236621 – Algorithms for Submodular Optimization

Notes of lectures by Roy Schwartz
written by Evgenii Zheltonozhskii

June 17, 2019

Contents

1 Introduction

We are looking on $f : 2^N \to \mathbb{R}$ for some set $N = \{1, \dots n\}$

Definition 1.1. f is submodular if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \tag{1.1}$$

Definition 1.2. Return of u wrt A is $f(A \cup \{u\}) - f(A)$

Definition 1.3 (Diminishing returns)**.** f has diminishing returns if for $A \subseteq B$

$$f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B) \tag{1.2}$$

Proposition 1.1. f is submodular iff f has diminishing returns

Proof. $\Rightarrow$:
Let $A \subseteq B \subseteq N$ and $u \notin B$. Lets use submodularity property on $A \cup \{u\}$ and B:

$$f(A \cup \{u\}) + f(B) \geq f(A \cup \{u\} \cup B) + f((A \cup \{u\}) \cap B) = f(B \cup \{u\}) + f(A) \tag{1.3}$$

Thus

$$f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B) \tag{1.4}$$

□

$\Leftarrow$:
We'll proof by induction over $|A \cup B| - |A \cap B|$, i.e., size of symmetric difference.
Basis: $|A \cup B| - |A \cap B| = 0$, then $A = B$, and then submodular property is fulfilled.
Step: assume $|A \cup B| - |A \cap B| = k$. WLOG let $u \in A$ such that $u \notin B$.

$$f(A) + f(B) = f(A) - f(A \setminus \{u\}) + f(A \setminus \{u\}) + f(B) \geq \tag{1.5}$$
$$\geq f(A) - f(A \setminus \{u\}) + f(A \setminus \{u\} \cup B) + f(A \setminus \{u\} \cap B) \geq \tag{1.6}$$
$$\geq f(A \cup B) - f(A \cup B \setminus \{u\}) + f(A \cup B \setminus \{u\}) + f(A \cap B) = f(A \cup B) + f(A \cap B) \tag{1.7}$$

Definition 1.4 (Monotonous function). f is non-decreasing monotonous if $\forall A \subseteq B \subseteq N$, $f(A) \leq f(B)$.

Definition 1.5 (Symmetric function). f is symmetric if $\forall S \subseteq N$, $f(S) \leq f(N \setminus S)$.

Definition 1.6 (Normalized function). f is normalized if $f(\emptyset) = 0$.

Examples

Linear function $\forall n \in N$ exists weight w_n and

$$f(S) = \sum_{u \in S} w_u + b \tag{1.8}$$

Such f is submodular.

Budget additive function (clipped linear function) $\forall n \in N$ exists weight w_n and

$$f(S) = \min\left\{\sum_{u \in S} w_u, b\right\} \tag{1.9}$$

Such f is submodular.

Coverage function Given set X and n subsets $S_1, S_2, \ldots, S_n \subset X$ define

$$f(S) = \left|\bigcup_{i \in S} S_i\right| \tag{1.10}$$

This f is obviously submodular.

Graph cuts Let $G + (V, E)$ be a graph and $w : E \to \mathbb{R}^+$ weights of edges. Given a cut $S \subseteq V$ define $\delta(S)$ to be sum of weights of all edges going through the cut. $\delta : 2^V \to \mathbb{R}^+$ is submodular, normalized, and symmetric.

Rank function Let $v_1, \ldots, v_n \in \mathbb{R}^d$ vectors, and

$$f(S) = \mathrm{rank}(S) = \dim \mathrm{span}\big(\{v_i | i \in S\}\big) \tag{1.11}$$

2 Submodular optimization

Given world N, submodular function $f : 2^N \to \mathbb{R}^+$, and a family of feasible solutions $\mathcal{I} \subseteq 2^N$

$$\max f(S) \tag{2.1}$$
$$\text{s.t. } S \in \mathcal{I} \tag{2.2}$$

Note Most of submodular functions (except for logarithm of determinant of submatrix) are nonnegative. We use the condition to have properly defined multiplicative approximation.

Note How f is given in input? Obviously, not as a list of values, since it's exponential in $|N|$. Thus we represent f with black box, and same applies for constraints. Usually, constraints are simple.

2.1 Examples of submodular optimization problems

Example f is submodular and there are no constraints. It generalizes MAX-CUT, MAX-DICUT

Example f is submodular and there is size constraint:

$$\max f(S) \tag{2.3}$$
$$\text{s.t. } |S| \leq k \tag{2.4}$$

. It generalizes MAX-K-COVER.

Submodular welfare Given k players and n goods, each player has submodular, monotone, non-negative value function T_i over subsets of N. The goal is to maximize total value by partitioning the goods.

3 Maximization of the submodular function with cardinality constraints

$$\max f(S) \tag{3.1}$$
$$\text{s.t. } |S| \leq k \tag{3.2}$$

Greedy algorithm If f is monotonic, greedy algorithm is an optimal approximating algorithm.

Algorithm 3.1 Nemhauser-Wolsey-Fisher

procedure GREEDY(N)
  $A \leftarrow \emptyset$
  for $i = 1$ to k **do**
    Let $u_i \in N$ maximize $f(A_{i-1} \cup \{u_i\}) - f(A_{i-1})$
    $A_i \leftarrow A_{i-1} \cup \{u_i\}$
  end for
  return A_k
end procedure

Lemma 3.1. For submodular $f : 2^N \to \mathbb{R}_+$,

$$f(A \cup B) - f(A) \leq \sum_{b_i \in B} f(A \cup \{b_i\}) - f(A) \tag{3.3}$$

Proof.

$$f(A \cup B) - f(A) = \sum_i f(A \cup \{b_1, \ldots b_{i-1}\} \cup \{b_i\}) - f(A \cup \{b_1, \ldots b_{i-1}\}) \leq \sum_i f(A \cup \{b_i\}) - f(A) \tag{3.4}$$

□

Proposition 3.2 (Nemhauser et al. [1978]). Algorithm 3.1 is $1 - \frac{1}{e}$ optimal.

Proof. For optimal set S^*

$$f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) \geq \max_{u \in S^*} \{f(A_{i-1} \cup \{u\}) - f(A_{i-1})\} \geq \frac{1}{k} \sum_{u \in S^*} [f(A_{i-1} \cup \{u\}) - f(A_{i-1})] \geq \tag{3.5}$$
$$\geq \frac{1}{k}\Big(f(A_{i-1} \cup S^*) - f(A_{i-1})\Big) \geq \frac{1}{k}\Big[f(S^*) - f(A_{i-1})\Big] \tag{3.6}$$

We got a recursion equation:

$$f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) \geq \frac{1}{k}\Big[f(S^*) - f(A_{i-1})\Big] \tag{3.7}$$

We can solve the recursion and acquire

$$f(A_k) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(S^*) + \left(1 - \frac{1}{k}\right)^k f(A_0) \geq \left(1 - \frac{1}{e}\right) f(S^*) \tag{3.8}$$

□

Theorem 3.3 (Nemhauser and Wolsey [1978]). For all constant $\epsilon > 0$ each algorithm acquiring $1 - \frac{1}{e} + \epsilon$ requires exponential number of requests to value oracle.

Theorem 3.4 (Feige [1998]). For MAX-K-COVER all constant $\epsilon > 0$ each algorithm acquiring $1 - \frac{1}{e} + \epsilon$ requires exponential number of requests to value oracle unless $P = NP$.

Note Runtime of Algorithm 3.1 on the preceding page is $\mathcal{O}(nk)$. It is possible to acquire $\mathcal{O}\big(n \lg(\frac{1}{\epsilon})\big)$ runtime and $1 - \frac{1}{e} - \epsilon$ optimality by looking on some subset of N at each step instead of the whole set.

3.1 Non-monotonic functions

What happens if f is not monotonic? First of all, does greed algorithm work? Not only it is not optimal approximation, it can be as bad as $\frac{2}{N}$. However, it can be fixed. The idea is to randomize algorithm to prevent it from "bad" choices.

Algorithm 3.2

```
procedure RANDOMIZED GREEDY(N)
    A ← ∅
    for i = 1 to k do
        M_i ← arg max_{B⊆N : |B|≤k} Σ_{u∈B} f(A_{i-1} ∪ {u}) − f(A_{i-1})
        A_i ← { A_{i-1} ∪ {u}   ∀u ∈ M_i with P = 1/k
              { A_{i-1}             with P = 1 − |M_i|/k
    end for
    return A_k
end procedure
```

Randomized greedy algorithm

Theorem 3.5 (Buchbinder et al. [2014]). In monotonic case, Algorithm 3.2 is $1 - \frac{1}{e}$ optimal in expectation.

