Chapter 9 : Influence maximization

→ $\boxed{\text{IM\_MAX is NP-hard.}}$

→ 좋은 approximation algorithm은? (− polynomial time / − solution quality는 original solution과 가까운)

→ 상황마다 다르다. 과거의 lesson 등을 통해서 그런 문제에 좋은 approximation을 만든다.

정리

$f(C)$

$C \rightarrow \boxed{f} \rightarrow$ 결과 (평균)

$C' \rightarrow \boxed{f} \rightarrow$ 결과

$\boxed{C \subset C'}$

$f(C') \geq f(C)$ (당연)

• diminishing return

철수: $C+A$ 미미: $C'+A$ ($C' \supset C$)

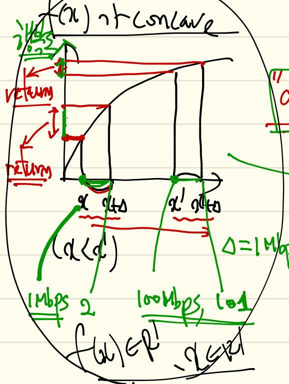
$$f(C+A) - f(C) \geq f(C'+A) - f(C')$$

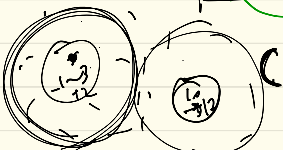
우리가 몇개가 이런상황?) diminishing return

↓ set function

f(x) 가 concave

"diminishing return"





Approximation

(increasing diminishing return을 가지는 set function의 max)

→ Submodular function (set)

greedy algorithm이 approximation을 잘하며, 분석도 많이 이뤄졌었다.

결과적으로

2 → 3

10 → 11

(Homework) Submodular set function의 처음 3개가 동치임을 증명하라

책 110 page, 9.1

(Homework) $F_t(C)$가 submodular 함을 증명하라.

$$F_t(C) = E(|V(C)|)$$ ⇒ 끝까지 연결성 있게. $\underline{V(C)}$ random set

$$F_t(C) = E(|V(C)|) \text{ : random variable}$$

Greedy algorithm

- Choose nodes, $v_1, \ldots, v_k \in V$ recursively as follows:
  in the i-th step,

$$v_i \in \operatorname*{argmax}_{v \in V \setminus C_{i-1}} \left(F_t(C_{i-1} \cup \{v\}) - F_t(C_{i-1}) \right), \quad i = 1, \ldots, k,$$

F_t로 계산

→ C_{i-1} 증가분 C_i

where $C_0 = \phi$ and $C_j = \{v_1, \ldots, v_j\}$, $j = 1, \ldots, k-1$.

증가분이 큰 node를 하나씩 해서

- Then $C = C_k$

좀더 generalize 하면? Define a relaxed, generalize version of the greedy algorithm, which we call (ϵ, δ)-greedy algorithm, as follows; "a class of greedy algorithm"

$\epsilon \in [0, 1]$

$\delta \geq 0$

We say that the sequence $(v_1, v_2, \ldots, v_k)$ is (ε, δ)-greedy if the following is satisfied: at each step i

$$F(C_{i-1} \cup \{v_{i}\}) - F(C_{i-1}) \geq \varepsilon \max_{v \in V \setminus C_{i-1}} \left(F(C_{i-1} \cup \{v\}) - F(C_{i-1}) \right) - \delta$$

$\varepsilon \in (0,1]$ → multiplicative suboptimality factor

δ → additive factor

$(1,0)$-greedy → original greedy

x^* → $\varepsilon x^* - \delta$

(Question)

~~Q1: Greedy algorithm의 성능?~~

Q2: (ε, δ)-greedy의 성능? → $\max F(C)$ 얼마나 가까운 solution을 produce하는가?

Thm 9.2 Consider a submodular function $F : 2^V \to \mathbb{R}$ that takes non-negative values and is non-decreasing. Let $(v_1, \ldots, v_k)$ be an (ε, δ)-greedy sequence. Then

$$\boxed{F(C_k) \geq (1 - e^{-\varepsilon}) \max_{C \subset V, |C| = k} F(C) - \frac{k\delta}{\varepsilon}}$$

이 정도 값까지는 보장할 수 있다.

Pf) Let us consider a subset $C = \{w_1, \ldots, w_k\}$ which is fixed.

