

Utility Function

Food = {Sushi, Gyu-don, Pasta, Melon-pan, Cola}

Food In One Meal \subseteq Food

$f(\text{Food In One Meal}) :=$ How much happiness you have when you are provided
 ↑
 Food In One Meal
 Utility function

$$f(\{\text{Sushi}, \text{Cola}\}) = 100$$

$$f(\{\text{Melon-pan}, \text{Cola}\}) = 10$$

Properties of Utility Function

1. Monotonicity

$$f(s) \leq f(s') \quad \text{when } s \leq s'$$

Ex

$$f(\{\text{Sushi}, \text{Cola}\}) \leq f(\{\text{Sushi}, \text{Melon-pan}, \text{Cola}\})$$

2. Subadditivity

$$f(s \cup \{e\}) - f(s) \geq f(s' \cup \{e\}) - f(s') \quad \begin{array}{l} \text{we are really happy to have more} \\ \text{we do not have enough food} \end{array} \quad \begin{array}{l} \text{we are not that happy to have more.} \\ \text{we already have enough food} \end{array}$$

Ex

$$\begin{aligned} f(\{\text{Cola}\} \cup \{\text{Sushi}\}) &\sim f(\{\text{Cola}\}) \\ &\geq f(\{\text{Pasta, Cola}\} \cup \{\text{Sushi}\}) - f(\{\text{Pasta, Cola}\}) \end{aligned}$$

$$\Delta_c f(s) := f(s \cup \{e\}) - f(s) \Leftrightarrow \frac{f(x+\varepsilon) - f(x)}{\varepsilon} = \frac{df(x)}{dx} = f'(x)$$

$$\Delta_c f(s) \geq \Delta_c f(s') \Leftrightarrow f'(x) \geq f'(x')$$

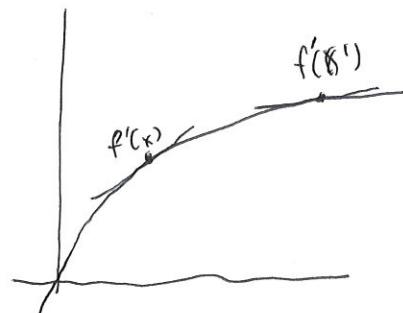
when $s \leq s'$

when $x \leq x'$

concave function

Discrete version of
concave function

\Leftrightarrow



By Lovasz extension, submodular function is also close to convex function.

More example : Set of persons $\sim \{1, 2, 3, \dots, 11\}$

Mailing list A = $\{1, 2, 3\}$

Mailing list B = $\{2, 3, 4, 5\}$

Mailing list C = $\{5, 9, 10, 11\}$

Mailing list D = $\{4, 6, 7, 8\}$

$S :=$ set of mailing lists that we send e-mails to

$f(S) :=$ #users that we can reach by S.

Utility of S

$$f(\{A, B, C\}) = |\{1, 2, 3\} \cup \{2, 3, 4, 5\} \cup \{5, 9, 10, 11\}| = \{1, 2, 3, 4, 5, 9, 10, 11\} = 8$$

$$f(\{A, B, C, D\}) = 11 \quad f(\{A, C\}) = 7 \quad f(\{A, C, D\}) = 11$$

$$\frac{f(\{A, B, C, D\}) - f(\{A, B, C\})}{11 - 8} \leq \frac{f(\{A, C, D\}) - f(\{A, C\})}{11 - 8}$$

$$\Delta_D f(\{A, B, C\}) \leq \Delta_D f(\{A, C\})$$

#persons that #persons that

1. can reach by Mailing list D \equiv 1. can reach by Mailing list D

2. cannot reach by Mailing list $\{A, B, C\}$ \leq 2. cannot reach by Mailing list A, C

$\therefore f$ is a submodular function.

Also, when $S \subseteq S'$, we can reach more mailing lists with S' than with S
more persons

$$f(S) \leq f(S').$$

$\therefore f$ is a monotone function.

Problem : Maximum Coverage Problem

Input : # persons n (set of persons = $\{1, \dots, n\}$) , positive integer k .

Set of persons that can be reached by each mailing list

$$S_1, \dots, S_m \subseteq \{1, \dots, n\}$$

→ set of persons that can be reached by mailing list i .