Proof. Take a look at i^{th} iteration and condition on previous iterations, denote a chosen element from M_i as u_i:

$$\mathbb{E}\Big[f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) | A_{i-1}\Big] = \frac{1}{k} \sum_{u_i \in M_i} f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) \geq \frac{1}{k}(f(A_{i-1} \cup S^*) - f(A_{i-1})) \geq \tag{3.9}$$

$$\geq \frac{1}{k}(f(S^*) - f(A_{i-1})) \geq \frac{1}{k}(f(S^*) - f(A_{i-1})) \tag{3.10}$$

If the inequality is right for any A_{i-1} it is right, from tower property, in expectation over A_{i-1}:

$$\mathbb{E}\Big[f(A_{i-1} \cup \{u_i\}) - f(A_{i-1})\Big] \geq \frac{1}{k}(f(S^*) - \mathbb{E}[f(A_{i-1})]) \tag{3.11}$$

And thus we can once again solve the recurrence and acquire same result as in Proposition 3.2 on the previous page. □

Lemma 3.6. Given set $B \subseteq N$ such that

$$\forall u \in N \quad P(u \in B) \leq p \tag{3.12}$$

then

$$\mathbb{E}[f(B)] \geq (1 - p) f(\emptyset) \tag{3.13}$$

Proof. WLoG $p(u_1 \in B) \geq p(u_2 \in B) \geq \cdots \geq p(u_n \in B)$. Denote

$$X_i = \mathbb{1}_{u_i \in B} N_i = \bigcup_{j=1}^{i} u_j \tag{3.14}$$

We can then rewrite

$$f(B) = f(N_0) + \sum_{i=1}^{n} X_i \Big(f(B \cap N_i) - f(B \cap N_{i-1}) \Big) \tag{3.15}$$

$$\mathbb{E}[f(B)] = f(N_0) + \sum_{i=1}^{n} \mathbb{E}\Big[X_i \Big(f(B \cap N_i) - f(B \cap N_{i-1}) \Big) \Big] \geq \tag{3.16}$$

$$\geq f(N_0) + \sum_{i=1}^{n} \Big(f(N_i) - f(N_{i-1}) \Big) \mathbb{E}[X_i] = f(N_0) + \sum_{i=1}^{n} \Big(f(N_i) - f(N_{i-1}) \Big) p_i = \tag{3.17}$$

$$= f(N_0)(1 - p_1) + \sum_{i=1}^{n} f(N_i) \underbrace{(p_i - p_{i+1})}_{\leq 0} \geq f(N_0)(1 - p_1) \geq f(\emptyset)(1 - p) \tag{3.18}$$

□

Lemma 3.7. Given set $A \subseteq N$ and set $B \subseteq N$ such that

$$\forall u \in N \quad P(u \in B) \leq p \tag{3.19}$$

$$\mathbb{E}[f(A \cup B)] \geq (1 - p) f(A) \tag{3.20}$$

Proof. Define

$$g_A(S) = f(A \cup S) \tag{3.21}$$

Obviously, g_A is also submodular (from diminishing returns). Then, from Lemma 3.6 on the preceding page

$$\mathbb{E}[f(A \cup B)] = \mathbb{E}[g(B)] \geq (1 - p) g(\emptyset) = (1 - p) f(A) \tag{3.22}$$

□

Theorem 3.8 (Buchbinder et al. [2014])**.** In non-monotonic case, Algorithm 3.2 on the previous page is $\frac{1}{e}$ optimal in expectation.

Proof. Similarly to monotonic case, take a look at i^{th} iteration and condition on previous iterations, denote a chosen element from M_i as u_i:

$$\mathbb{E}\Big[f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) | A_{i-1} \Big] = \frac{1}{k} \sum_{u_i \in M_i} f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) \geq \frac{1}{k} (f(A_{i-1} \cup S^*) - f(A_{i-1})) \tag{3.23}$$

Since

$$P(u \in A_{i-1}) \leq 1 - \left(1 - \frac{1}{k}\right)^{i-1} \tag{3.24}$$

from Lemma 3.7

$$\mathbb{E}[f(A_{i-1} \cup S^*)] \geq \left(1 - \frac{1}{k}\right)^{i-1} f(S^*) \tag{3.25}$$

Thus, taking expectation

$$\mathbb{E}\Big[f(A_{i-1} \cup \{u_i\}) - f(A_{i-1}) \Big] \geq \frac{1}{k} (f(A_{i-1} \cup S^*) - f(A_{i-1})) \geq \frac{1}{k} \left[\left(1 - \frac{1}{k}\right)^{i-1} f(S^*) - \mathbb{E}\Big[f(A_{i-1}) \Big] \right] \tag{3.26}$$

$$\mathbb{E}\Big[f(A_i)\Big] \geq \frac{1}{k}(f(A_{i-1} \cup S^*) - f(A_{i-1})) \geq \frac{1}{k}\left[\left(1 - \frac{1}{k}\right)^{i-1} f(S^*) - \mathbb{E}\Big[f(A_{i-1})\Big]\right] \tag{3.27}$$

Solving the recurrence we get

$$\mathbb{E}[f(A_i)] \geq \frac{i}{k}\left(1 - \frac{1}{k}\right)^{k-1} f(S^*) \geq \frac{1}{e} f(S^*) \tag{3.28}$$

i.e.,

$$\mathbb{E}[f(A_k)] \geq \left(1 - \frac{1}{k}\right)^{k-1} f(S^*) \geq \frac{1}{e} f(S^*) \tag{3.29}$$

□

Note Algorithm 3.2 on page 4 is not optimal. In addition, the upper bound of the best approximation is 0.49.

Runtime Runtime of Algorithm 3.2 on page 4 is $\mathcal{O}(nk)$.

4 Maximization of the submodular function without constraints

$$\max f(S) \tag{4.1}$$

Examples

- MAX-CUT
- MAX-DIRECTED-CUT
- Max Facility Location
- MAX-SAT (with all literals in a clause having same sign).

Proposition 4.1 ([Feige et al., 2011]). Algorithm which choose random solution as following: $u \in S$ with probability $\frac{1}{2}$ independently, is $\frac{1}{4}$ approximation in expectation:

$$\mathbb{E}[f(S)] \geq \frac{1}{4} f(S^*) \tag{4.2}$$

Proposition 4.2 ([Feige et al., 2011]). If f is symmetric, the same algorithm is $\frac{1}{2}$ approximation in expectation:

$$\mathbb{E}[f(S)] \geq \frac{1}{2} f(S^*) \tag{4.3}$$

Proposition 4.3 ([Feige et al., 2011]). For any constant $\epsilon > 0$ it is impossible to acquire $\left(\frac{1}{2} + \epsilon\right)$ approximation in polynomial time, even in symmetric case.

Note that for $\bar{f}(S) = f(\bar{S})$, we can use the same oracle. So a "conjugate" algorithm would be start from N and drop elements from it.

Proposition 4.4. It's impossible that both $f(X_{i-1} \cup \{u_i\}) - f(X_{i-1}) < 0$ and $f(Y_{i-1} \setminus \{u_i\}) - f(Y_i) < 0$.

