9. Submodular function optimization

- Submodular function maximization
- Greedy algorithm for monotone case
- Influence maximization
- Greedy algorithm for non-monotone case
Combinatorial optimization problems

Knapsack problem
given \( n \) items each with cost \( c_i \) and value \( v_i \), and a budget \( B \), the goal of the **knapsack problem** is to find a subset of items \( S^* \) such that the value is maximized while cost does not exceed the budget, i.e.

\[
\begin{align*}
\text{maximize} & \quad v^T x \\
\text{subject to} & \quad c^T x \leq B \\
& \quad x_i \in \{0, 1\}, \quad \forall i \in [n]
\end{align*}
\]

MaxCover problem
consider a set of \( n \) elements \([n] = \{1, \ldots, n\}\), and \( m \) subsets \( T_1, \ldots, T_m \subseteq [n] \), and the goal of **MaxCover problem** is to find \( K \) subsets whose union has the largest cardinality, i.e.

\[
\begin{align*}
\text{maximize} & \quad 1^T z \\
\text{subject to} & \quad z_j \leq \sum_{i:j \in T_i} x_i \\
& \quad 1^T x \leq K \\
& \quad z_j \leq 1, \quad \forall j \in [n] \\
& \quad x_i \in \{0, 1\}, \quad \forall i \in [m]
\end{align*}
\]
Maximizing submodular functions

consider a set of elements $[n] = \{1, \ldots, n\}$ and a real-valued function over a subset of the elements

$$ f : 2^n \rightarrow \mathbb{R} \quad \text{such that} \quad X \subseteq [n] \mapsto f(X) $$

★ a function $f$ is monotone if for all $S \subseteq T$, $f(S) \leq f(T)$
★ the marginal contribution of a function $f$ of an element $i$ to a set $S$ is defined as $f_S(i) = f(S \cup \{i\}) - f(S)$
★ a function $f$ is submodular if for any $i \in [n]$ and any $S \subseteq T$ we have

$$ f_S(i) \geq f_T(i) $$

and we are interested in the following optimization problem for a submodular objective function $f$

$$ \text{maximize} \quad f(S) $$
$$ \text{subject to} \quad S \in \mathcal{F} $$

for a subset $S \subseteq [n]$ that satisfy a certain constraint represented by the feasible set $\mathcal{F}$
equivalently, a function is submodular if and only if for all $S, T \subseteq [n]$,

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$$

knapsack problem is a submodular maximization with $f(S) = \sum_{i \in S} v_i$ and $\mathcal{F} = \{S \subseteq [n] : c(S) \leq B\}$ where $c(S) = \sum_{i \in S} c_i$, and $f$ is submodular and monotone

MaxCover problem is a submodular maximization with $f(S) = |\bigcup_{i \in S} T_i|$ and $\mathcal{F} = \{S \subseteq [n] : |S| \leq K\}$, and $f$ is submodular and monotone

many interesting problems can be formulated as submodular function maximization problem
Greedy algorithm with approximation guarantee

first, consider a special case of **monotone and submodular** $f$, and the constraint is **cardinality constraint** with $\mathcal{F} = \{S \subseteq [n] : |S| \leq K\}$

**Greedy algorithm**

1. set $S = \emptyset$
2. while $|S| \leq K$
   
   add to $S$ an element $i$ that maximizes $f(S \cup \{i\})$
3. end while.

surprisingly, this greedy algorithm gives a good approximate solution as proved by Fisher, Nemhauser and Wolsey in 1978

**theorem.** (approximation) If $f$ monotone, submodular and $f(\emptyset) = 0$, then the greedy algorithm returns a solution that achieves

$$f(S) \geq \left(1 - \frac{1}{e}\right)OPT$$

where $OPT = \max_{S : |S| \leq K} f(S)$

**theorem.** (converse result by Feige 1998) it is NP-hard to get a $(1 - \frac{1}{e}) + \varepsilon$ approximation for any $\varepsilon > 0$ for MaxCover problem (proof is beyond the scope of this lecture)

Submodular function maximization
proof of the approximation guarantee.

let $S_i$ be the set after $i$ iterations of the greedy algorithm, then by monotonicity $f(S^*) \leq f(S_i \cup S^*)$, where $S^*$ is the optimal set by submodularity,

$$
f(S^*) \leq f(S_i \cup S^*) \\
\leq f(S_i) + (f(S_i \cup \{x_1\}) - f(S_i)) + (f(S_i \cup \{x_1, x_2\}) - f(S_i \cup \{x_1\})) + \cdots \\
\leq f(S_i) + (f(S_i \cup \{x_1\}) - f(S_i)) + f_{S_i}(x_2) + \cdots + f_{S_i}(x_k) \\
\leq f(S_i) + K(f(S_{i+1}) - f(S_i))
$$