(ε, δ)-greedy

(Lemma 1) For any index $i \in \{1, \dots, k-1\}$, we have:

$$\boxed{F_i(C) - F_i(C_{i+1}) \le \left(1 - \frac{\varepsilon}{k}\right)\left(F_i(C) - F_i(C_i)\right) + \delta}$$

where C_i is the set $\{v_1, \dots, v_k\}$ of the i-first element in an arbitrary (ε, δ)-greedy sequence.

$\Rightarrow$ 2번째

Pf

등비수열의 점화식

$$\boxed{a_{n+1} \le k a_n + l}$$

$$F_i(C) - F_i(C_k) \le \left(1 - \frac{\varepsilon}{k}\right)\left(F_i(C) - F_i(C_{k-1})\right) + \delta$$

$$\le \left(1 - \frac{\varepsilon}{k}\right)\left(\left(1 - \frac{\varepsilon}{k}\right)\left(F_i(C) - F_i(C_{k-2})\right) + \delta\right) + \delta$$

$$\vdots$$

$$\le \left(1 - \frac{\varepsilon}{k}\right)^k \left(F_i(C) - \overset{0}{F_i(C_0)}\right) + \delta \sum_{i=0}^{k-1} \left(1 - \frac{\varepsilon}{k}\right)^i$$

$$\le \left(1 - \frac{\varepsilon}{k}\right)^k F_i(C) + \delta \cdot \frac{1}{1 - 1 + \frac{\varepsilon}{k}} \rightarrow \frac{\delta k}{\varepsilon}$$

$$\le e^{-\varepsilon} F_i(C) + \frac{\delta k}{\varepsilon}$$

$\left(1 - \frac{\varepsilon}{k}\right)^k \le e^{-\varepsilon}$

극한정의, $k \to \infty$

$$F_i(C_k) \ge \max_C \left(1 - e^{-\varepsilon}\right) F_i(C) - \frac{\delta k}{\varepsilon}$$

arbitrary

$\square$

Proof of Lemma

$$F_i(C) - F_i(C_{i+1}) \le \left(1 - \frac{\varepsilon}{k}\right)\left(F_i(C) - F_i(C_i)\right) + \delta$$

$$F_i(C_{i+1}) - F_i(C_i) \ge \frac{\varepsilon}{k}\left(\boxed{F_i(C) - F_i(C_i)}\right) - \delta \quad \Leftarrow \text{증명.}$$

1번

증명의 핵심 이용성질 (i) F_i is monotone, submodular
(ii) $C_i \to (\varepsilon, \delta)$-greedy sequence 로 볼 때 성립

~~Since~~ ~~Since~~

$$\underbrace{F_i(C_{i+1})}_{= F_i(C_i \cup \{v_{i+1}\})} - F_i(C_i) \geq \varepsilon \cdot \left(F_i(C_i \cup \{\underbrace{\text{some node}}_{w_j}\}) - F_i(C_i) \right) - \delta \quad \text{—①}$$

for a given i,

Define
$$D_j = C_i \cup \{w_1, \dots, w_j\}$$
$$D_0 = C_i$$
$$D_k = C_i \cup C$$

$$\sum_{j=1}^{k} \left(F_i(D_j) - F_i(D_{j-1}) \right) = F_i(D_k) - F_i(D_0) = F_i(C_i \cup C) - F_i(C_i)$$

monotone

$$\geq F_i(C) - F_i(C_i)$$

$$\exists j^*, \text{ s.t } F_i(D_{j^*}) - F_i(D_{j^*-1}) \geq \frac{1}{k}\left(F_i(C) - F_i(C_i)\right)$$

$$\sum_{j=1}^{k} a_j \geq b \implies \exists j^*, \text{ s.t } a_{j^*} \geq \frac{b}{k}$$

For notational simplicity

$$\exists j, \text{ s.t } F_i(D_j) - F_i(D_{j-1}) \geq \frac{1}{k}\left(F_i(C) - F_i(C_i)\right) \quad \text{—②}$$

$$\underbrace{F_i(C_i \cup \{w_j\})}_{w_j \text{ 추가}} - F_i(C_i) \geq F_i(\underbrace{D_j}_{D_{j-1} \cup \{w_j\}}) - F_i(D_{j-1}) \quad \text{—③}$$

diminishing returns

w_j 추가

$$C_i \subset D_{j-1}$$

$$F_i(C_{i+1}) - F_i(C_i) \geq \frac{\varepsilon}{k}\left(F_i(C) - F_i(C_i)\right) - \delta$$

남아있는 것은?