Proof. From diminishing returns:

$$f(X_{i-1} \cup \{u_i\}) - f(X_{i-1}) \geq f(Y_i) - f(Y_{i-1} \setminus \{u_i\}) \tag{4.4}$$
$$f(X_{i-1} \cup \{u_i\}) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1} \setminus \{u_i\}) \geq 0 \tag{4.5}$$

Thus at least one of $f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})$ and $f(Y_{i-1} \setminus \{u_i\}) - f(Y_i)$ is greater than 0. □

Algorithm 4.1

procedure DOUBLE GREEDY(N)
  $X \leftarrow \emptyset$, $Y \leftarrow N$
  for $i = 1$ to n **do**
    $a_i = f(X_{i-1} \cup \{u_i\}) - f(X_i)$
    $b_i = f(Y_{i-1} \setminus \{u_i\}) - f(Y_i)$
    if $a_i > b_i$ **then**
      $X_i \leftarrow X_{i-1} \cup \{u_i\}$
      $Y_i \leftarrow Y_{i-1}$
    else
      $X_i \leftarrow X_{i-1}$
      $Y_i \leftarrow Y_{i-1} \setminus \{u_i\}$
    end if
  end for
  return X_N
end procedure

Algorithm 4.2

procedure RANDOMIZED DOUBLE GREEDY(N)
  $X \leftarrow \emptyset$, $Y \leftarrow N$
  for $i = 1$ to n **do**
    $a_i = \max\{0, f(X_{i-1} \cup \{u_i\}) - f(X_i)\}$
    $b_i = \max\{0, f(Y_{i-1} \setminus \{u_i\}) - f(Y_i)\}$
    $(X_i, Y_i) \leftarrow \begin{cases} (X_{i-1} \cup \{u_i\}, Y_i) & \text{with } P = \frac{a_i}{a_i+b_i} \\ (X_{i-1}, Y_{i-1} \setminus \{u_i\}) & \text{with } P = \frac{b_i}{a_i+b_i} \end{cases}$
  end for
  return X_N
end procedure

Lemma 4.5. Let S^* be an optimal solution and

$$S_i^* = S^* \cup X_i \cap Y_i \tag{4.6}$$

i.e., optimal solution to which we add everything Algorithm 4.1 added and drop everything it dropped. For all i:

$$f(S_{i-1}^*) - f(S_i^*) \leq f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1}) \tag{4.7}$$

Lemma 4.6. Let S^* be an optimal solution and

$$S_i^* = S^* \cup X_i \cap Y_i \tag{4.8}$$

i.e., optimal solution to which we add everything Algorithm 4.2 added and drop everything it dropped. For all i:

$$\mathbb{E}\Big[f(S_{i-1}^*) - f(S_i^*)\Big] \leq \frac{1}{2}\mathbb{E}\Big[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})\Big] \tag{4.9}$$

Proof. Take a look at i^{th} iteration and condition on previous iterations:

$$\mathbb{E}\Big[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1}) \Big| X_{i-1}, Y_{i-1}\Big] = \tag{4.10}$$

$$= \frac{a_i}{a_i + b_i} \underbrace{\left(f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})\right)}_{=a_i \text{ if } a_i \neq 0} + \frac{b_i}{a_i + b_i} \underbrace{\left(f(Y_{i-1} \cup \{u_i\}) - f(Y_{i-1})\right)}_{=b_i \text{ if } b_i \neq 0} = \frac{a_i^2 + b_i^2}{a_i + b_i} \tag{4.11}$$

Now divide into two cases: $u_i \in S^*$ and $u_i \notin S^*$.

- If $u_i \notin S^*$, in particular, $u_i \notin S^*_{i-1}$:

$$\mathbb{E}\big[f(S^*_{i-1}) - f(S^*_i)\big] = \frac{a_i}{a_i+b_i}\big(f(S^*_{i-1}) - f(S^*_{i-1} \cup \{u_i\})\big) \overset{S^*_{i-1}\subseteq Y_{i-1}\setminus\{u_i\}}{\leq} \tag{4.12}$$

$$\leq \frac{a_i}{a_i+b_i}\big(f(Y^*_{i-1} \setminus \{u_i\}) - f(Y^*_{i-1})\big) \leq \frac{a_i b_i}{a_i+b_i} \tag{4.13}$$

- If $u_i \in S^*$, in particular, $u_i \in S^*_{i-1}$:

$$\mathbb{E}\big[f(S^*_{i-1}) - f(S^*_i)\big] = \frac{b_i}{a_i+b_i}\big(f(S^*_{i-1}) - f(S^*_{i-1} \setminus \{u_i\})\big) \overset{X_{i-1}\subseteq S^*_{i-1}\setminus\{u_i\}}{\leq} \tag{4.14}$$

$$\leq \frac{b_i}{a_i+b_i}\big(f(X^*_{i-1} \cup \{u_i\}) - f(X^*_{i-1})\big) \leq \frac{a_i b_i}{a_i+b_i} \tag{4.15}$$

And since $a_i^2 - 2a_ib_i + b_i^2 = (a_i - b_i)^2 \geq 0$ (and by tower property), we get the required.

□

Theorem 4.7 (Buchbinder et al. [2015]). Algorithm 4.2 on the previous page is $\frac{1}{2}$ approximation in expectation.

Proof. Denote

$$S_{alg} = S^*_n = X_n = Y_n \tag{4.16}$$

Then

$$\mathbb{E}\Big[f(S^*_0) - f(S^*_n)\Big] \leq \frac{1}{2}\mathbb{E}\Big[f(X_n) - f(X_0) + f(Y_n) - f(Y_0)\Big] \tag{4.17}$$

$$\mathbb{E}\Big[f(S^*) - f(S_{alg})\Big] \leq \frac{1}{2}\mathbb{E}\Big[2S_{alg} - f(X_0) - f(Y_0)\Big] \overset{f(S)\geq 0}{\leq} \mathbb{E}\Big[S_{alg}\Big] \tag{4.18}$$

Thus

$$\mathbb{E}[S_{alg}] \geq \frac{1}{2}\mathbb{E}[f(S^*)] \tag{4.19}$$

□

Collary 4.7.1. Algorithm 4.1 on the preceding page is $\frac{1}{3}$ approximation.

Note Algorithms 4.1 and 4.2 on the previous page run in $\mathcal{O}(N)$ time.

5 Knapsack constraints

Let each element of set have price c_i and budget B, then

$$\max f(S) \tag{5.1}$$

$$\text{s.t.} \sum_{i\in S} c_i \leq B \tag{5.2}$$

Note that this is generalization of cardinality constraint.

Proposition 5.1 (Khuller et al. [1999]). Algorithm 5.2 on the next page is $\frac{1}{2}\left(1 - \frac{1}{e}\right)$-optimal.

Proposition 5.2. Algorithm 5.2 on the following page is $\left(1 - \frac{1}{\sqrt{e}}\right)$-optimal.

Theorem 5.3 (Khuller et al. [1999], Sviridenko [2004]). If a set of l most dense items in optimal solution S^*, it is possible to get good approximation to the optimal solution.
Enumerating all sets of up to 3 most dense items in optimal solution S^*, we can acquire $1 - \frac{1}{e}$-approximation of optimal solution.
Since cardinality constraint is a particular case of knapsack constraint, this is best polynomial approximation.

Algorithm 5.1

procedure DENSITY GREEDY(N)
  $S \leftarrow \emptyset$
  while $N \neq \emptyset$ **do**
    $x^* \leftarrow \arg\max \left\{ \frac{f(S \cup \{x\}) - f(S)}{c_i} \right\}$
    if $c(S) + c_{x^*} \leq B$ **then**
      $S \leftarrow S \cup \{x^*\}$
    end if
    $N \leftarrow N \setminus \{x^*\}$
  end while
  return S
end procedure

Algorithm 5.2

procedure OPTIMIZED DENSITY GREEDY(N)
  $S_1 \leftarrow$ output of Algorithm 5.1
  $S_2 \leftarrow \left\{ \arg\max_{\substack{i \in N \\ c_i \leq B}} f(i) \right\}$
  return $\arg \max_{S \in \{S_1, S_2\}} f(S)$
end procedure

6 Introduction to matroids

Matroid is a basic concept in combinatorial optimization. It was first defined by Whitney [1935].

Definition 6.1 (matroid)**.** Matroid $\mathcal{M}$ is a pair $(E, \mathcal{I})$. E is is a finite set (called the ground set) and $\mathcal{I} \neq \emptyset$ is a family of subsets of E (called the independent sets) with the following properties:

1. If $Y \in \mathcal{I}$ then for all $X \subseteq Y$, $X \in \mathcal{I}$.
2. If $X, Y \in \mathcal{I}$ and $|Y| > |X|$, then exists $e \in Y \setminus X$, $X \cup \{e\} \int \mathcal{I}$.

Notes All maximal independent sets have same size. Those sets are called basis.

Examples

Uniform manifold

$$\mathcal{M}_k = \left(E, \left\{X \subseteq E \middle| |X| \leq k\right\}\right) \tag{6.1}$$

Linear manifold Let $A \in \mathbb{R}^{m \times n}$ be a matrix. Let E be a set of columns of A. The set $X \subseteq E$ is independent if its elements are independent. Alternatively, for sub-matrix A_X consisting of columns of A:

$$\mathcal{I} = \{X \subset E | \operatorname{rank}(A_x) = |X|\} \tag{6.2}$$

Graphic matroids Let $G = (V_G, E_G)$ be a graph, $E = E_G$ and

$$\mathcal{I} = \{X \subseteq E_G | X \text{ is forest}\} \tag{6.3}$$

Proposition 6.1. $M = (E_G, \mathcal{I})$ is matroid.

