Approx Algos

Topics

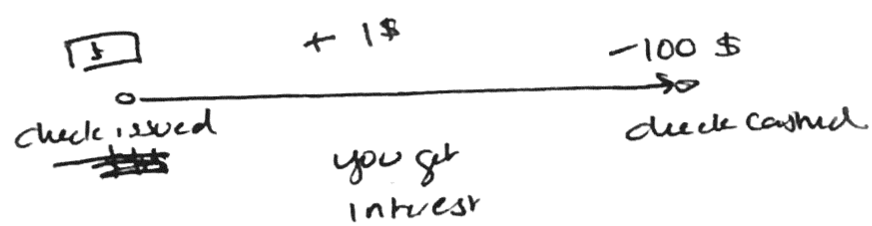
1. Monotone Submodular Maximization
2. Non-monotone submodular maximization.

[1] Setting -

Main setting consider these three problems.

(1) Maximizing Float in bank account.

- Prior to electronic checking you could technically accrue money the time you issue check and the time its cashed via interest.



So now imagine you are a company and you regularly need to make ~~P pay~~ folks in set P payments.

You can open at most k bank accounts among B banks and you want to open them to maximize the float you gain. (maybe from interest, time it takes to clear a check, friend of bank rewards ...).

<u>Input</u>:

1. P a set of payees
2. B a set of banks
3. k the max # of banks you can open before SEC is like delete this.
4. V_{ij} := float you get from paying $j \in P$ w/ bank $i \in B$.

<u>Output</u>:

A set $S \subseteq B$ such that S maximizes

$$f(S) = \sum_{j \in P} \underbrace{\max_{i \in S} V_{ij}}_{\text{always pick best bank in } S \text{ to pay } j \in P \text{ w/}}$$

[2] ~~Max~~ Maximizing Social Influence. (Kempe Kleinberg Tardos 2003)

Suppose you have a friend network. G if you want to model influence.

— You want to buy an iPad.

- If you see k fraction of your friends (nbrs) buy iPads. You will buy an iPad.

~~Now Apple, really wants a few folks to buy iPads.~~ everyone

- ~~Obama.~~
- ~~Satish~~
- ...

~~The question is, if you have only k ipads to~~ give for free ~~how do you distribute them in such a way~~

Apple has a couple free iPads to give. How do you distribute those k iPads so that you maximize the number of folks who will buy iPads?

Input:

- Social network G
- activation parameter α.
- max # of things to give k.

Output

- $S \subseteq V$ w/ $|S| \leq k$ maximizing # of folks who will be activated.

3 MaxCut

Input : $G = (V, E)$ a weighted graph.

Output : $S \subseteq V$ s.t.

$$w(S) = \sum_{e \in E(S, V-S)} w(e)$$

is maximized. Here

$$E(S, V-S) = \{ e = (u,v) : u \in S \Rightarrow v \notin S \}$$

What do all these problems have in common?

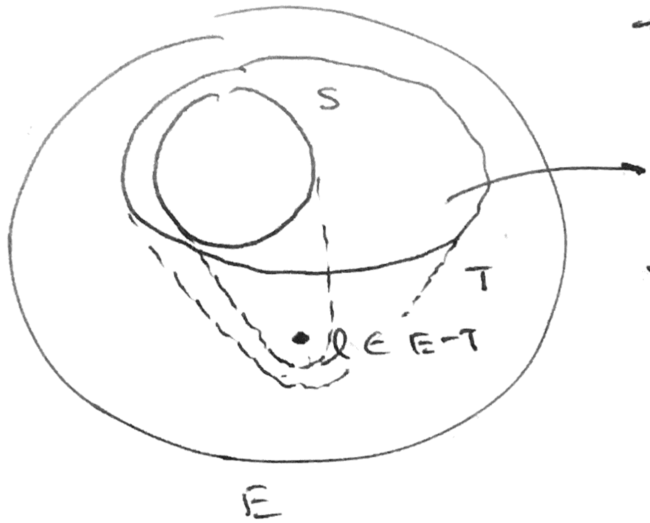
$\Rightarrow$ they're maximizing a sub-modular function

(Submodular): Let $E = \{e_1 \dots e_n\}$ be a ground set of elements. Let $f : 2^E \to \mathbb{R}_{\geq 0}$ be a function on subsets $S \subseteq E$ where $f(\emptyset) = 0$.

- Think of $S \subseteq E$ some selection of choices
- $f(S)$ is cost of S.
- $f(\emptyset) = 0$.