Output : Set of mailing list to send an e-mail to $S \subseteq \{1, \dots, m\}$

Constraint : $|S| \leq k$

If $i \in S$, we send an e-mail to Mailing list i .

[We cannot send emails to more than k mailing lists]

Objective Function : Maximize # persons we can reach by S .

$$\text{Maximize } \left(\bigcup_{i \in S} S_i \right) f(S)$$

Reformulate the problem

Input : # persons n , integer k , $f: 2^{\{1, \dots, n\}} \rightarrow \mathbb{R}$.

(f is given as a code in Python (oracle))

Output : $S \subseteq \{1, \dots, n\}$

(Ex person reached)

Constraint : $|S| \leq k$

(f is a monotone submodular function)

Objective Function Maximize $f(S)$

Submodular function maximization
under size constraint.

Algorithm,

1: $S \leftarrow \emptyset$

2: for $i=1$ to k

3: Find j that maximizes $\Delta_j f(S)$

4: $S \leftarrow S \cup \{j\}$

F k=3

Step 1 $S = \emptyset$

$$\Delta_A f(S) = 3 \quad \Delta_B f(S) = 4 \quad \Delta_C f(S) = 4 \quad \Delta_D f(S) = 4$$

maximum

o Choose B

Step 2 $S = \{B\}$

$$\Delta_A f(S) = 1 \quad \Delta_B f(S) = 0 \quad \Delta_C f(S) = 3 \quad \Delta_D f(S) = 3$$

maximum

o Choose C

Step 3 $S = \{B, C\}$

$$\Delta_A f(S) = 1 \quad \Delta_B f(S) = 0 \quad \Delta_C f(S) = 0 \quad \Delta_D f(S) = 3$$

maximum.

o Choose D

Solution from the algorithm = $\{B, C, D\}$ $f(\{B, C, D\}) = 10$

Best solution = $\{A, C, D\}$ $f(\{A, C, D\}) = 11$

Theorem If SOL := solution from the algorithm

OPT := best solution, then

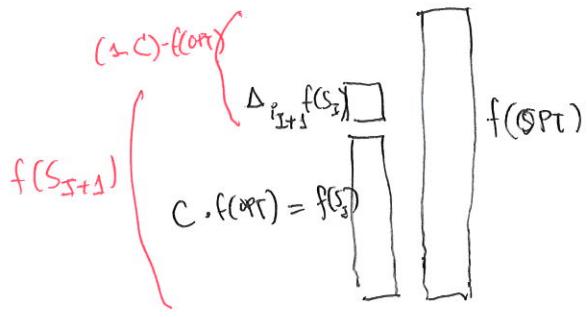
$$f(SOL) \geq 0.63 \cdot f(OPT)$$

Notation i_t := element added at iteration t

S_t := set S at the end of iteration t = $\{i_1, \dots, i_t\}$

$SOL := \{p_1, \dots, p_k\}$, and i_t maximizes $\Delta_{i_t} f(\{i_1, \dots, i_{t-1}\})$

$OPT := \{i_1^*, \dots, i_k^*\}$



$$c \geq 1 - (1 - \frac{1}{k})^k$$

By lemma,

$$\Delta_{i,i+1} f(S_i) \geq \frac{1}{k} (f(OPT) - f(S_i))$$

$$= \frac{1}{k} (1 - c) \cdot f(OPT)$$

$$\begin{aligned}
 f(S_{i+1}) &= c \cdot f(OPT) + \Delta_{i+1} f(S_i) \\
 &\geq c \cdot \underline{f(OPT)} + \frac{1}{k} (1 - c) \cdot \underline{f(OPT)} \\
 &= [c + \frac{1}{k} (1 - c)] \cdot f(OPT) \\
 &= [(1 - \frac{1}{k})c + \frac{1}{k}] \cdot f(OPT) \\
 &\geq [(1 - \frac{1}{k}) \cdot (1 - (1 - \frac{1}{k})^k) + \frac{1}{k}] \cdot f(OPT) \\
 &= [1 - \cancel{\frac{1}{k}} - (1 - \frac{1}{k})^{k+1} + \cancel{\frac{1}{k}}] \cdot f(OPT) \\
 &= [1 - (1 - \frac{1}{k})^{k+1}] \cdot f(OPT) \\
 f(S_{i+1}) &\geq [1 - (1 - \frac{1}{k})^{k+1}] \cdot f(OPT) \quad \text{□}
 \end{aligned}$$