The basis is then spanning trees (or forests if graph is not connected).

Partition matroid For a set E let $E_1, \dots E_k$ be some partition of E. Then

$$\mathcal{I} = \left\{X \subseteq E \middle| \forall i = 1..k \; |X| \cup E_i \leq 1\right\} \tag{6.4}$$

Proposition 6.2. $M = (E, \mathcal{I})$ is matroid.

Note that partition matroid encodes constraints of submodular welfare problem.
Constraint of matching in the bipartite graph can be defined as intersection of two partition matroids.

Definition 6.2 (Circuit)**.** Circuit in matroid $M = (E, \mathcal{I})$ is a dependent set X ($X \notin \mathcal{I}$) and for all $x \in X$, $X \setminus \{x\} \in \mathcal{I}$.

Definition 6.3 (Rank function)**.** For matroid $M = (E, \mathcal{I})$ rank function $r : 2^{\mathbb{E}} \to \mathbb{N}$ is defined as

$$r(x) = \max\left\{|Y| \middle| Y \subset X, Y \in \mathcal{I}\right\} \tag{6.5}$$

Definition 6.4 (Rank of matroid)**.** For matroid $M = (E, \mathcal{I})$ rank of matroid is $\mathrm{rank}(E)$.

Proposition 6.3. Rank of matroid is submodular function.

Algorithm 6.1

```
procedure GREEDY(E, I)
    S ← ∅
    for e ∈ E do
        if S ∪ {x} ∈ 𝓘 then
            S ← S ∪ {x}
        end if
    end for
    return S
end procedure
```

Proposition 6.4. Algorithm 6.1 returns basis of E.

Proof. Assume S is not a basis and let B be a basis. Exists $x \in B \setminus S$ such that $S \cup \{x\}$ is independent. However, since we have not added x to S, it got to be dependent with S. ☐

Question Given matroid over E (via independence oracle), let weight function $w : E \to \mathbb{R}$ and weight of set be $w(X) = \sum_{x \in X} w(x)$. We want to find independent set (pr basis) of maximal weight.

Algorithm 6.2

```
procedure GREEDY(E, I)
    S ← ∅
    for e ∈ E from heaviest to lightest do
        if S ∪ {x} ∈ 𝓘 then
            S ← S ∪ {x}
        end if
    end for
    return S
end procedure
```

Proposition 6.5. Algorithm 6.2 solves the problem of maximal weight basis.

Proof. We know that for $k = \mathrm{rank}(M)$, the size of the output of algorithm is k and so is size of optimal solution S^*. Lets assume S is not optimal, thus exists i such that $w(e_i^*) > w(e_i)$.
At iteration at which we added e_i to S. At this iteration $|S| = i - 1$. take a look at first i elements of S^*: this is independent set, and from definition of matroid, exists e' such that S is independent with e'. However, $w(e') \geq w(e^* - i) > w(e_i)$, thus we should have added it beforehand. ☐

7 Continuous extensions of submodular functions

Definition 7.1 (Continuous extensions of function)**.** Denote by $\mathbb{1}_S \in \{0,1\}^N$ indicator of S. For $f : 2^N \to \mathbb{R}$ extension of f is $F : [0,1]^N \to \mathbb{E}$ such that for all $S \in N$ $F(\mathbb{1}_S) = f(S)$.

There exist many extensions of submodular functions. In particular, there exist convex and concave extensions of submodular functions.

Main idea For point $x \in [0,1]^N$ define distribution on subsets of N, D_x such that for $R \sim D_x$:

$$P(i \in R) = x_i \tag{7.1}$$

Then we define

$$F(x) = \mathbb{E}_{R \sim D_x}[f(R)] \tag{7.2}$$

Example For all x choose D_x such that $F(x)$ is maximized:

$$f^+(x) = \max_{D_x} \mathbb{E}_{R \sim D_x}[f(R)] \tag{7.3}$$

Similarly, we can choose D_x such that $F(x)$ is minimized:

$$f^-(x) = \min_{D_x} \mathbb{E}_{R \sim D_x}[f(R)] \tag{7.4}$$

Proposition 7.1. f^+ is concave and f^- is convex.

Proof. Let $x, y \in [0,1]^N$ lets show that for

$$z = \lambda x + (1-\lambda)y \tag{7.5}$$

concave property is fulfilled:

$$f^=(z) \geq \lambda f^+(x) + (1-\lambda)f^+(y) \tag{7.6}$$

Let $\{\alpha_S\}_{S \subseteq N}$ be a distribution which defined $f(x)$: $P(R = S) = \alpha_S$, which fulfills:

$$f^+(x) = \mathbb{E}_{R \sim \alpha_S}[f(R)] \tag{7.7}$$

$$\sum_{S: i \in S} \alpha_S = x_i \tag{7.8}$$

Similarly, let $\{\beta_S\}_{S \subseteq N}$ be a distribution defining $f(y)$.
Now, take a look at linear combination of α_S and β_S:

$$P(S) = \lambda \alpha_S + (1-\lambda)\beta_S \tag{7.9}$$

Note that this distribution conserves marginal values of z, since:

$$\sum_{S: i \in S} \lambda \alpha_S + (1-\lambda)\beta_S = \lambda \sum_{S: i \in S} \alpha_S + (1-\lambda) \sum_{S: i \in S} \beta_S = \lambda x_i + (1-\lambda) y_i = z_i \tag{7.10}$$

By definition,

$$f^+(z) \geq \mathbb{E}_{R \sim \lambda \alpha_S + (1-\lambda)\beta_S}[f(R)] = \sum_{S \subseteq N} P(R = S) f(S) = \sum_{S \subseteq N} [\lambda \alpha_S + (1-\lambda)\beta_S] f(S) = \lambda f^+(x) + (1-\lambda) f^+(y) \tag{7.11}$$

□

Proposition 7.2. Evaluating concave extensions of submodular function in some point is NP-hard.

Definition 7.2 (Lovasz extension)**.**

$$f_L(x) = \mathbb{E}_{\theta \sim [0,1]}[f(\{i : x_i \geq \theta\})] \tag{7.12}$$

Theorem 7.3 (Lovasz)**.** $f_L(x) = f^-(x)$ iff f is submodular.

Proof. $\Leftarrow$: Denote $\{\alpha_S\}_{S \subseteq N}$ a distribution defining $f^-(x)$, and out of those the one that maximizes $\sum_{S \subseteq N} \alpha_S |S|^2$.
We'll show that for such α, the sets for which $\alpha_S > 0$ are chain (i.e., a set of sets such that for A, B in the set either $A \subseteq B$ or $B \subseteq A$).
Note that there is unique distribution that conserves marginal values and its support is chain: the Lovasz distribution.
Lets show how we can "fix" the distribution α_S which support is not a chain (uncrossing). Suppose there are A, B such that $\alpha_A \geq \alpha_B > 0$ and $A \not\subseteq B$ and $B \not\subseteq A$. For that, lets reduce the probability of A and B by α_B, and increase probability of $A \cap B$ and $A \cup B$ by α_B.