for $S^* = \{x_1, \ldots, x_k\}$, and this gives

$$
f(S_{i+1}) - f(S_i) \geq \frac{1}{K} (OPT - f(S_i))
$$

now we use induction to prove that

$$
f(S_i) \geq \left(1 - (1 - \frac{1}{K})^i \right) OPT
$$

it is trivially true for $i = 0$ and we write

$$
f(S_{i+1}) \geq f(S_i) + \frac{1}{K} (OPT - f(S_i)) \\
= \left(1 - \frac{1}{K}\right)f(S_i) + \frac{1}{K} OPT \\
\geq \left(1 - \frac{1}{K}\right) \left(1 - \left(1 - \frac{1}{K}\right)^i\right) OPT + \frac{1}{K} OPT \quad \text{[by induction hypothesis]} \\
= \left(1 - \left(1 - \frac{1}{K}\right)^{i+1}\right) OPT
$$

this implies $f(S) \geq (1 - (1 - 1/K)^K) OPT \geq (1 - 1/e) OPT$
data-dependent bound

suppose $S$ is the output of the greedy algorithm and $S^*$ is the optimal solution

$\star$ let $\delta_i = f_S(i)$ and suppose $\delta_i$’s are sorted such that $\delta_1 \geq \delta_2 \geq \delta_3 \ldots$

$\star$ then similar argument shows that

$$f(S^*) \leq f(S \cup S^*) \leq f(S) + \delta_1 + \delta_2 + \cdots + \delta_k$$

gain is significant if $\delta_1 \ll f(S)$
next, we consider a more general case where $f$ is still monotone and submodular, but the constraint is the **budget constraint** with

$$\mathcal{F} = \{S \subseteq [n] : C(S) \leq B\}$$

where $C(S) = \sum_{i \in S} c_i$ for some elementwise cost $c_i$

**Cost-benefit greedy algorithm**

1. set $S = \emptyset$
2. while there exists an $i \in [n]$ such that $C(S \cup \{i\}) \leq B$
   
   add to $S$ an element $i$ that maximizes
   $$\frac{f(S \cup \{i\}) - f(S)}{c_i}$$
   
   subject to $C(S \cup \{i\}) \leq B$
3. end while.

★ performance of the above algorithm can be arbitrarily bad, for example if $B = 1$ and $f(\{1, 2\}) = f(1) + f(2)$ with

item 1 : $f(1) = 2\epsilon$, $c_1 = \epsilon$

item 2 : $f(2) = 1$, $c_2 = 1$

Cost-benefit greedy algorithm chooses $S = \{1\}$ with $F(S) = 2\epsilon$, whereas the optimal solution is $F(\{2\}) = 1$
theorem. [Leskovec et al KDD ’07] let $S_{CB}$ be the solution of Cost-benefit greedy algorithm and let $S_{UC}$ be the output of the greedy algorithm treating the costs as equal, then

$$\max\{f(S_{CB}), f(S_{UC})\} \geq \frac{1}{2} \left(1 - \frac{1}{e}\right) OPT$$
Examples of submodular function maximization

Influence maximization

motivation: given a target customers in a social network, how can we choose a set $S$ of early adopters and market them in order to generate a cascade of adoptions?

define influence function $f(S)$ as the expected number of influenced nodes at the end of the process starting with $S$ in a finite network $G(V,E)$ (to be formally defined below)

$$\text{maximize} \quad f(S)$$

subject to $c(S) \leq B$

**Theorem.** [Kempe et al. '03] under general cascade model, influence maximization is NP-hard to approximate to a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$
independent cascade model

for each edge \((i, j)\) there is a probability \(p_{ij}\) such that if \(i\) is activated
then it has one chance to activate \(j\) with probability \(p_{ij}\)

**Theorem.** [Kempe et al '03] \(f\) is submodular under independent cascade model

**Proof.**
- Consider an equivalent scenario where a biased coin is flipped for each edge in the beginning to determine whether the edge is active or not, then a node \(i\) is active if there exists a path of active edges from the node \(i\) to set \(S\).
- For a given instance of the active edges, the function of influence is submodular, since for all \(S \subseteq T\), the (deterministic) function \(f(S \cup \{i\}) - f(S)\) is the number of nodes that are reachable from \(i\) using active edges that is not already reachable from \(S\).
- And if \(T\) includes \(S\), this number only gets smallet for \(T\), i.e.

\[
 f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)
\]

- Since the function is submodular for each instance of the random trial, it is submodular in expectation as well, from the linearity of expectation

\[
 f(S) = \sum_Z f_Z(S) \mathbb{P}(Z)
\]
linear threshold model

for each edge $(i, j)$ there is a nonnegative weight $w_{ij}$ such that

$\sum_i w_{ij} \leq 1$

each node $i$ has threshold $\theta_i$ uniformly a trandom in $[0, 1]$ such that $i$
becomes active if

$$\sum_{\text{active } i} w_{ij} \geq \theta_j$$

**theorem.** [Kempe et al '03] $f$ is submodular under linear threshold
model
feature selection
given random variables \( Y, X_1, \ldots, X_n \), want to predict \( Y \) from subset \( X_A = \{X_{i_1}, \ldots, X_{i_k}\} \)

\[
\begin{align*}
Y & : \text{patient is sick} \\
X_1 & : \text{patient has fever} \\
X_2 & : \text{patient has rash} \\
X_3 & : \text{patient is male}
\end{align*}
\]

how do we find \( k \) most informative features?