Theorem

$$f(SOL) = f(S_k) \geq 0.63 \cdot f(OPT)$$

Proof

$$f(S_k) \geq (1 - (1 - \frac{1}{k})^k) \cdot f(OPT)$$

$\leq e^{-\frac{1}{k}}$

$$1 - x \leq e^{-x}$$

$$1 - \frac{1}{k} \leq e^{-\frac{1}{k}}$$

$$\geq (1 - (e^{-\frac{1}{k}})^k) \cdot f(OPT)$$

$$= (\underbrace{1 - e^{-1}}_{\geq 0.63}) \cdot f(OPT)$$

$$\geq 0.63 \cdot f(OPT) \quad \text{□}$$

Lemma $f(\text{OPT}) - f(\text{SOL}) \leq \Delta_{i_1^*} f(\text{SOL}) + \Delta_{i_2^*} f(\text{SOL}) + \dots + \Delta_{i_k^*} f(\text{SOL})$

Proof

$$\begin{aligned} f(\text{SOL} \cup \{i_1^*\}) - f(\text{SOL}) &= \Delta_{i_1^*} f(\text{SOL}) \\ f(\text{SOL} \cup \{i_1^*, i_2^*\}) - f(\text{SOL} \cup \{i_1^*\}) &= \Delta_{i_2^*} f(\text{SOL} \cup \{i_1^*\}) \leq \Delta_{i_2^*} f(\text{SOL}) \\ f(\text{SOL} \cup \{i_1^*, i_2^*, i_3^*\}) - f(\text{SOL} \cup \{i_1^*, i_2^*\}) &= \Delta_{i_3^*} f(\text{SOL} \cup \{i_1^*, i_2^*\}) \leq \Delta_{i_3^*} f(\text{SOL}) \\ &\vdots \\ f(\text{SOL} \cup \{i_1^*, \dots, i_k^*\}) - f(\text{SOL} \cup \{i_1^*, \dots, i_{k-1}^*\}) &= \Delta_{i_k^*} f(\text{SOL} \cup \{i_1^*, \dots, i_{k-1}^*\}) \leq \Delta_{i_k^*} f(\text{SOL}) \end{aligned}$$

$$f(\text{OPT}) - f(\text{SOL}) \leq f(\text{SOL} \cup \text{OPT}) = f(\text{SOL} \cup \{i_1^*, \dots, i_k^*\}) \leq \sum_k \Delta_{i_k^*} f(\text{SOL})$$

$$f(\text{OPT}) - f(\text{SOL}) \leq \Delta_i \Delta_{i_k^*} f(\text{SOL})$$

□

Lemma $\Delta_{i_{j+1}^*}(S_j) \geq \frac{1}{k} (f(\text{OPT}) - f(S_j))$

Proof

$$\begin{aligned} \frac{f(\text{OPT}) - f(S_j)}{k} &\leq \frac{\Delta_{i_1^*} f(S_j) + \dots + \Delta_{i_k^*} f(S_j)}{k} \quad \text{average of } \Delta_{i_j^*} f(\text{SOL}) \\ &\leq \max_j \Delta_{i_j^*} f(S_j) = \Delta_{i_{j+1}^*} f(S_j) \end{aligned}$$

□

Lemma

$$f(S_j) \geq (1 - (1 - 1/k)^j) \cdot f(\text{OPT})$$

Proof

When $j=0$,

$$\begin{aligned} 0 = f(\emptyset) &= f(S_0) \geq (1 - (1 - 1/k)^0) \cdot f(\text{OPT}) \\ &= (1 - 1) \cdot f(\text{OPT}) = 0 \end{aligned}$$

Proof by induction,

Assume that $f(S_j) \geq (1 - (1 - 1/k)^j) \cdot f(\text{OPT})$,

we will show that $f(S_{j+1}) \geq (1 - (1 - 1/k)^{j+1}) \cdot f(\text{OPT})$