Submodular Maximization with Matroid and Packing Constraints in Parallel

Alina Ene [1], Huy L. Nguyen [2], Adrian Vladu [1]
[1]Boston University, [2]Northeastern University

Submodular Functions and Submodular Maximization

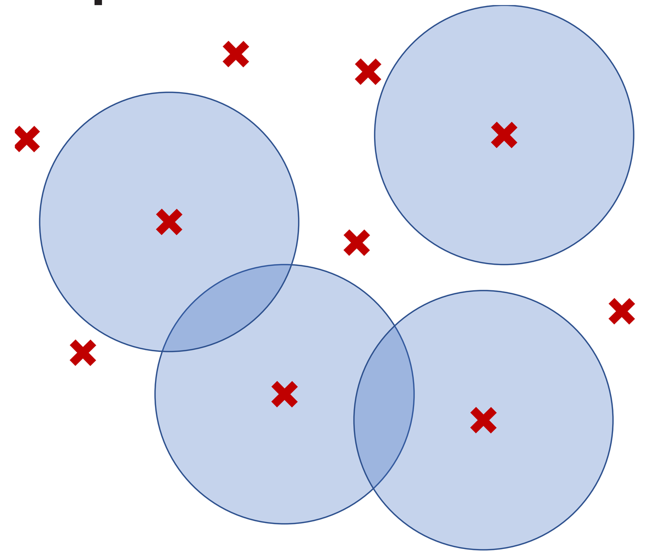
Submodular functions are functions that satisfy the diminishing returns property. Formally, they are defined as $F : 2^{[n]} \to \mathbb{R}$, such that

$$F(A \cup B) + F(A \cap B) \le F(A) + F(B), \text{ for all } A, B \subseteq [n].$$

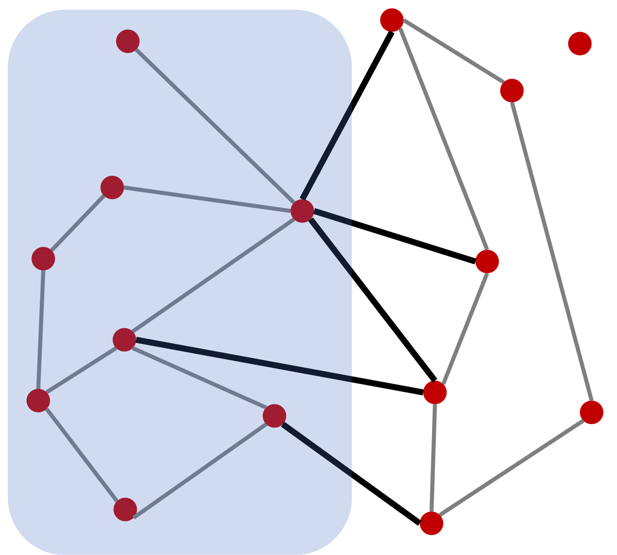
Equivalently, they satisfy the *diminishing returns* property:

$$F(A \cup \{e\}) - F(A) \ge F(B \cup \{e\}) - F(B) \text{ for all } A \subseteq B \subseteq [n], e \in [n].$$

Examples:



$F(S) = \text{area}\left(\cup_{i\in S} D_i\right)$



$F(S) = |E(S, \overline{S})|$

Submodular maximization with down-monotone constraints

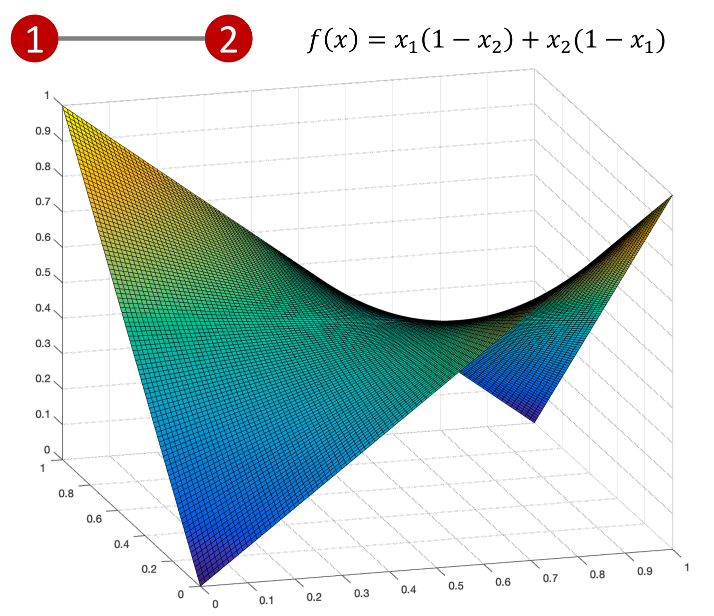
- We want to maximize F, but the set of elements we are allowed to choose is constrained (i.e. cardinality/matroid constraint) to $\mathbf{P}$.
- Classically, one can obtain $(1-1/e)$-approximation for monotone F, and $(1/e)$-approximation for non-monotone F.
- *Key tool: multilinear extension.* Extends F to $[0,1]^n$ via polynomial