Does the new distribution conserve marginal values? For all of cases $x \in A \cap B$, $x \in A \setminus B$ and $x \in B \setminus A$, the probability did not change.
What happened to $\mathbb{E}[f(R)]$? From submodularity,

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \tag{7.13}$$

and since we removed LHS and added RHS multiplied same constant, the expectation can not grow.
What happens to $\sum_{S \subseteq N} \alpha_S |S|^2$?

$$|A \cup B|^2 + |A \cap B|^2 = (|A| + |B \setminus A|)^2 + (|B| - |B \setminus A|)^2 = |A|^2 + |B|^2 + 2|B \setminus A|(|A| - |B| + |B \setminus A|) = \tag{7.14}$$
$$= |A|^2 + |B|^2 + 2|B \setminus A| \underbrace{(|A \cup B| - |B|)}_{>0} > |A|^2 + |B|^2 \tag{7.15}$$

Thus, if we choose the set which maximizes $\sum_{S \subseteq N} \alpha_S |S|^2$, there are no two sets A, B such that $A \not\subseteq B$ and $B \not\subseteq A$, i.e., support is the chain.
$\Rightarrow$: □

Definition 7.3 (Multilinear extension)**.**

$$F(x) = \sum_{S \subseteq N} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i) \tag{7.16}$$

i.e., each element is chosen independently.
Proposition 7.4. Let f be monotonous function. Then for $\mathbf{x} \in [0,1]^N$, $\mathbb{R}^n \ni \mathbf{y} > 0$ (coordinate-wise), and $g(t) = F(\mathbf{x} + t\mathbf{y})$, g is monotonous, i.e.,

$$\frac{\partial F}{\partial \mathbf{y}} \geq 0 \tag{7.17}$$

In other words, for $i \in N$

$$\frac{\partial F}{\partial x_i} \geq 0 \tag{7.18}$$

Proposition 7.5. Let f be submodular function. Then for $\mathbf{x} \in [0,1]^N$, $\mathbb{R}^n \ni \mathbf{y} > 0$ (coordinate-wise), and $g(t) = F(\mathbf{x} + t\mathbf{y})$, g is concave, i.e.,

$$\frac{\partial^2 F}{\partial \mathbf{y}^2} \leq 0 \tag{7.19}$$

In other words, for $i, j \in N$

$$\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0 \tag{7.20}$$

8 Matroid constraints

Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid.

$$\max f(S) \tag{8.1}$$
$$\text{s.t. } S \in \mathcal{I} \tag{8.2}$$

Let

$$\mathcal{P}_{\mathcal{M}} = \left\{ z \in [0,1]^N \middle| \forall S \subseteq N \sum_{i \in S} z_i \leq \text{rank}(S) \right\} \tag{8.3}$$

Lemma 8.1. For $\mathbf{x} \in [0,1]^N$

$$\sum_{i \in S} F(\max\{\mathbf{x}, \mathbb{1}_i\}) - F(\mathbf{x}) \geq F(\max\{\mathbf{x}, \mathbb{1}_S\}) - F(\mathbf{x}) \tag{8.4}$$

Algorithm 8.1

procedure CONTINUOUS GREEDY(N)
  $\mathbf{y}(0) \leftarrow \mathbf{0}$
  for $t' \in (0,1)$ **do**
    $\mathbf{x}(t') \leftarrow \arg\max_{\mathbf{x}\in\mathcal{P}_{\mathcal{M}}} \{\mathbf{x}\cdot\boldsymbol{\nabla} F(\mathbf{y}(t'))\}$
    $\frac{\partial \mathbf{y}}{\partial t}(t') \leftarrow \mathbf{x}(t')$
  end for
  return $\mathbf{y}(1)$
end procedure

Proof. Denote by $D_{\mathbf{x}}$ random distribution of taking each element independently with probability x_i, i.e.,

$$F(\mathbf{x}) = \mathbb{E}_{R\sim D_{\mathbf{x}}}[f(R)] \tag{8.5}$$

Then

$$\sum_{i\in S} F(\max\{\mathbf{x},\mathbb{1}_i\}) - F(\mathbf{x}) = \sum_{i\in S}\mathbb{E}_{R\sim D_{\max\{\mathbf{x},\mathbb{1}_i\}}}[f(R)] - \mathbb{E}_{R\sim D_{\mathbf{x}}}[f(R)] = \sum_{i\in S}\mathbb{E}_{R\sim D_{\mathbf{x}}}[f(R\cup\{i\}) - f(R)] = \tag{8.6}$$

$$= \mathbb{E}_{R\sim D_{\mathbf{x}}}\left[\sum_{i\in S} f(R\cup\{i\}) - f(R)\right] \geq \mathbb{E}_{R\sim D_{\mathbf{x}}}[f(R\cup S) - f(R)] = F(\max\{\mathbf{x},\mathbb{1}_S\}) - F(\mathbf{x}) \tag{8.7}$$

□

Theorem 8.2 (Calinescu et al. [2011])**.** For monotonous submodular f,

$$F(\mathbf{y}(1)) \geq \left(1 - \frac{1}{\epsilon}\right) f(S^*) \tag{8.8}$$

where $\mathbf{y}(t)$ is output of Algorithm 8.1:

$$\mathbf{y}(t) = \int_0^t \mathbf{x}(t')\,\mathrm{d}t' \tag{8.9}$$

and S^* is optimal solution of

$$\max f(S) \tag{8.10}$$
$$\text{s.t. } S\in\mathcal{I} \tag{8.11}$$

Proof. Lets bound $\frac{\partial F}{\partial t}$:

$$\frac{\partial F}{\partial t} = \boldsymbol{\nabla} F\cdot\frac{\partial y}{\partial t} = \boldsymbol{\nabla} F(\mathbf{y}(t'))\cdot\mathbf{x}(t') \geq \boldsymbol{\nabla} F(\mathbf{y}(t'))\cdot\mathbb{1}_{S^*} = \sum_{i\in S^*}(\boldsymbol{\nabla} F(\mathbf{y}(t')))_i \overset{F\text{is multilinear}}{\geq} \sum_{i\in S^*}\frac{F(\max\{\mathbf{y},\mathbb{1}_i\}) - F(\mathbf{y})}{1-\mathbf{y}_i} \geq \tag{8.12}$$

$$\geq \sum_{i\in S^*} F(\max\{\mathbf{y},\mathbb{1}_i\}) - F(\mathbf{y}) \geq F(\max\{\mathbf{y},\mathbb{1}_{S^*}\}) - F(\mathbf{y}) \geq f(S^*) - F(\mathbf{y}) \tag{8.13}$$

where max is coordinate-wise maximum. We got

$$F(\mathbf{y}(0)) \geq 0 \tag{8.14}$$
$$\frac{\partial F}{\partial t} \geq f(S^*) - F(\mathbf{y}) \tag{8.15}$$

with solution $F(\mathbf{y}(t)) \geq (1-e^{-t})f(S^*)$, i.e.,

$$F(1) \geq \left(1-\frac{1}{\epsilon}\right) f(S^*) \tag{8.16}$$

□

Algorithm 8.2

```
procedure MEASURED CONTINUOUS GREEDY(N)
    y(0) ← 0
    for t' ∈ (0,1) do
        x(t') ← arg max_{z ∈ P_M} { Σ_{i∈N} z_i · (F(max{y, 1_i}) − F(y)) }
        (∂y/∂t)_i (t') ← x_i(t')(1 − y_i(t'))
    end for
    return y(1)
end procedure
```

Note ∇F can be estimated efficiently with random sampling.

Lemma 8.3. For set $S \subseteq N$ and vector $\mathbf{z}$ such that $\|z\|_\infty \le p$

$$F(\max\{\mathbb{1}_S, \mathbf{z}\}) \ge (1-p) f(S) \tag{8.17}$$

Proof.

$$F(\max\{\mathbb{1}_S, \mathbf{z}\}) \ge f_L(\max\{\mathbb{1}_S, \mathbf{z}\}) \ge (1-p) f(S) \tag{8.18}$$

□

Theorem 8.4 (Feldman et al. [2011]). For submodular f,

$$F(\mathbf{y}(1)) \ge \left(1 - \frac{1}{\epsilon}\right) f(S^*) \tag{8.19}$$

where $\mathbf{y}(t)$ is output of Algorithm 8.2:

$$\mathbf{y}(t) = \int_0^t \mathbf{x}(t')\,\mathrm{d}t' \tag{8.20}$$

and S^* is optimal solution of

$$\max f(S) \tag{8.21}$$
$$\text{s.t. } S \in \mathcal{I} \tag{8.22}$$

Proof. Lets bound $\frac{\partial F}{\partial t}$:

$$\frac{\partial F}{\partial t} = \nabla F \cdot \frac{\partial y}{\partial t} = \nabla F(\mathbf{y}(t')) \cdot \mathbf{x}(t') \overset{F \text{is multilinear}}{=} \sum_{i \in N} \frac{F(\max\{\mathbf{y}, \mathbb{1}_i\}) - F(\mathbf{y})}{1 - \mathbf{y}_i} \cdot \mathbf{x}(t') \cdot (1 - \mathbf{y}_i(t')) = \tag{8.23}$$

$$= \left[F(\max\{\mathbf{y}, \mathbb{1}_i\}) - F(\mathbf{y})\right] \cdot \mathbf{x}(t') \ge \sum_{i \in S^*} F(\max\{\mathbf{y}, \mathbb{1}_i\}) - F(\mathbf{y}) \ge F(\max\{\mathbf{y}, \mathbb{1}_{S^*}\}) - F(\mathbf{y}) \tag{8.24}$$

where max is coordinate-wise maximum.
Since

$$\frac{\partial y_i}{\partial t} \le 1 - y_1 \tag{8.25}$$

we get

$$y_i \le 1 - e^{-t} \tag{8.26}$$

From 8.3 we got

$$\frac{\partial F}{\partial t} \ge F(\max\{\mathbf{y}, \mathbb{1}_{S^*}\}) - F(\mathbf{y}) \ge e^{-t} f(S^*) - F(\mathbf{y}) \tag{8.27}$$

