Maximizing Submodular Functions Exponentially Faster

Eric Balkanski   Yaron Singer
Harvard University

Presented by Kent Quanrud, UIUC

Eric Balkanski

Yaron Singer

under the weather

Submodular Function Maximization in Parallel via the Multilinear Relaxation

Chandra Chekuri   Kent Quanrud

UIUC

GOAL:

maximize $f(S)$ s.t. $|S| \leq k$

in parallel in the oracle model

where

- $f: 2^N \to \mathbb{R}_{\geq 0}$ is a monotone submodular function

- $k \in \mathbb{N}$

Monotone submodular functions

set function $f: 2^N \to \mathbb{R}_{\geq 0}$

- $f(\emptyset) = 0$
- (increasing) $S \subseteq T \Rightarrow f(S) \leq f(T)$
- [decreasing marginal returns] denote $f_S(U) = f(S \cup U) - f(S)$

  = "marginal value of U to S"

$$S \subseteq T \Rightarrow f_S(U) \geq f_T(U)$$

- Oracle: given S, returns $f(S)$

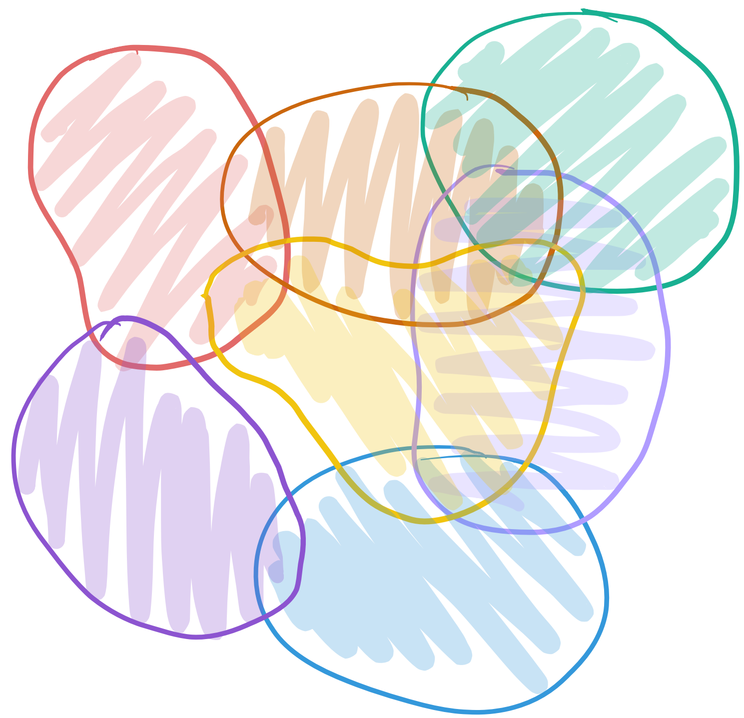
e.g. Maximum coverage

Input: n set $S_1, S_2, \ldots, S_n$, cardinality k

Goal: choose k sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$

maximizing their union $\left| \bigcup_{j=1}^{k} S_{i_j} \right|$

- NP-hard
- $1-\frac{1}{e}$ APX-hardness
- $1-\frac{1}{e}$ APX by greedy



Greedy algorithm [Nemhauser, Wolsey]