Now... f is submodular if $\forall S, T \subseteq E$ s.t. $S \subseteq T$ and $\ell \in E - T$.

$$f(T \cup \{\ell\}) - f(T) \leq f(S \cup \{\ell\}) - f(S)$$



$\underbrace{f(T\cup\{l\}) - f(T)}_{\Delta f \text{ if you add } l \text{ to } T}$

$\underbrace{f(S\cup\{l\}) - f(S)}_{\Delta S \text{ if you add } l \text{ to } S.}$

$$\underbrace{f(T\cup\{l\}) - f(T)}_{\text{you get less by adding } l \text{ to } T \text{ where } S\subseteq T.} \le f(S\cup\{l\}) - f(S)$$

Models "diminishing returns"

<u>Example:</u>

1. Maximizing Flow is submodular.

Here for $S \subseteq B \rightarrow f(S) = \sum_{j\in P} \max_{i\in S} V_{ij}$

$f(S\cup\{l\}) - f(S) =$

$$= \sum_{j\in P}\left(\max_{i\in S\cup\{l\}} V_{ij} - \max_{i\in S} V_{ij}\right)$$

if max is $i\in S$ then $\Delta = 0$ o.w. $V_{lj} > V_{ij}$ $\forall i \in S$.

$$= \sum_{j\in P} \max\left(0, V_{lj} - \max_{i\in S} V_{ij}\right)$$

Similarly

$$f(T\cup\{l\}) - f(T) = \sum_{j\in P} \max\left(0, V_{lj} - \max_{i\in T} V_{ij}\right)$$

However $S \subseteq T$.

$$\rightarrow \max_{i\in T} V_{ij} \ge \max_{i\in S} V_{ij}$$

→ This means

$$f(T \cup \{l\}) - f(T) = \sum_{j \in P} \max(0, v_{lj} - \max_{i \in T} v_{ij})$$

$$\leq \sum_{j \in P} \max(0, v_{lj} - \max_{i \in S} v_{ij})$$

because you subtract

$$= f(S \cup \{l\}) - f(S). \square$$

Maximizing Monotone Submodular Functions.

Furthermore $f(S)$ in maximizing float is monotone. you can only do better by adding more banks...

(Monotone): $f: 2^E \to \mathbb{R}_{\geq 0}$ is monotone if $\forall S, T \subseteq E$ s.t. $S \subseteq T$.

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

Thus... if we want to defraud banks... we should study maximizing monotone submodular functions w/ size constraints.

Problem: Maximizing monotone submodular functions w/ size constraint.

Input : $f : 2^E \to \mathbb{R}_{\geq 0}$ for E a ground set of

elements $E = \{e_1 \ldots e_m\}$

- f monotone
- f submodular.

Output : $S \subseteq E$ s.t. $|S| \leq k$ and $f(S)$ is maximized

$\Rightarrow$ $(1 + \frac{1}{e})$ - approximation $\to$ Cornuejols, Fisher, Nemhauser, 1978

$\hookrightarrow$ Greedy algorithm.

Algorithm :

1. $S^0 \leftarrow \emptyset$.
2. While $|S| \leq k$ do $t = 1 \ldots k$.
   - $i_t = \underset{i \in E}{\text{argmax}} \left(f(S \cup \{i\}) - f(S) \right)$
   - $S^t = S^{t-1} \cup \{i_t\}$.

Do the obvious thing...

$\Rightarrow$ Analysis:

First let me state a lemma w/o proof.

Lemma: $\forall S \subseteq E$ w/ $|S| \leq k$ let O be the optimal set of

elements... then

$$\max_{i \in E} \left(f(S \cup \{i\}) - f(S) \right) \geq \frac{1}{k} \left(f(O) - f(S) \right)$$

$\Rightarrow$ you'll always close a $\frac{1}{k}$ factor b/w cost of your soln

and the optimal.

Immediately you should think this is a $1-\frac{1}{e}$ approx because you're always closing a $\frac{1}{k}$ factor of the optimality gap.

Theorem: Algorithm returns a $1-\frac{1}{e}$ approx for monotone submodular maximization.