Solving differential inequality

$$F(\mathbf{y}(0)) \ge 0 \tag{8.28}$$
$$\frac{\partial F}{\partial t} \ge f(S^*) - F(\mathbf{y}) \tag{8.29}$$

we acquire the solution $F(\mathbf{y}(t)) \ge te^{-t} f(S^*)$ with maximum in $t = 1$:

$$F(1) \ge \frac{1}{\epsilon} f(S^*) \tag{8.30}$$

Note that now $\mathbf{y}$ is not a linear combination of elements of polytope, but rather something which is at most linear combination. □

Note It is known that $\frac{1}{e}$ is not optimal, but rather at least 0.372 [Ene and Nguyen, 2016].

Rounding of the fractional solution Remember the submodular welfare problem. We define polytope

$$\mathcal{P}_M = \left\{ \mathbf{z} \in [0,1]^N \,\Big|\, \sum_{j=1}^{k} z_{i,j} \le 1 \right\} \tag{8.31}$$

Then an intuitive way to round the vector in polytope is just sample from random variable Z_i:

$$P(Z_i = j) = z_{i,j} \tag{8.32}$$

with Z_i independent. Denote by S the set acquired by this rounding.

Proposition 8.5.

$$\mathbb{E}[f(S)] \ge F(\mathbf{z}) \tag{8.33}$$

Note For submodular welfare with k players we can get $1 - \left(1 - \frac{1}{k}\right)^k$ approximation, by running Algorithm 8.2 on the previous page and stopping after $t = 1$.

Pipage rounding We choose two components of vector $\mathbf{z}$, z_i and z_j and move mass from z_i to z_j such that

- At least one of z_i and z_j becomes integer
- Total value isn't reduced (due to concavity in two variables)
- New vector is still in $\mathcal{P}_M$.

Note There exists effective algorithm to find, given $\mathbf{x}$, a convex combination of basis $\{B_1, B_2, \ldots, B_n\}$.

Algorithm 8.3

procedure MERGE BASIS($\beta_1, B_1, \beta_2, B_2$)
  while $B_1 \neq B_2$ **do**
    $i \in B_1, j \in B_2$ such that $i \in B_1 \setminus B_2, j \in B_2 \setminus B_1$
    $(B_1, B_2) = \begin{cases} (B_1, B_2 \setminus \{j\} \cup \{i\}) & \text{w.p. } \frac{\beta_2}{\beta_1+\beta_2} \\ (B_1 \setminus \{i\} \cup \{j\}, B_2) & \text{w.p. } \frac{\beta_1}{\beta_1+\beta_2} \end{cases}$
  end while
  return B_1
end procedure

Algorithm 8.4

procedure SWAP ROUND($x = \sum_{l=1}^{n} \beta_l \mathbb{1}_{B_l}$)
  $C_1 = B_1$
  for $k = 1..n-1$ **do**
    $C_{k+1} = \text{MergeBasis}\left(\sum_{l=1}^{k} \beta_l, C_k, \beta_{k+1}, B_{k+1}\right)$
  end for
  return C_n
end procedure

Swap rounding Decompose $\mathbf{z}$ as a convex combination over basis. From two vectors in combination we create a new one: there exists $x \in B_i \setminus B_j$ and $y \in B_j \setminus B_i$ such that $B_i \setminus \{x\} \cup \{y\}$ and $B_j \setminus \{y\} \cup \{x\}$ is a basis.

Theorem 8.6 (Chekuri et al. [2010]). *Proof.* What is value of $\mathbf{x}$ after one iteration of basis merge? To simpilfy, assume $\beta_1 = \beta_2 = \frac{1}{m}$. The expectation is

$$\mathbb{E}[F(\mathbf{x}')] = \frac{1}{2}F(\mathbf{x} + \beta_1(\nVdash_i) + \frac{1}{m}(\nVdash_i - \nVdash_j)) + \frac{1}{2}F(\mathbf{x} + \beta_1(\nVdash_i) + \frac{1}{m}(\nVdash_j - \nVdash_i)) \tag{8.34}$$

Define

$$g(s) = F(\mathbf{x} + s(\mathbb{1}_i - \mathbb{1}_j)) \tag{8.35}$$

Then, from convexity

$$\mathbb{E}[F(\mathbf{x}')] = \frac{1}{2}g\left(\frac{1}{m}\right) + \frac{1}{2}g\left(-\frac{1}{m}\right) \geq g(0) = F(\mathbf{x}) \tag{8.36}$$

□

9 Complexity of unconstrained submodular optimization

Theorem 9.1. For all $\epsilon > 0$ exists submodular function such that random algorithm that achieves $\left(\frac{1}{2} + \epsilon\right)$ approximation in expectation requires at least $e^{\frac{1}{8}\epsilon^2 N}$ queries to value oracle.

Proof. We'll show two submodular functions such that

- Ratio between values is approximately 2.
- The functions can be distinguished only with exponential number of queries to value oracles.

Let $g(S) = |S|(n - |S|)$ be a cut size of clique.
To define f suppose there is partition of N into two sets of equal size: $C \cup D + N$, $|C| = |D| = \frac{n}{2}$.
Denote for $S' \subseteq N$, denote $k = |S \cap C|$ and $l = |S \cap D|$. Then define

$$f(S) = \begin{cases} |S|(n - |S|) & |k - l| \leq \epsilon n \\ |k|(n - 2l) + l(n - 2k) - \underbrace{\left(\epsilon^2 n^2 - 2\epsilon n|k - l|\right)}_{\mathcal{O}(\epsilon n^2)} & |k - l| \geq \epsilon n \\ \end{cases} \tag{9.1}$$

where we added $\mathcal{O}\left(\epsilon n^2\right)$ term to make f submodular. □

The optimal value of g is $\frac{n^2}{4}$. For f, if $S = C$, $f(S) = \left(\frac{1}{2} - \epsilon\right)n^2$.
The choice of sets C, D is random and unknown to algorithm. Define

$$Y_i = \begin{cases} 1 & i \in C \\ -1 & i \in D \end{cases} \tag{9.2}$$

Then $k - l = \sum_{i \in S} Y_i$, and $P(Y_i = 1) = P(Y_i = -1) = \frac{1}{2}$. We know that for independent $-1 \leq Y_i \leq 1$ such that $\mathbb{E}[Y_i] = 0$ then

$$P\left(\sum_{i=1}^{n} Y_i > \lambda\right) \leq e^{-\frac{1}{2}\frac{\lambda^2}{n}} \tag{9.3}$$

In our case, we want to bound $P\left(\left|\sum_{i \in S} Y_i\right| > \epsilon n\right)$. To remove the dependence we define Z_i to be independent variables. Then

$$P\left(\left|\sum_{i \in S} Y_i\right| > \epsilon n\right) = P\left(\left|\sum_{i \in S} Z_i\right| > \epsilon n | \sum_{i \in N} Z_i = 0\right) = \frac{P(\left|\sum_{i \in S} Z_i\right| > \epsilon n, \sum_{i \in N} Z_i = 0)}{P\left(\sum_{i \in N} Z_i = 0\right)} \leq \frac{P(\left|\sum_{i \in S} Z_i\right| > \epsilon n)}{P\left(\sum_{i \in N} Z_i = 0\right)} \leq \tag{9.4}$$

$$\leq \frac{2e^{-\frac{1}{2}\frac{\epsilon^2 N^2}{|S|}}}{\frac{1}{2N+1}} \leq 2(2N+1)e^{-\frac{1}{2}\epsilon^2 N} \leq e^{-\frac{1}{4}\epsilon^2 N} \tag{9.5}$$

To bound the probability to get at least one non-balanced query, we'll use union bound. Even for exponential number of queries, the probability to get one such query is exponentially small.
We can use Yao's lemma to show that exists input for any algorithm that requires exponential number of queries.