$$f(\mathbf{x}) = \sum_{S\subseteq[n]} \left(F(S)\cdot \Pi_{i\in S} x_i \cdot \Pi_{i\notin S}(1-x_i)\right).$$

  - Can approximate $f(\mathbf{x})$, $\nabla f(\mathbf{x})$ very well by evaluating F at *poly*(n) samples.
  - $\nabla^2 f \le 0$ pointwise; equivalently f is concave in nonnegative directions.

Classical results overview

Multilinear extension for cut function.



- Monotone f: continuous greedy
  $\dot{\mathbf{x}} = \arg\max_{\mathbf{v}\in\mathbf{P}} \langle \nabla f(\mathbf{x}), \mathbf{v}\rangle$.
  - Frank-Wolfe constrained to nonnegative directions.
  - Makes at least as much progress as it would make by moving towards $\mathbf{x} \vee \mathbf{x}^* \in \mathbf{P}$, so can always gain $f(\mathbf{x}\vee\mathbf{x}^*) - f(\mathbf{x}) \ge f(\mathbf{x}^*) - f(\mathbf{x})$ per unit of step due to concavity.
  - Gain given by solving $\dot{v}(t) = f(\mathbf{x}^*) - v(t)$, $v(0) = 0$, $v(t) = f(\mathbf{x}^*)(1-e^{-t})$, so $v(1) = f(\mathbf{x}^*)(1-1/e)$.
- Non-monotone f: dampen step

$$\dot{\mathbf{x}} = (1-\mathbf{x})\cdot \arg\max_{\mathbf{v}\in\mathbf{P}} \langle \nabla f(\mathbf{x}), \mathbf{v}\rangle.$$

  - Issue: since f is non-monotone, we do not have $f(\mathbf{x}\vee\mathbf{x}^*) \ge f(\mathbf{x}^*)$, so can not gain the same amount of progress.
  - Instead, use $f(\mathbf{x}\vee\mathbf{x}^*) \ge (1-\|\mathbf{x}\|_\infty) f(\mathbf{x}^*)$.
  - Dampening the step, slows the growth of $\|\mathbf{x}\|_\infty$.

Can these algorithms be made practical?

Our Results

- We are concerned with maximizing f in few parallel rounds of adaptivity (as defined by [1]). Pay for queries to f and ∇f.
- We only need to consider the more general setup for arbitrary functions concave along non-negative directions (DR-submodular).

For monotone and non-monotone DR-submodular f, we obtain approximations that match the classical bounds up to additive ε.

constraint type \ regime	monotone: $(1-1/e-\varepsilon)$-approx.	non-monotone: $(1/e-\varepsilon)$-approx.
packing constraints $\mathbf{Ax} \le \mathbf{1}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$	$O\left(\frac{\log(n/\varepsilon)\log m}{\varepsilon^2}\right)$	$O\left(\frac{\log(n/\varepsilon)\log(m+n)\log(1/\varepsilon)}{\varepsilon^2}\right)$
matroid constraint	$O\left(\frac{\log^2 n}{\varepsilon^3}\right)$	$O\left(\frac{\log^2 n}{\varepsilon^3}\right)$

Rounds of adaptivity required by our algorithms.

Warm-up: Monotone Functions with Packing Constraints

Discretizing continuous greedy.