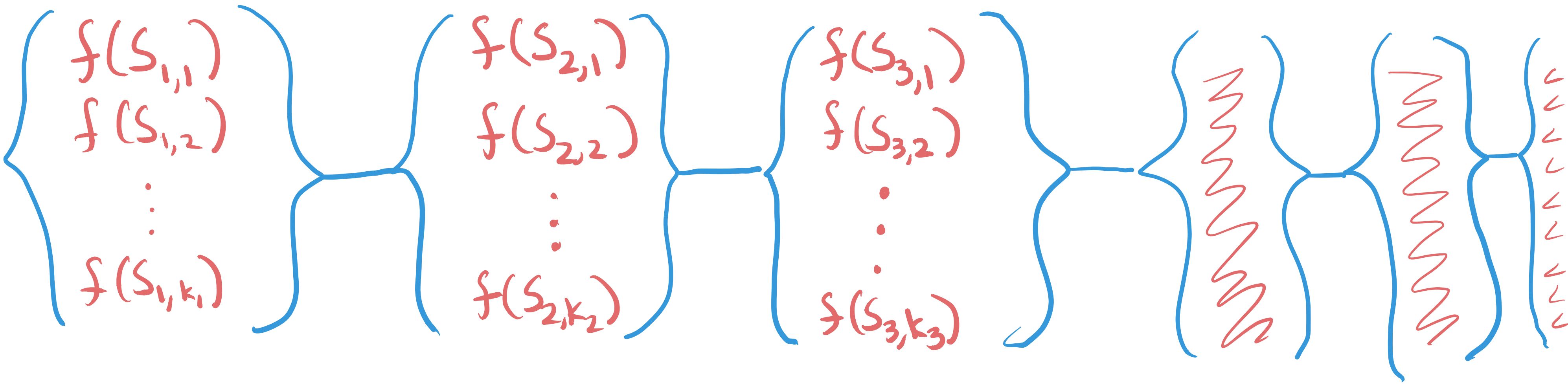
Input: $f: 2^N \to \mathbb{R}_{\geq 0}$, N, k

1. $S \leftarrow \emptyset$
2. while $|S| < k$
   a. $e \leftarrow \underset{e \in N}{\operatorname{argmax}} f_S(e)$
   b. $S \leftarrow S + e$
3. return S

- simple
- Optimal $1 - \frac{1}{e}$ APX
- Sequential: next selection depends on previous selections

Adaptivity and parallelization

- Queries to f divided into "adaptive rounds"
- Choice of queries can only depend on queries to f of previous rounds



- e.g. greedy: nk total queries

  k adaptive rounds

Q: how much adaptivity/depth needed to maximize $f(S)$ s.t. $|S| \le k$?

- greedy needs k
- [Balkanski & Singer] $\Omega\left(\frac{\log n}{\log\log n}\right)$ necessary

  $\frac{1}{3}$-APX w/ $O(\log n)$ rounds
- [Balkanski, Rubinstein, Singer] [Ene, Nguyen] $(1-\frac{1}{e}-\varepsilon)$ w/ $O\left(\frac{\log n}{\mathrm{poly}(\varepsilon)}\right)$ rounds
- [Chekuri, Q.]: for general packing constraints

- [Balkanski, Rubinstein, Singer] [Ene, Nguyen] $(1-\frac{1}{e}-\varepsilon)$ w/ $O\left(\frac{\log n}{\mathrm{poly}(\varepsilon)}\right)$ rounds
- [Chekuri, Q.]: for general packing constraints
  - different (simpler?) algorithm for cardinality constraints ("parallel-greedy") via multilinear extension of f
  - Cardinality ⇒ Knapsack ⇒ general packing via parallel multiplicative weight updates

multilinear extension of $f: 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$

$F: [0,1]^{\mathcal{N}} \longrightarrow \mathbb{R}_{\geq 0}$ defined by

$F(x) = \mathbb{E}[f(S)]$, where $\left\{ \begin{array}{l} S \text{ samples } e \in \mathcal{N} \\ \text{independently with} \\ \text{probability } x_e \end{array} \right\}$

- multilinear, monotone
- monotone-concave: for $x \in [0,1]^{\mathcal{N}}$, $v \in \mathbb{R}^{\mathcal{N}}_{\geq 0}$, $\delta > 0$

  $F(x + \delta v)$ is concave in δ
- concentrated $\Rightarrow$ easy to estimate, round

Continuous Greedy

[Calinescu, Chekuri, Pál, Vondrák]