????

$\bar{\Pi}(C) = E(|U(C)|)$

set C는 initial seed set으로 감염시켰을때, 감염된 최종 set들. ⇐ random

For a given set C, $\bar{\Pi}(C)$의 계산이 쉬운가?

$f(x) = x^2 - 2x + 3$

$f(1)$? ⟶ 쉽다.

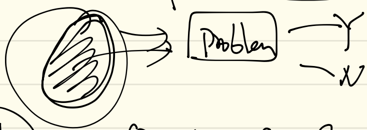
만약 어렵다면 greedy, (ε, δ)-greedy 활용(x)
(computation cost 이 많이 걸린다)

Computation Theory

Problem: 풀기 쉽다/어렵다.

Yes?
No? (optimization)

· certificate: 어떤 problem (decision) 에서 Yes를 produce하는 입력

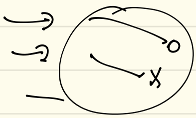


· NP class: difficulty of finding certificates
(즉, 주어진 input이 certificate이 되는지 않는지 판단이 어렵다) 복잡도가 크다.

a class of problem

반드시

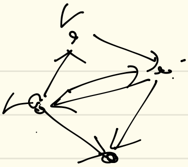
· #P class: counting problem의 difficulty
(sharp-P) (즉, certificate이 되는것이 몇개인지 알기가 어렵다)
→ exponential하기 때문?



· ~~BPP class~~

=

Example (Graph Reliability Problem)



- Directed Graph over n nodes
- Each node can fail w.p $\frac{1}{2}$

~~Que~~ (Q) Probability that node 1 has a path to n ~~nodes~~ node

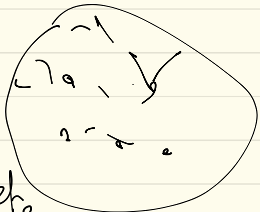
Under this simple failure model, the surviving graph is uniformly chosen at random from all ~~s~~ subgraphs of the original graph

$$= \frac{\#\text{ of subgraphs in which node 1 has a path to } n}{2^n}$$

Counting → 엄청 어렵다 ⟹ #P class

확률계산 → counting 문제

$f(c) = E(|V(C)|)$



$\boxed{|V(C)| = 100 \text{ 이 되는 확률}}$

→ $\boxed{\#P \text{ 문제}}$, 즉 $f(c)$의 계산이 어렵다.

← 확률계산이 어렵다 → (Sampling simulation)을 뽑는다.

Bernoulli

X $E(X)$? ~~Q~~ $p(X=1)$?

→ 동전던지기를 M번 반복 M ↑ 더 정확해진다.

Chernoff bound

F

$F_i(c)$의 sampling : M i.i.d sample

$$\hat{F_i}(c) = \frac{|U_1(c)| + |U_2(c)| + |U_3(c)| + \cdots + |U_M(c)|}{M}$$

조정, r r

(Q1) $F_i(c)$와 $\hat{F_i}(c)$의 차이는?

$\boxed{\text{Lemma 4}}$ (Chernoff bound)

$$P_r\left(|\hat{F_i}(c) - F_i(c)| \geq r F_i(c)\right) \leq 2 \cdot e^{-M h(r/m)} \quad M \to \infty \quad \to 0$$

$\boxed{U_i(c) : \{0, \ldots m\} \text{ valued R.V}}$ $\{0, 1\}$-valued R.V

수정

where $h(x) = (1+x)\log(1+x) - x$.

$\boxed{\text{Algorithm } \mathbb{P}}$. Greedy algorithm을 적용하는데 $F_i(c)$가 필요할때 M개의 i.i.d sample을 만들어서 $\hat{F_i}(c)$를 가지고 greedy scheme을 적용한다.

오차에 근한 suboptimality (optimal algorithm과 비교)

① F_i를 정확히 compute 할수있다는 가정을 하면 greedy

② F_i를 정확히 compute 할수 없기 때문에 sampling

(Q) $\mathbb{P}$의 성능은? (optimal algorithm에 비해서)

→ M을 얼마나 크게 선택해야 하는가? $M = n^4 \log n$

Prop 9.3 Let $r > 0$ be fixed. (내가 마음대로 선택할 수 있는 parameter)
Assume that M random graphs are generated whenever $F(c)$ needs to be computed. Assume that we use "pure greedy" with estimated $F(c)$ ($\hat{F}(c)$).

