(2-\delta) \rceil$.
- (`E-StochG`) a heuristic we called extended stochastic greedy, $\ell = \lceil \log\frac{2k}{\epsilon} \rceil$.
- (`RS`) Random Selection.
- (`G-Avg`) the lazy greedy algorithm on the average function $\frac{1}{k}\sum_{i\in[k]} f_i$.

### Experiment #1: Non-parametric Learning.

- In this experiment we show that our improved algorithms clearly outperform `prevE-G`.
- Let $X_V$ be a set of random variables corresponding to bio-medical measurements, indexed by a ground set of patients $V$. We assume for every subset $S \subseteq V$, $X_S \sim \mathcal{N}(\mu_S, \Sigma_{S,S})$.
- We assume the covariance matrix $\Sigma_{S,S} = [\mathcal{K}_{e,e'}]_{e,e'\in S}$ to be given in terms of the squared exponential kernel $\mathcal{K}_{e,e'} = \exp(-\|x_e - x_{e'}\|_2^2/h)$ with $h = 0.75$.
- We use the Parkinson Telemonitoring dataset [8] consisting of $n = 5{,}875$ patients with early-stage Parkinsons disease and the corresponding bio-medical voice measurements with 22 attributes.
- Assuming the Informative Vector Machine (IVM) model, we want to solve

$$\max_{S\in\mathcal{I}} \min_{i\in[20]} \left\{ \frac{1}{2}\log\det(\mathbf{I} + \sigma^{-2}\Sigma_{SS}) + \sum_{e\in S\cap\Lambda_i} \eta_e \right\},$$

where $\sigma^2 = 1$, each $\Lambda_i$ is a random set of size 1,000, and $\eta \sim [0,1]^V$ is a uniform error vector.