Note There exists a general method called "symmetry gap" [Vondrák, 2013] to proof complexity results on submodular optimization.

10 Submodular minimization

Simplest minimization problems are unconstrained ones

$$\min_{S \subseteq N} \{f(S)\} \tag{10.1}$$

which generalizes MIN-CUT, MIN-S-T-CUT. Iwata et al. [2001] have extended the dual maximal flow problem to general unconstrained submodular minimization.

10.0.1 Continuous relaxation

The Lovasz extension together with convex minimization can be used to minimize submodular function [Grötschel et al., 1981]. Since Lovasz extension is convex, we can easily minimize it. Now we only need to round it. Choose $\theta \sim U[0,1]$ and output $\{i : x_i^* \geq \theta\}$. By definition

$$\mathbb{E}_\theta[f(\{i : x_i^* \geq \theta\})] = f_L(\bar{X}^*) \tag{10.2}$$

Note For any θ, $\{i : x_i^* \geq \theta\}$ got to be optimal, since any solution can not be better than optimum.

10.0.2 Cardinality constraint

Even for monotone f, the best known algorithm has $\tilde{\mathcal{O}}(\sqrt{n})$-approximation [Svitkina and Fleischer, 2011].

10.1 Multiway partition

Given the graph $G = (V, E)$, set of terminals $V \supseteq T = \{t_1, t_2, \dots, t_k\}$, and weight function $w : E \to \mathbb{R}_+$. We want to find $F \subseteq E$ such that in $G_F = (V, E \setminus F)$ there is no path from t_i to t_j, minimizing $\sum_{e \in F} w_e$.
On the other hand, we want to partition graph into k parts, each of which contains one terminal with minimal weight of connecting weights.

Algorithm 10.1

```
procedure PARTITION({t_1, ..., t_k})
    for i = 1..k do
        F_i ← minimal cut between t_1 and {t_2, ..., t_k}.
    end for
    return ⋃_{i=1}^k F_i
end procedure
```

Lemma 10.1. Output of Algorithm 10.1 is always legal.

Lemma 10.2. Algorithm 10.1 provides 2-approximation of multicut.

Proof. For optimal solution S^*,

$$\sum_{e \in F_i} w_e \leq \sum_{e \in \delta(S_i^*)} w_e \tag{10.3}$$

Thus, for S, the output of Algorithm 10.1

$$f(S) \leq \sum_i \sum_{e \in \delta(S_i^*)} w_e \leq 2f(S^*) \tag{10.4}$$

□

10.2 Submodular multiway partition

Given the world N and $N \supset T = \{t_1, t_2, \dots t_k\}$ and submodular $f : 2^N \to \mathbb{R}_+$, we want to find partition of N into $\{S_1, \dots S_k\}$ such that $t_i \in S_i$ and minimize

$$\sum_{i=1}^{k} f(S_k) \tag{10.5}$$

Examples

- Multiway cut: f if cut function of undirected graph.
- Hypergraph multiway cut
  - Each hyperedge cost is number of groups it is present in:

$$f(S) = \sum_{\substack{e \in E \\ 0 \leq |e \cap S| \leq |e|}} w_e \tag{10.6}$$

  - Each hyperedge cost is w_e if it is cut and 0 otherwise. In this case, for each edge we choose a vertex $u_e \in e$:

$$f(S) = \sum_{\substack{e \in E \\ u_e \in S \\ |e \cap S| \leq |e|}} w_e \tag{10.7}$$

For symmetric f there exists 1.5-approximation and for general case there exists 2-approximation.

Relaxation of the problem Define for each vertex u variable $x(u, i)$:

$$x(u, i) \geq 0 \tag{10.8}$$

$$\sum_{i=1}^{k} x(u, i) = 1 \tag{10.9}$$

Our goal is

$$\min \sum_{i=1}^{k} f_L(\mathbf{x}_i) \tag{10.10}$$

$$\text{s.t.} \sum_{i=1}^{k} x(u, i) = 1 \tag{10.11}$$

$$x(t_i, i) = 1 \tag{10.12}$$

$$x(u, i) \geq 0 \tag{10.13}$$

Denote by $\mathbf{x}_i \in [0,1]^N$ such that $(\mathbf{x}_i)_u = x(u, i)$. We can solve the relaxed problem (since it is convex problem). But how do we round the solution?
Choose threshold $\theta \in [0, 1]$ and define $A_i(\theta) = \{u | x(u, i) \geq \theta\}$. Then, for example, if $\theta \sim U[0, 1]$

$$\mathbb{E}[f(A_i(\theta))] = f_L(\mathbf{x}_i) \tag{10.14}$$

There are two "problems" we want to fix: $A_i(\theta) \cap A_j(\theta) \neq \emptyset$ and $\bigcup_i A_i(\theta) \neq N$.
Let A, B such that $A \cap B \neq \emptyset$. In symmetric case

$$f(A) + f(B) = f(N \setminus A) + f(B) \geq f((N \setminus A) \cup B) + f((N \setminus A) \cap B) = f(A \cap B) + f(B \setminus A) \tag{10.15}$$

Since

$$\frac{1}{2}[f(A \setminus B) + f(B)] + \frac{1}{2}[f(B \setminus A) + f(A)] \leq f(A) + f(B) \tag{10.16}$$

at least one of choices $\{A \setminus B, B\}$ and $\{B \setminus A, A\}$ is at least as good as $\{A, B\}$. This action is called uncrossing. By repeating the process we will get disjoint sets.
Now, denote $A(\theta) = \bigcup_i A_i(\theta)$ and $V(\theta) = N \setminus A(\theta)$. What would be value of output:

$$\sum_{i=1}^{k-1} f(A_i(\theta)) + f(A_k(\theta) \cup f(V(\theta))) \leq \sum_{i=1}^{k} f(A_i(\theta)) + f(V(\theta)) \tag{10.17}$$

Thus we only need to bound $f(V(\theta))$:

$$\mathbb{E}_\theta[f(V(\theta))] \leq \alpha \sum_{i=1}^{k} f_L(\mathbf{x}_i) \tag{10.18}$$

Naively, in symmetric case, we could bound by

$$f(V(\theta)) = f(A(\theta)) \leq \sum_{i=1}^{k} f(A_i(\theta)) \tag{10.19}$$

which provides 2-approximation.

Lemma 10.3. For all $\frac{1}{2} \leq \delta \leq 1$

$$\int_0^1 f(V(\theta))\,\mathrm{d}\theta \leq \int_0^\delta f(A_1(\theta) \cap A_2(\theta))\,\mathrm{d}\theta + \int_0^\delta f\left(\Big[A_1(\theta) \cup A_2(\theta)\Big] \cap A_3(\theta)\right)\mathrm{d}\theta + \tag{10.20}$$

$$+ \cdots + \int_0^\delta f\left(\Big[A_1(\theta) \cup A_2(\theta) \cup \cdots \cup A_{k-1}\Big] \cap A_k(\theta)\right)\mathrm{d}\theta = \sum_{i=1}^{k-1} \int_0^\delta f\left(\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)\right)\mathrm{d}\theta \tag{10.21}$$

Proof. First assume $\delta = 1$, then

$$\int_0^1 f\big(\big[A_1(\theta) \cup \cdots \cup A_i(\theta)\big] \cap A_{i+1}(\theta)\big) = f_L(\mathbf{y}_i) \tag{10.22}$$

where

$$\mathbf{y}_i = \min\left\{\mathbf{x}_{i+1}, \max_{1 \leq j \leq i} \mathbf{x}_j\right\} \tag{10.23}$$

for elementwise maximum and minimum. Since

$$\min\{x_1, x_2\} + \min\{\max\{x_1, x_2\}, x_3\} + \cdots + \min\{\max\{x_1, x_2, \ldots, x_{n-1}\}, x_n\} \max\{x_1, x_2, \ldots, x_n\} = \sum x_i \tag{10.24}$$

we get

$$\sum_{i=1}^{k-1} \mathbf{y}_i + \max_{1 \leq j \leq k} \mathbf{x}_j = \mathbf{1} \tag{10.25}$$

