Deterministic (1/2 + ε)-Approximation for Submodular Maximization over a Matroid

We are given a matroid on a set N and there is a function $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$.

This function has two properties:

1) Monotone (value in a set is at least the value in all subsets)
2) Submodular, means: f(S + u) - f(S) <= f(T + u) - f(T) if S is a subset of T (i.e. some generalization of a convex function)

We want to find the maximum value of the function for an independent set of a matroid.

Results

We will introduce a 0.5008 approximation, compared to the previous known 0.5.

This algorithm is more easy when it is probabilistic, but it is also possible to derandomize.

Split Algorithm

Algorithm 1: $\texttt{Split}(f, \mathcal{M}, p)$

Initialize: $A_0 \leftarrow \varnothing$, $B_0 \leftarrow \varnothing$.

for $i = 1$ **to** k **do**

  Let $u_i^A = \arg\max_{u \in \mathcal{M}/(A_{i-1} \cup B_{i-1})} \{f(u \mid A_{i-1})\}$.

  Let $u_i^B = \arg\max_{u \in \mathcal{M}/(A_{i-1} \cup B_{i-1})} \{f(u \mid B_{i-1})\}$.

  if $p \cdot f(u_i^A \mid A_{i-1}) \geq (1-p) \cdot f(u_i^B \mid B_{i-1})$ **then**

    $A_i \leftarrow A_{i-1} + u_i^A$.

  else

    $B_i \leftarrow B_{i-1} + u_i^B$.

return (A_k, B_k).

What is it for

Observation 3.1. *The output sets A_k and B_k of Algorithm 1 are disjoint, and their union is a base of $\mathcal{M}$.*

Our next objective is to lower bound the values of the output sets of Algorithm 1.

Lemma 3.2. *Let T be a base of $\mathcal{M}$ and $\frac{1}{5} \leq \beta \leq \frac{4}{5}$, then for $p = \frac{\beta}{\beta + \sqrt{(1-\beta)\beta}}$, Algorithm 1 satisfies*

$$\beta \cdot f(A_k) + (1-\beta) \cdot f(B_k) \geq \frac{2}{3}\left(1 - \sqrt{(1-\beta)\beta}\right) \cdot f(T) \ .$$

Proof: straightforward counting, omitted

Lemma 2.2 (Proved by [14, 21]). *Given two bases B_1 and B_2 of a matroid $\mathcal{M}$, and a partition $B_1 = X_1 \mathbin{\dot\cup} Y_1$, there is a partition $B_2 = X_2 \mathbin{\dot\cup} Y_2$ such that $X_1 \mathbin{\dot\cup} Y_2$ and $X_2 \mathbin{\dot\cup} Y_1$ are both bases of $\mathcal{M}$.*

Lemma 3.3. *For every base T of $\mathcal{M}$, there exists a partition of T into two disjoint sets $T_A \mathbin{\dot\cup} T_B$ such that*

- *$A_k \mathbin{\dot\cup} T_A$ and $B_k \mathbin{\dot\cup} T_B$ are both bases of $\mathcal{M}$.*
- *$f(A_k) + f(A_k \cup T_A) \geq f(T)$ and $f(B_k) + f(B_k \cup T_B) \geq f(T)$.*

Residual Random Greedy

Algorithm 2: Residual Random Greedy – RRGreedy$(f, \mathcal{M})$

Initialize: $A_0 \leftarrow \varnothing$.

for $i = 1$ **to** k **do**

  Let M_i be a base of $\mathcal{M}/A_{i-1}$ maximizing $\sum_{u \in M_i} f(u \mid A_{i-1})$.

  Let $A_i \leftarrow A_{i-1} + u_i$, where u_i is a uniformly random element from M_i.

Return A_k.

Proposition 3.10. *Given bases T_1 and T_2 of $\mathcal{M}$, the output set A of Algorithm 2 obeys*

1. $\mathbb{E}[f(A)] \geq f(T_1)/2$.
2. $3\mathbb{E}[f(A)] \geq (1 + g(x)) \cdot f(T_1) + (1 - x) \cdot f(T_2 \mid T_1)$ *for every* $x \in [0, 1]$.

The best friend of our algorithm is this proposition, because here we get a comparison with some value involving other base! And that makes clear why the following algorithm makes at least some sense.