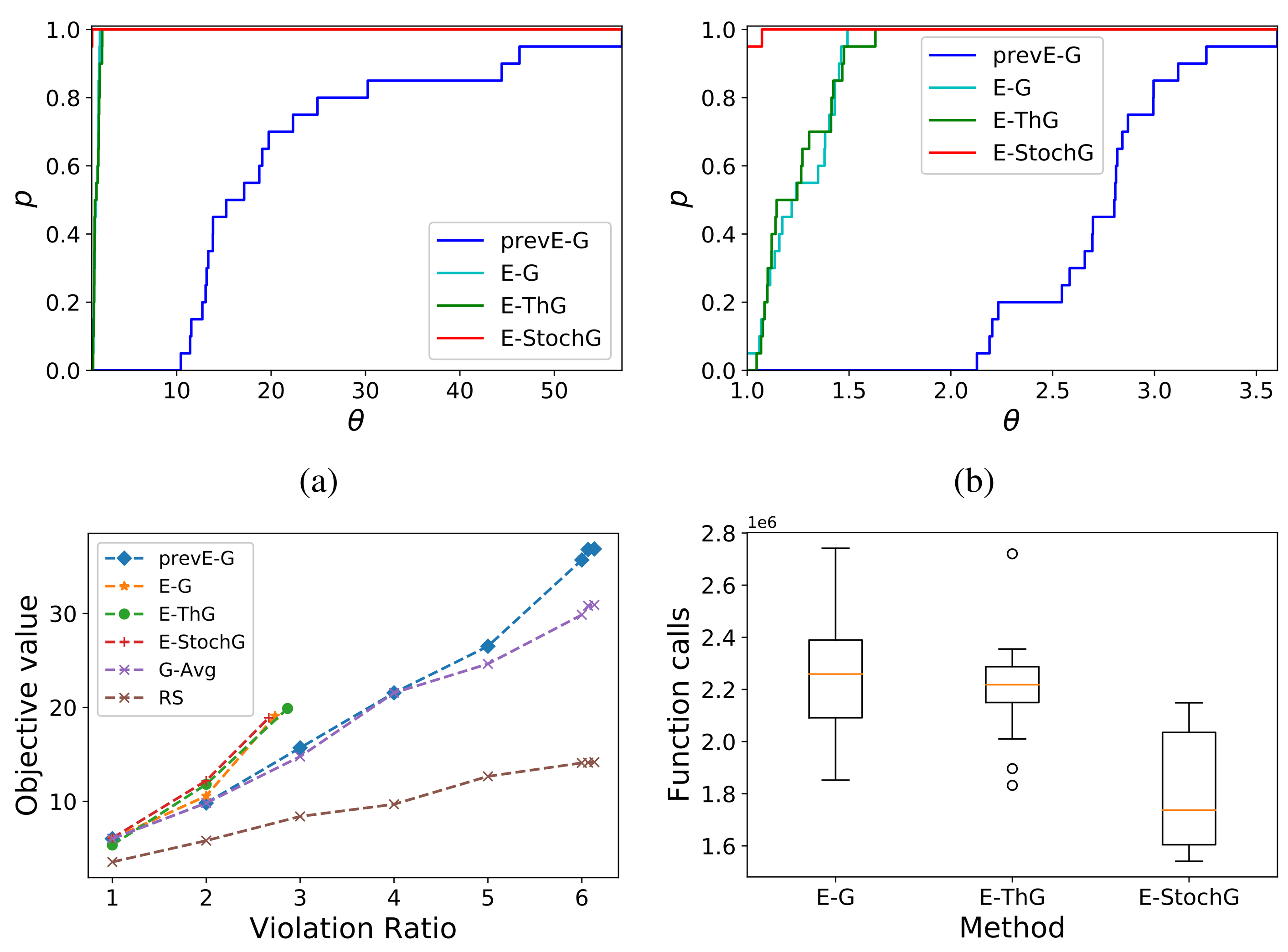


**Figure 1:** *Non-parametric learning.* We made 20 random runs considering $q = 3$ parts and budget $b = 5$. Performance profiles (a) for running time (note that `E-G` is covered by `E-ThG`) and (b) for function calls. In (c) is the objective value versus the violation ratio in a single run of each method. In (d) is the box-plot for the function calls.

### Experiment #2: Exemplar-based Clustering.

- In this experiment we show that `E-ThG` outperforms the rest since its no-dependency on the rank. We do not implement `prevE-G`.
- Solving the $k$-medoid problem is a common way to select a subset of exemplars that represent a large dataset $V$ [5]. This is done by minimizing the sum of pairwise dissimilarities between elements.
- Formally, define $L(S) = \frac{1}{V}\sum_{e\in V} \min_{v\in S} d(e, v)$, where $d : V \times V \to \mathbb{R}_+$ is a distance function that represents the dissimilarity between a pair of elements. By introducing an appropriate auxiliary element $e_0$, it is possible to define a new objective $f(A) := L(\{e_0\}) - L(A + e_0)$ that is monotone and submodular [4], thus maximizing $f$ is equivalent to minimizing $L$.
- We use the VOC2012 dataset [3] where the ground set $V$ corresponds to images. These images are represented by feature vectors (number of elements in them).
- We use the Euclidean distance $d(e, e') = \|x_e - x_{e'}\|$ and $x_{e_0}$ as the origin. We want to solve

$$\max_{S\in\mathcal{I}} \min_{i\in[20]} \left\{ f(S) + \sum_{e\in S\cap\Lambda_i} \eta_e \right\},$$

where $f$ is the function defined above, each $\Lambda_i$ is a random set of fixed size with $i \in [20]$, and $\eta \sim [0,1]^V$ is a uniform error vector.

- We consider two experiments: (`small`) with $n = 3{,}000$ images, 20 random instances considering $q = 6$ and $b = 70$, $|\Lambda_i| = 500$ and (`large`) with $n = 17{,}125$ images, 20 random instances $q \in \{10, \dots, 29\}$ parts and budget $b = 5$, $|\Lambda_i| = 3{,}000$.