Then, with probability at least $1 - 2nk\, e^{-M h(r/n^2)}$ → 확신의 확률

$$F(c_k) \geq \left(1 - \frac{1}{e} - 4r - O\left(\frac{1}{n}\right)\right) \sup_{c \subset V,\, |c| = k} F(c)$$

algorithm의 결과물 — optimal 값 solution

($r \uparrow$ → 확신확률 증가한다
→ suboptimality gap은 커진다.)

$F(c)$의 r-거리

Lemma 2 Sampling을 통해서 $\hat{F}(c)$가 만들어진 $\left|\hat{F}(c) - F(c)\right| \leq \gamma F(c)$를

homework 마무리해 보기 (증명 과정 정리하기)

IP algorithm은 greedy algorithm $F(c)$

$\left(\frac{1-\gamma}{1+\gamma}, \frac{4\gamma_n}{1+\gamma}\right)$-greedy

ε δ Sequence를 생성해낸다

한 요소씩 추가할 때

증명: p113 중반 ~ p114 중반

Proof of 9.3

Choose γ, such that $\gamma = \frac{r}{n}$. (결국은 r을 choose 하겠다는 말과 같다.)

II4

Then, ~~w~~ at least w.p. $1 - 2nk\, e^{-Mh(r/n^2)}$

$$\hat{F}(C_k) \geq (1-e^{-\varepsilon}) \max_{C\subset V,\, |C|=k} \hat{F}(C) - \frac{k\delta}{\varepsilon}, \text{ where}$$

$$\varepsilon = \frac{1-\gamma}{1+\gamma}, \quad \delta = \frac{4\gamma n}{1+\gamma}. \qquad \gamma = \frac{r}{n}$$

Why (2.)

$$\boxed{1 - 2nk\, e^{-Mh(\gamma/n)}} \qquad \frac{r}{n^2}$$

ε, δ greedy 를 만들어내냐.

k번째 seed selection
at maximum n개를 $\hat{F}(C)$를 구해봐야
argmax

Note that for all C such that $|C|=k \Rightarrow \hat{F}(C) \geq k$

$$\geq (1-e^{-\varepsilon}) \max \hat{F}(C) - \frac{\hat{F}(C)\cdot\delta}{\varepsilon} \qquad \gamma = \frac{r}{n}$$

$$\boxed{\hat{F}(C_k) \geq \left(1-e^{-\varepsilon} - \frac{\delta}{\varepsilon}\right) \max \hat{F}(C)}$$

$$\varepsilon = \frac{1-\gamma}{1+\gamma}, \quad \delta = \frac{4\gamma n}{1+\gamma}$$

We choose $\varepsilon = 1 - O\left(\frac{1}{n}\right)$, $\frac{\delta}{\varepsilon} = 4r - O\left(\frac{1}{n}\right)$.

$$\boxed{1 - e^{-1} - 4r - O\left(\frac{1}{n}\right)}$$

$$\varepsilon = \frac{1-\frac{r}{n}}{1+\frac{r}{n}} = \frac{n-r}{n+r} = \frac{n+r-2r}{n+r} = 1 - \frac{2r}{n+r}$$

$$\frac{\delta}{\varepsilon} = \frac{\frac{4\gamma n}{1+\gamma}}{\frac{1-\gamma}{1+\gamma}} = \frac{4\gamma n}{1-\gamma} = \frac{4\frac{r}{n}n}{1-\frac{r}{n}} = \frac{4r}{1-\frac{r}{n}} = 4r + O\left(\frac{1}{n}\right)$$

(**Last Question**): $\boxed{\text{M은 어떻게 잡아야}}$ ~~to th~~ with high probability로 얼마나 어느 정도의 approximation을 보장할 수 있나?

Sampling complexity

$$h(x) = \bigcirc$$

$$h\left(r/n^2\right) = \Theta\left(\frac{r^2}{n^4}\right)$$

$$1 - 2n^k e^{-M\, h(r/n^2)}$$

$$n \cdot \boxed{e^{-M \cdot \frac{r^2}{n^4}}} \longrightarrow 0$$

Θ: $\frac{1}{n}$ 보다 작은 order

$$\boxed{M = n^4 \log n}$$

$$n \cdot e^{-\log n \ast r^2}$$

$$= n \cdot \left(\frac{1}{n}\right)^{r^2} = o(1)$$

$r^2 > 1$