Proof: ~~Observe~~ let S be returned by alg and O opt.

$$
\begin{aligned}
f(S) &= f(S^k) \\
&= f(S^{k-1} \cup \{i_k\}) \\
&\geq \frac{1}{k} f(O) + \left(1-\frac{1}{k}\right) f(S^{k-1}) \quad \text{by lemma.} \\
&= \frac{1}{k} f(O) + \left(1-\frac{1}{k}\right) f(S^{k-2} \cup \{i_{k-1}\}) \\
&\geq \frac{1}{k} f(O) + \left(1-\frac{1}{k}\right)\left(\frac{1}{k} f(O) + \left(1-\frac{1}{k}\right) f(S^{k-2})\right) \\
&= \frac{1}{k} f(O)\left(1 + \left(1-\frac{1}{k}\right)\right) + \left(1-\frac{1}{k}\right)^2 f(S^{k-2}) \\
&= \dots \\
&= \frac{1}{k} f(O)\left(1 + \left(1-\frac{1}{k}\right) + \dots + \left(1-\frac{1}{k}\right)^{k-1}\right) \\
&= \frac{1}{k} f(O) \cdot \sum_{i=0}^{k-1} \left(1-\frac{1}{k}\right)^i \\
&= \frac{1}{k} f(O) \cdot \left(\frac{1-(1-1/k)^k}{1-(1-1/k)}\right) \\
&= f(O) \cdot \left(1-\left(1-\frac{1}{k}\right)^k\right)
\end{aligned}
$$

But we all know the theoretician's favorite inequality...

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

Thus...

$$\begin{aligned} & f(O)\cdot\left(1-\left(1-\tfrac{1}{k}\right)^k\right) \\ & \le f(O)\cdot\left(1-\left(e^{-1/k}\right)^k\right) \\ & = f(O)\cdot\left(1-\tfrac{1}{e}\right) \to \left(1-\tfrac{1}{e}\right)\cdot OPT. \quad \square \end{aligned}$$

Nice now to show the lemma...

<u>Lemma</u>: $\forall S \subseteq E$ where $|S| \le k$. let O be opt.

$$\max_{i\in E}\left(f(S\cup\{i\}) - f(S)\right) \ge \frac{1}{k}\left(f(O) - f(S)\right)$$

Okay why should this be true ~~in general~~

$\Rightarrow$ Think about float. and fix opt O.

$$\underbrace{\circ \quad \circ \quad \circ \quad \circ \quad \circ \quad \circ \quad \circ}_{k.} \quad \text{Banks.}$$

- allocate V_{i^*j} $\forall j \in P$ to i^* bank where

$$i^* = \arg\max_{i\in O} V_{ij}$$

- Since $|O| \le k$. at least one bank i' will have $\ge \frac{f(O)}{k}$ cost assigned.

now what i do we add to S on the first round

$$i_1 = \underset{i \in E}{\operatorname{argmax}} \left(f(S^0 \cup \{i\}) - f(S^0) \right)$$

$$= \underset{i \in E}{\operatorname{argmax}} \; f(\{i\})$$

And so we'll at least increase our utility

$$f(S^1) \geq \frac{f(O)}{k}.$$

Now after adding i_1, there's another bank i_2 st. its assigned

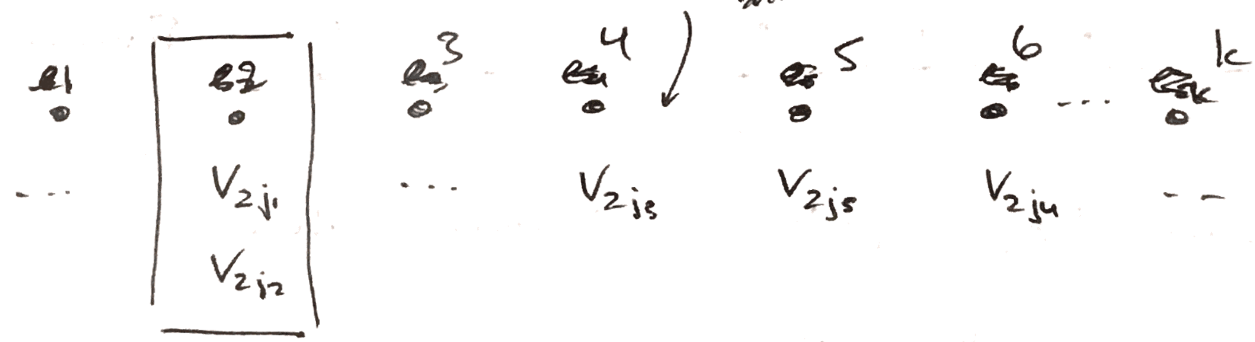
$$\frac{1}{k} \cdot \text{whatever is left overs}$$