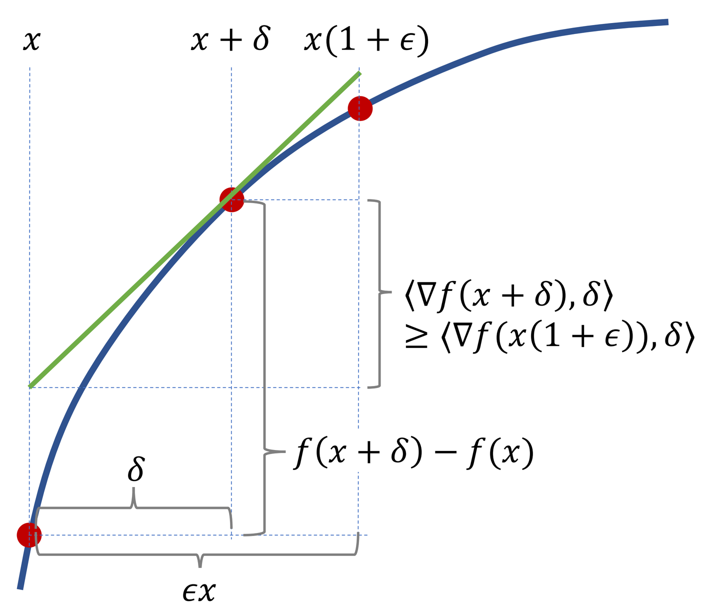
- Potential function: $\frac{f(\mathbf{x})}{\mathtt{smax}(\mathbf{Ax})}$, where $\mathtt{smax}(\mathbf{y}) = \frac{1}{\eta}\log\sum_i \exp(\eta y_i)$ is a smooth approximation for $\max(\mathbf{y})$.
- Iteratively update $\mathbf{x}$ such that we gain what we deserve:

$$\frac{f(\mathbf{x}+\boldsymbol{\delta}) - f(\mathbf{x})}{\mathtt{smax}(\mathbf{A}(x+\delta)) - \mathtt{smax}(\mathbf{Ax})} \ge M - f(\mathbf{x})$$

  where M is a guess on $f(\mathbf{x}^*)$.

1. Lower bound increase in f using gradient at future point: if $\boldsymbol{\delta} \le \varepsilon\mathbf{x}$,
$$f(\mathbf{x}+\boldsymbol{\delta}) - f(\mathbf{x}) \ge \langle \nabla f(\mathbf{x}+\boldsymbol{\delta}), \boldsymbol{\delta}\rangle \ge \langle \nabla f(\mathbf{x}(1+\varepsilon)), \boldsymbol{\delta}\rangle.$$
2. Upper bound increase in smax using smoothness:
$$\langle \mathtt{smax}(\mathbf{A}(\mathbf{x}+\boldsymbol{\delta})) - \mathtt{smax}(\mathbf{Ax}) \lesssim \langle \nabla_{\mathbf{x}}\mathtt{smax}(\mathbf{Ax}), \boldsymbol{\delta}\rangle$$

How to update x?



- Scale coordinates x_i with sufficient relative gain
$$\frac{\nabla_i f((1+\varepsilon)\mathbf{x})}{\nabla_{x_i}\mathtt{smax}(\mathbf{Ax})} \ge M - f(\mathbf{x})$$
  (based on [2]).
- Failure to increase any coordinate certifies that we are done.

Why does it terminate fast?

- Can almost black box packing LP solver [3]; main difference is that the linear objective $\nabla f(\mathbf{x}(1+\varepsilon))$ changes.
- Gradient is monotone, so the linear objective only decreases. We can still analyze packing LP with decreasing objective. Termination in $\tilde{O}(\varepsilon^{-2})$ parallel rounds.

What about non-monotone functions?

- Issue: some gradient coordinates may be negative.
- Solution: do not touch them until they are positive, careful analysis.

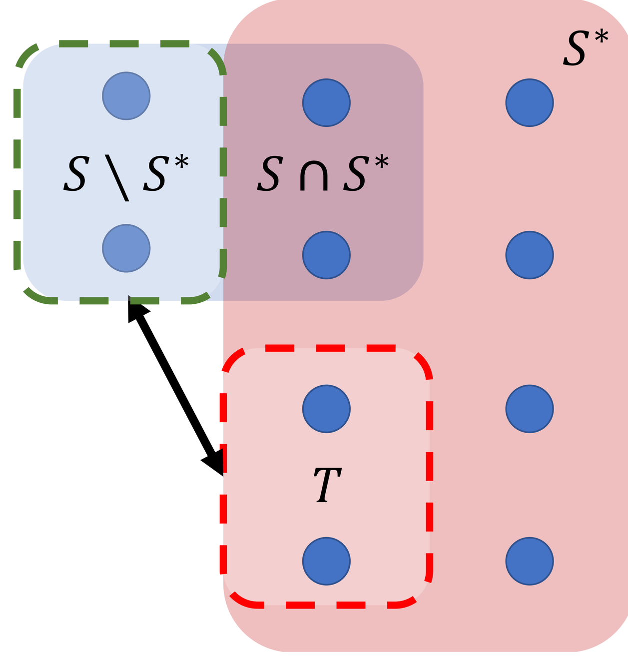
Matroid Constraints

Observation 1: Greedy algorithm gives 1/2-approximation [4].