\[
f(S) = I(Y; X_S) = H(Y) - H(Y | X_S)
\]

\( f \) monotonic and submodular if \( X_i \)'s are conditionally independent given \( Y \) [Krause, Guestrin UAI '05]
Maximizing non-monotone but submodular function $f$

Now we slightly relax the previous assumptions and suppose that $f$ is submodular, but not necessarily monotone, and there is no constraint on $f$

- this problem is NP-hard in general,
- if $f$ can take negative values, hard to even approximate, i.e. $O(n^{1-\epsilon})$ approximation is NP-hard

however, when $f$ is non-negative, there is an efficient greedy algorithm:

**Non-negative greedy algorithm**

1. set $S = \{i\}$ where $i$ maximizes $f(\{i\})$
2. while there exists an $i \in [n] \setminus S$ such that $f(S \cup \{i\}) \geq f(S)$
   - add to $S$ an element $i$
   - return to 2.
3. while there exists an $i \in S$ such that $f(S \setminus \{i\}) \geq f(S)$
   - remove $i$ from $S$
   - return to 2.

and achieves good approximation guarantee

**Theorem.** [Feige ’07] for non-negative and submodular $f$, Non-negative greedy algorithm achieves $\frac{1}{3}$-approximation.
proof.  

let $S^*$ be the optimal solution and $S$ be the output of the greedy algorithm, then

$$f(S^*) \leq f(S^*) + f(\emptyset) + f([n]) \quad \text{[by non-negativity]}$$

$$\leq f(S^* \cap S) + f(S^* \cap S^c) + f([n]) \quad \text{[by sub-modularity]}$$

$$\leq f(S^* \cap S) + f(S^* \cup S) + f(S^c) \quad \text{[by submodularity]}$$

$$\leq f(S) + f(S) + f(S^c) \quad \text{[by the following lemma]}$$

this proves that

$$\max\{S, S^c\} \geq \frac{1}{3} OPT$$

\[\Box\]

lemma. for any subset $T$ of $S$, i.e. $T \subseteq S$ or a superset $T$ of $S$, i.e. $S \subseteq T$, we have

$$f(S) \geq f(T)$$

proof. consider any sequence of increasing sets

$$\emptyset = T_0 \subseteq T_1 \cdots \subseteq T_k = S \subseteq T_{k+1} \subseteq \cdots \subseteq T_n = [n]$$

where $|T_{i+1} \setminus T_i| = 1$ and $a_i \triangleq t_{i+1} \setminus T_i$

★ for $i < k$,

$$f(T_{i+1}) - f(T_i) \geq f(S) - f(S \setminus \{a_i\}) \quad \text{[by submodularity]}$$

$$\geq 0 \quad \text{[by greedy search]}$$
for \( i > k \),

\[
f(T_i) - f(T_{i-1}) \leq f(S \cup \{a_i\}) - f(S) \quad \text{[by submodularity]}
\]

\[
\leq 0 \quad \text{[by greedy search]}
\]
further reading

- submodular function maximization with matroid constraints
  - greedy algorithm achieves \((1 - 1/e)\) approximation

- robust optimization
  \(\max_S \min_i f_i(S)\) for a class of submodular functions \(f_i\)'s [Krause et al '07]
  does not admit any approximation unless \(P=NP\)

- minimizing submodular function
Minimizing submodular functions

example: factoring distributions

given random variables $X_1, \ldots, X_n$, partition the variables into two sets $S$ and $[n] \setminus S$ as independent as possible

\[
\begin{align*}
\text{minimize} & \quad f(S) \\
\text{subject to} & \quad c(S) \leq B
\end{align*}
\]

where

\[
f(S) = I(X_S; X_{[n]\setminus S}) = H(X_{[n]\setminus S}) - H(X_{[n]\setminus S} | X_S)
\]

$f$ is submodular and \textbf{symmetric} where

da sub modular function is symmetries if and only if for all $S \subseteq [n]$

\[
f(S) = f(S^c)
\]

- another example of a symmetric submodular set function is graph cut

Submodular function maximization
remark. A symmetric submodular function is (trivially) minimized at $f(\emptyset) = f([n])$

proof. 

$$f(S) = \frac{1}{2}(f(S) + f(S^c)) \geq \frac{1}{2}(f(\emptyset) + f([n])) = f(\emptyset)$$

we are interested in the following unconstrained minimization problem:

$$\text{minimize}_{\emptyset \subset S \subset [n]} f(S)$$

theorem. [Queyranne ’98] For symmetric submodular $f$, there is an algorithm with runtime $O(n^3)$ for solving the unconstrained minimization problem.