1. $x \leftarrow 0$
2. continuously from $t=0$ to k
   a. $v \leftarrow \underset{v \in P}{\operatorname{argmax}} \langle F'(x), v \rangle$

      where $P = \{ v \in [0,1]^N : \langle v, \mathbb{1} \rangle \leq 1 \}$

   b. $\frac{dx}{dt} = v$
3. return x

- $F'(x)$ = continuous analogue of marginal values
- greedily select fractional point v w/ maximum marginal value
- easy to discretize

- works for any downward closed polytope (e.g. matroids)

Continuous greedy in parallel

1. $x \leftarrow 0$

2. continuously from $t=0$ to k

   a. $\frac{dx}{dt} = \frac{S}{|S|}$, where $S = \{e : F'_e(x) \geq (1-\varepsilon) \max_d F'_d(x)\}$

3. return x

- S gathers <u>all</u> $(1-\varepsilon)$-OPT points in polytope

- increase x along all of S uniformly (instead of just the best point).

GOAL: discretize continuous algo to have low depth

Parallel greedy

1. $x \leftarrow \mathbb{0}$, $\lambda \leftarrow OPT$
2. while $\langle x, \mathbb{1} \rangle \le k$ and $\lambda \ge e^{-1} OPT$
   a. $S = \{ e \in N : F_e'(x) \ge (1-\varepsilon)\lambda / k \}$
   b. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
   c. $\lambda \leftarrow (1-\varepsilon)\lambda$
3. return x

- λ = upper bound on margin of any competing set
- S = {good coordinates w/r/t threshold λ}
- $\delta > 0$ chosen large as possible s.t. S remains $(1-O(\varepsilon))$-good: max δ s.t. $F_x(x+\delta S) \ge (1-\varepsilon)^2 \frac{\delta |S|}{k} \lambda$

Parallel greedy

1. $x \leftarrow \mathbb{0}$, $\lambda \leftarrow \mathrm{OPT}$
2. while $\langle x, \mathbb{1} \rangle \leq k$ and $\lambda \geq e^{-1}\mathrm{OPT}$
   a. $S = \{e \in N : F_e'(x) \geq (1-\varepsilon)\lambda / k\}$
   b. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
   c. $\lambda \leftarrow (1-\varepsilon)\lambda$
3. return x

δ is:

- small enough to keep directional deriv $\langle F'(x+\delta S), S \rangle$ good, $\Rightarrow$ continuous greedy
- big enough to decrease directional deriv enough that ε-fraction of S drops out

Parallel greedy

1. $x \leftarrow \mathbb{0}$, $\lambda \leftarrow \mathrm{OPT}$
2. while $\langle x, \mathbb{1} \rangle \leq k$ and $\lambda \geq e^{-1}\mathrm{OPT}$
   a. $S = \{e \in N : F_e'(x) \geq (1-\varepsilon)\lambda / k\}$
   b. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
   c. $\lambda \leftarrow (1-\varepsilon)\lambda$
3. return x

"primal-dual" approach

- continuous greedy (along S) ensures $(1-\frac{1}{e})$ APX
- greedy step size drives down margins and limits # iterations

Randomized-Parallel-Greedy

("combinatorial version")

1. $Q \leftarrow \emptyset$, $\lambda \leftarrow OPT$
2. while $|Q| \leq k$ and $\lambda \geq e^{-1} OPT$
   a. $S = \{e \in \mathcal{N} : f_Q(e) \geq (1-\varepsilon)\lambda/k\}$
   b. while $S \neq \emptyset$
      (i) $Q \leftarrow Q \cup R$ for R sampling δS w/r/t greedy step size $\delta > 0$
   c. $\lambda \leftarrow (1-\varepsilon)\lambda$
3. return Q

---

maintain discrete set Q by immediately rounding δS in each iteration

Conclusion

- Recent interest in parallel submodular max

- [continuous greedy]
  + [bulk update]
  + [greedy step size]

  parallel greedy w/ low depth

  - easily discretized
  - extends to general packing via MWU

- Recent results (e.g. [Ene, Nguyen, Vladu] [Fahrback, Mirrokni, Zadimoghaddam]) improve complexity, extend to non-monotone setting.

THANKS