Main Algorithm

[redacted]

Algorithm 3: Matroid Split and Grow$(f, \mathcal{M})$

$(A_1, B_1) \leftarrow \texttt{Split}(f, \mathcal{M}, p)$.
$A_2 \leftarrow \mathsf{RRGreedy}(f(\cdot \mid A_1), \mathcal{M}/A_1)$.
$B_2 \leftarrow \mathsf{RRGreedy}(f(\cdot \mid B_1), \mathcal{M}/B_1)$.
Return the better solution out of $A = (A_1 \cupdot A_2)$ and $B = (B_1 \cupdot B_2)$.

[redacted]

Proposition 4.3. *The approximation ratio of Algorithm 3 is at least* 0.5008.

Proposition 4.3. *The approximation ratio of Algorithm 3 is at least* 0.5008.

Proof. Let

$$\beta = \frac{2 - x - 2g(x)}{(2 - x - 2g(x)) + 2(1 - x)} = \frac{2 - x - 2g(x)}{4 - 3x - 2g(x)} \enspace .$$

Plugging this value of β into the guarantee of Lemma 3.2 for $T = OPT$, and choosing the value of p accordingly, we get

$$(2 - x - 2g(x)) \cdot f(A_1) + 2(1 - x)f(B_1) \geq (4 - 3x - 2g(x)) \cdot w(\beta) \cdot f(OPT) \enspace ,$$

where $w(\beta) \triangleq \frac{2}{3}\left(1 - \sqrt{(1 - \beta)\beta}\right)$. Combining this inequality with the guarantee of Lemma 4.2, we get

$$\begin{aligned}\max\{\mathbb{E}[f(A)], \mathbb{E}[f(B)]\} &\geq \frac{3\mathbb{E}[f(A)] + 2(1 - x) \cdot \mathbb{E}[f(B)]}{5 - 2x} \\ &\geq \frac{1 + g(x) + (4 - 3x - 2g(x)) \cdot w(\beta)}{5 - 2x} \cdot f(OPT) \enspace .\end{aligned}$$

Setting $x = 0.9$, the coefficient of $f(OPT)$ in the last inequality becomes larger than 0.5008. Moreover, it can be verified that for this value of x, $\beta \approx 0.35$ which is in the range $[\frac{1}{5}, \frac{4}{5}]$, as required by Lemma 3.2. $\square$

Derandomization

Algorithm 4: Residual Parallel Greedy – RPGreedy$(f, \mathcal{M}, B)$

Initialize: $A_0^j \leftarrow \varnothing$ and $B_0^j \leftarrow B$ for every $j = 1, \dots, k$.
for $i = 1$ **to** k **do**
    For every $j = 1, \dots, k$, let M_i^j be a base of $\mathcal{M}/A_{i-1}^j$ maximizing $\sum_{u \in M_i^j} f(u \mid A_{i-1}^j)$.
    Construct a weighted bipartite (multi-)graph $G_i = (V_L, V_R, E, w)$ as follows.

- $V_L \triangleq B$ and $V_R \triangleq \{1, \dots, k\}$.
- For each $u \in M_i^j$ and $v \in B$, add an edge $e = (v, j)$ with weight $w_e = f(u \mid A_{i-1}^j)$ if
    - $v \in B_{i-1}^j$, and $(A_{i-1}^j + u) \cup (B_{i-1}^j - v)$ is a base of $\mathcal{M}$.
    - $f(u \mid A_{i-1}^j) \geq f(v \mid A_{i-1}^j)$.

    Find a maximum weight perfect matching R_i in G_i.
    for *every* $j = 1$ **to** k **do**
        Let $e = (v_i^j, j)$ be the single edge in the matching R_i which hits j, and let $u_i^j \in M_i^j$ be the element that corresponds to this edge.
        Set $A_i^j \leftarrow A_{i-1}^j + u_i^j$ and $B_i^j \leftarrow B_{i-1}^j - v_i^j$.
return *the best set out of* $A_k^1, A_k^2, \dots, A_k^k$.

Algorithm 5: Matroid Split and Grow - Deterministic$(f, \mathcal{M})$

$(A_1, B_1) \leftarrow \texttt{Split}(f, \mathcal{M}, p)$.

$A_2 \leftarrow \mathsf{RPGreedy}(f(\cdot \mid A_1), \mathcal{M}/A_1, B_1)$.

$B_2 \leftarrow \mathsf{RPGreedy}(f(\cdot \mid B_1), \mathcal{M}/B_1, A_1)$.

Return the better solution out of $A = (A_1 \mathbin{\dot\cup} A_2)$ and $B = (B_1 \mathbin{\dot\cup} B_2)$.