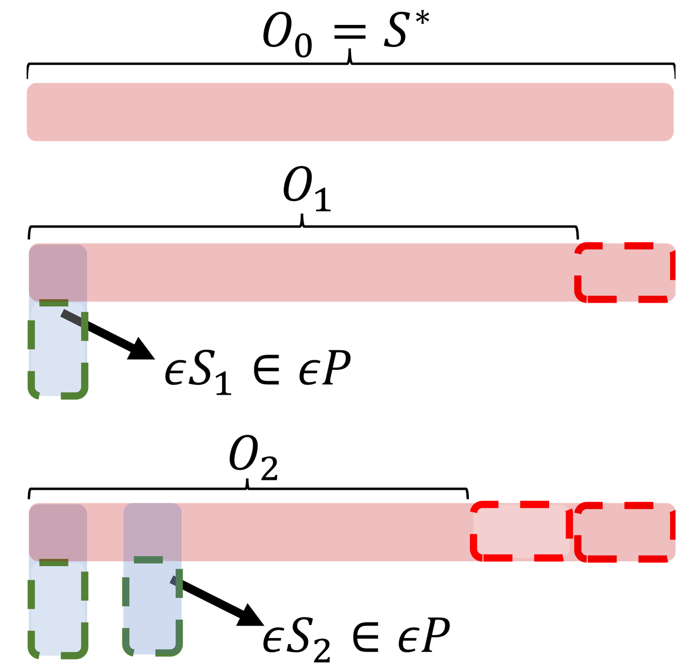
$$S' = S \cup \arg\max_{e: S\cup\{e\}\in\mathbf{P}} f(S\cup\{e\}) - f(S).$$

Proof idea: Maintain $S \in \mathbf{P}$ and matroid base $O = S^* \cup S \setminus T$, where $T \subseteq S^* \setminus S$ is obtained via exchange property. Greedy enforces $f(S) \ge f(S^*) - f(O)$.

Observation 2: Running greedy $O(1/\varepsilon)$ times improves to $1 - 1/e - \varepsilon$.
Proof idea: Break greedy over $O(1/\varepsilon)$ iterations.



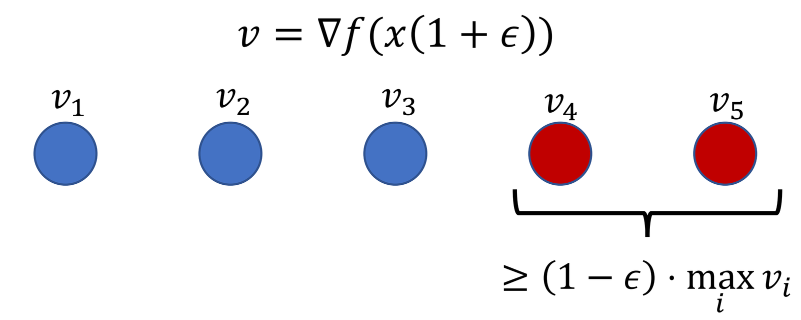
New base $S^* \cup S \setminus T$ can be reached from S.



Amplifying greedy.

- Both observations apply to continuous f.

Observation 3:
Parallelize greedy by simultaneously increasing all x_i with large marginal gains (use gradient lookahead).



- Marginal gains only decrease.
- Coordinates are scaled by $1+\varepsilon$, none gets touched too many times.

Takeaway notes

- Continuous optimization makes discrete problems easy – the only property of f we use is concavity along non-negative directions.
- Submodular maximization with packing constraints is no harder than solving positive LP's in parallel.
- Extend techniques to other discrete problems? What else is there?

References

[1] E. Balkanski and Y. Singer, "The adaptive complexity of maximizing a submodular function", in *STOC 2018*.

[2] C. Chekuri and K. Quanrud, "Submodular function maximization in parallel via the multilinear relaxation", in *SODA 2019*.

[3] M. W. Mahoney, S. Rao, D. Wang, and P. Zhang, "Approximating the solution to mixed packing and covering LPs in parallel $\tilde{O}(\varepsilon^{-3})$ time", in *ICALP 2016*.

[4] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions", *Mathematical programming*, 1978.