On the other hand, since $(1-\theta)$ is also uniform

$$\int_0^1 f(V(\theta))\,\mathrm{d}\theta = f_L\left(\mathbf{1} - \max_{1 \leq j \leq k} \mathbf{x}_j\right) = f_L\left(\sum_{i=1}^{k-1} \mathbf{y}_i\right) = (k-1) f_L\left(\frac{1}{k-1} \sum_{i=1}^{k-1} \mathbf{y}_i\right) \overset{\text{convex}}{\leq} \sum_{i=1}^{k-1} f_L(\mathbf{y}_i) \tag{10.26}$$

Now, assuming $f(\emptyset) = 0$, we can rewrite

$$\int_0^1 f\left(\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)\right)\mathrm{d}\theta = \int_0^\delta f\left(\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)\right)\mathrm{d}\theta + \int_\delta^1 f\left(\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)\right)\mathrm{d}\theta \tag{10.27}$$

However, since we are on simplex, for $\delta > \frac{1}{2}$, $\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)) = \emptyset$. ∎

Lemma 10.4.

$$\sum_{i=1}^{k} f(A_i(\theta)) \geq f(A_1(\theta) \cap A_2(\theta)) + f\left(\Big[A_1(\theta) \cup A_2(\theta)\Big] \cap A_3(\theta)\right) + \tag{10.28}$$

$$+ \cdots + f\left(\Big[A_1(\theta) \cup A_2(\theta) \cup \cdots \cup A_{k-1}\Big] \cap A_k(\theta)\right) + f\left(\bigcup_{i=1}^{k} A_i(\theta)\right) \tag{10.29}$$

Proof. By induction. Basis. For $k = 2$:

$$f(A_1(\theta)) + f(A_2(\theta)) = f(A_1(\theta) \cap A_2(\theta)) + f(A_1(\theta) \cup A_2(\theta)) \tag{10.30}$$

Step

$$\sum_{i=1}^{k} f(A_i(\theta)) \geq \sum_{i=1}^{k-2} f\left(\bigcup_{j=1}^{i} A_j(\theta) \cap A_{i+1}(\theta)\right) + f\left(\bigcup_{j=1}^{k-1} A_j(\theta)\right) + f(A_k(\theta)) \geq \tag{10.31}$$

$$\geq \sum_{i=1}^{k-1} f\left(\bigcup_{j=1}^{i} A_j(\theta) \cap A_{i+1}(\theta)\right) + f\left(\bigcup_{j=1}^{k} A_j(\theta)\right) \tag{10.32}$$

□

Theorem 10.5. For all $\frac{1}{2} \leq \delta \leq 1$

$$\sum_{i=1}^{k} \int_0^{\delta} f(A_i(\theta))\,\mathrm{d}\theta \geq \int_0^1 f(V(\theta))\,\mathrm{d}\theta + \int_0^{\delta} f(A(\theta))\,\mathrm{d}\theta \tag{10.33}$$

Proof. Thus, substituting Lemma 10.3 on the previous page into Lemma 10.4 on the preceding page

$$\sum_{i=1}^{k} \int_0^{\delta} f(A_i(\theta))\,\mathrm{d}\theta \geq \sum_{i=1}^{k-1} \int_0^{\delta} f\left(\left[\bigcup_{j=1}^{i} A_j(\theta)\right] \cap A_{i+1}(\theta)\right) + f\left(\bigcup_{i=1}^{k} A_i(\theta)\right) \mathrm{d}\theta \geq \int_0^1 f(V(\theta))\,\mathrm{d}\theta + \int_0^{\delta} f(A(\theta))\,\mathrm{d}\theta \tag{10.34}$$

□

Theorem 10.6 (Chekuri and Ene [2011])**.** There exist 1.5-approximation for symmetric submodular multiway partition.

Proof. For symmetric f, we choose $\delta = 1$ in Theorem 10.5, and get:

$$\sum_{i=1}]^{k} \int_0^1 f(A_i(\theta))\,\mathrm{d}\theta \geq \int_0^1 f(V(\theta))\,\mathrm{d}\theta + \int_0^1 f(A(\theta))\,\mathrm{d}\theta = 2\int_0^1 f(V(\theta))\,\mathrm{d}\theta \tag{10.35}$$

$$2\mathbb{E}_{\theta} f(V(\theta)) \leq \mathbb{E}_{\theta} \sum_{i=1}]^{k} f(A_i(\theta)) = \sum_{i=1}^{k} f(A_i(\theta)) \tag{10.36}$$

i.e., $\alpha = \frac{1}{2}$ and we got 1.5-approximation. □

Theorem 10.7 (Chekuri and Ene [2011])**.** There exist 2-approximation for submodular multiway partition.

Proof. If f is not symmetric, we can choose $\theta \sim U[0.5, 1]$, A_i will be guaranteed to be disjoint. In this case

$$\mathbb{E}_{\theta}\left[\sum_{i=1}^{k-1} f(A_i(\theta)) + f(A_k(\theta) \cup f(V(\theta)))\right] \leq \mathbb{E}_{\theta}\left[\sum_{i=1}^{k} f(A_i(\theta)) f(f(V(\theta)))\right] = \sum_{i=1}^{k} \int_{\frac{1}{2}}^{1} 2f(A_i(\theta))\,\mathrm{d}\theta + \int_{\frac{1}{2}}^{1} 2f(V(\theta))\,\mathrm{d}\theta = \tag{10.37}$$

$$= 2\left[\sum_{i=1}^{k} f_L(\mathbf{x}_i) - \sum_{i=1}^{k} \int_0^{\frac{1}{2}} f(A_i(\theta))\,\mathrm{d}\theta + \int_{\frac{1}{2}}^{1} f(V(\theta))\,\mathrm{d}\theta\right] \overset{\text{Th. 10.5}}{\leq} 2\sum_{i=1}^{k} f_L(\mathbf{x}_i) \tag{10.38}$$

□

Lower bounds For symmetric problem, it is unknown whether $\frac{3}{2} - \frac{1}{k}$ is optimal. The known lower bound is 1.26 [Ene et al., 2013].

10.3 Submodular covering problems

10.3.1 Submodular vertex cover

For graph $G = (V, E)$ and submodular $f : 2^V \to \mathbb{R}_+$. The goal is to find $A \subset V$, such that G is vertex cover (i.e., for $(u, v) \in E$, $u \in A$ or $v \in A$) and $f(A)$.
For each vertex define variable x_i, and the relaxation is

$$\min f_L(\mathbf{x}) \tag{10.39}$$

$$\text{s.t. } \forall (u, v) \in E\ x_u + x_v \geq 1 \tag{10.40}$$

$$\text{s.t. } \forall u \in V\ 0 \leq x_u \leq 1 \tag{10.41}$$

Proposition 10.8. There exists half-integer optimum of Eq. (10.39) on the previous page, i.e., $\forall v \in V\ x_v^k \in \{0, \frac{1}{2}, 1\}$. It is possible to find it efficiently.

Proposition 10.9. Given half-integer optimal solution,

$$A = \left\{u | x_u^* \geq \frac{1}{2}\right\} \tag{10.42}$$

is 2-approximation of submodular vertex cover.

Proof.

$$f_L(vbx^*) = \frac{1}{2}f(A) + \frac{1}{2}f(\{u | x_u^* = 1\}) \leq \frac{1}{2}f(A) \tag{10.43}$$

□

Proposition 10.10. Under unique games conjecture, 2-approximation is optimal.

References

Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. Society for Industrial and Applied Mathematics, 2014. (cited on pp. 4 and 5)

Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015. (cited on p. 8)

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011. (cited on p. 13)

Chandra Chekuri and Alina Ene. Approximation algorithms for submodular multiway partition. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 807–816. IEEE, 2011. (cited on p. 20)

Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010. (cited on p. 15)

Alina Ene and Huy L. Nguyen. Constrained submodular maximization: Beyond 1/e. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 248–257. IEEE, 2016. (cited on p. 15)

Alina Ene, Jan Vondrák, and Yi Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 306–325. Society for Industrial and Applied Mathematics, 2013. (cited on p. 20)

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. (cited on p. 4)

Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011. (cited on p. 6)

Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 570–579. IEEE, 2011. (cited on p. 14)

Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. (cited on p. 17)

Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001. (cited on p. 17)

Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information processing letters*, 70(1):39–45, 1999. (cited on p. 8)

George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978. (cited on p. 4)

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical programming*, 14(1):265–294, 1978. (cited on p. 3)

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004. (cited on p. 8)

Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011. (cited on p. 17)

Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1): 265–304, 2013. (cited on p. 16)

Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935. (cited on p. 9)