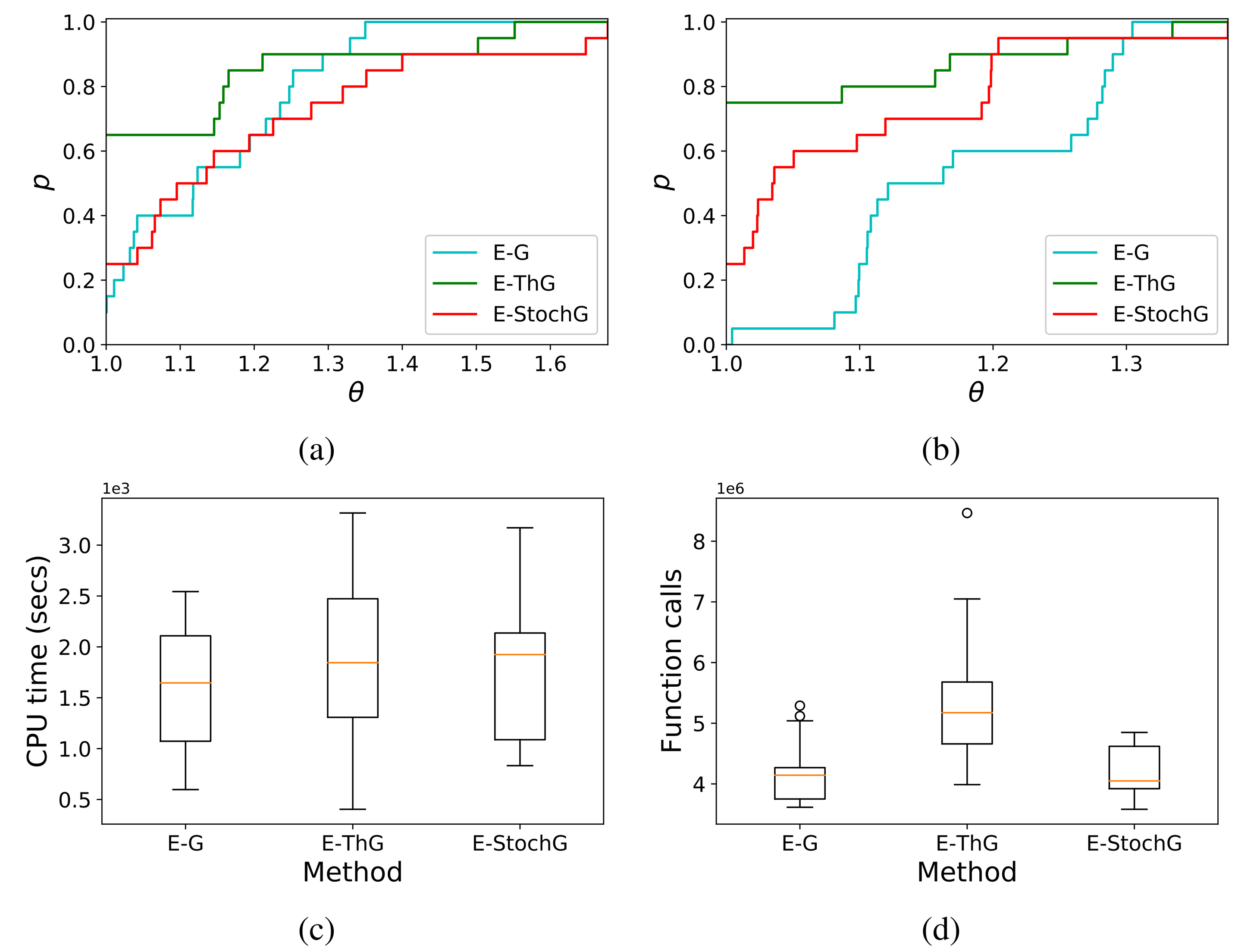


**Figure 2:** *Clustering:* (`small`) performance profiles (a) for the running time and (b) for the function calls. (`Large`) box-plots (c) for the running time and (d) for the function calls.

## References


[1] N. Anari, N. Haghtalab, S. Seffi, S. Pokutta, M. Singh, and A. Torrico. Robust submodular maximization: Offline and online algorithms. *arXiv preprint arXiv:1710.04740*, 2017.

[2] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1497–1514, 2014.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[4] R. Gomes and A. Krause. Budgeted nonparametric learning from data streams. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 391–398, 2010.

[5] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. Wiley-Interscience, 2009.

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

[7] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 1812–1818. AAAI Press, 2015.

[8] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig. Enhanced classical dysphonia measures and sparse regression for telemonitoring of parkinson's disease progression. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 594–597, March 2010.


[1] For any $e \in V$ and $A \subseteq B \subseteq V \setminus \{e\}$, $g_A(e) \ge g_B(e)$, where $g_A(e) := g(A + e) - g(A)$ and $A + e := A \cup \{e\}$