And in particular.

$$\underbrace{f(S \cup \{i_2\}) - f(S)}_{\text{what I gain by adding } i_2.} \geq \frac{1}{k} \big(\underbrace{f(O) - f(S)}_{\text{whatever's leftover.}} \big)$$

Picture to draw

assigned based on $\underset{i \in O}{\operatorname{argmax}} V_{ij} \quad \forall j \in P.$



- one must have $\geq \frac{f(O)}{k}$ assigned. Say its e_2.
- ~~$f(e_2)$~~ $f(\{e_2\}) \geq \frac{f(O)}{k}$ since it ~~s also~~ also has v's from other accounts ill.

Now to prove the lemma.

Proof: Observe by monotonicity $O \subseteq O \cup S$.

$$f(O) \le f(O \cup S).$$

now let $O = \{i_1^*, i_2^*, \dots i_p^*\}$ $p \le k$.

$$\left.\begin{aligned} f(O \cup S) = f(S) + \sum_{j=1}^{p} \Big(& f(S \cup \{i_1^* \dots i_j^*\}) \\ & - f(S \cup \{i_1^* \dots i_{j-1}^*\}) \Big) \end{aligned}\right\} \text{telescoping sum.}$$

But by submodularity

$$\le f(S) + \sum_{j=1}^{p} \left(f(S \cup \{i_j^*\}) - f(S) \right)$$

because $S \subseteq S \cup \{i_1^* \dots i_{j-1}^*\}$.

$$\le f(S) + p \cdot \max_{i \in E} \left(f(S \cup \{i\}) - f(S) \right)$$

$$\le f(S) + k \cdot \max_{i \in E} \left(f(S \cup \{i\}) - f(S) \right)$$

just flip the $f(S)$ and k to other side. □

Approx Algos

Nonmonotone Submodular Maximization

Okay now think back to Maxcut.

$\Rightarrow$ This is not monotone.

Adding a vertex could drop the weight of the cut.

$\Rightarrow$ Previous algorithm would yield $1/3$ - approx iirc.

Double Greedy (Buchbinder, Feldman, Naor, Schwartz 12)

Notation

Let ...

$$X_i \subseteq \{1 \dots i\}$$

$$\hat{X}_i = \underbrace{X_i}_{\text{may not always be } \{1 \dots i\}} \cup \{i+1 \dots n\}$$

For example.

$X_0 = \emptyset \qquad \hat{X}_0 = E \qquad X_i \subseteq \hat{X}_i \qquad X_n = \hat{X}_n$

Additionally we will hold these quantities.

$$a_i = f(X_{i-1} \cup \{i\}) - f(X_{i-1})$$

Value of adding i to X.

$r_i = f(\hat{X}_{i-1} - \{i\}) - f(\hat{X}_{i-1})$

or the ~~cost~~ value of removing i from $\hat{X}_i$. So think of

$X_i :=$ Solution we hold at iteration i.

$\hat{X}_i :=$ things we could potentially add to X_i after iteration i.

<u>Algorithm: Double Greedy</u>

1. Let $X_0 = \emptyset$
2. For $i = 1 \ldots n$ do...
   - Compute a_i and r_i
   - If $a_i \geq 0$ and $r_i < 0$ then add i:

     $X_i = X_i \cup \{i\}$ $\quad (\hat{X}_i = \hat{X}_{i-1})$
   - If $r_i \geq 0$ and $a_i < 0$ then ignore i.

     $X_i = X_{i-1}$ $\quad (\hat{X}_i = \hat{X}_{i-1} - \{i\})$
   - If $a_i \geq 0$ and $r_i \geq 0$ ... flip a coin.

$$X_i = \begin{cases} X_{i-1} \cup \{i\} & \text{w/ prob } \dfrac{a_i}{a_i + r_i} \\ X_{i-1} & \text{w/ prob } \dfrac{r_i}{a_i + r_i} \end{cases}$$

Oh but wait... what if $a_i < 0$ and $r_i < 0$?

Claim: $a_i + r_i \ge 0 \quad \forall i$

Proof: Observe that $X_{i-1} \subseteq \hat{X}_{i-1} - \{i\}$

By submodularity...

$$\underbrace{f(\underbrace{\hat{X}_{i-1}}_{=(\hat{X}_{i-1} - \{i\}) \cup \{i\}}) - f(\hat{X}_{i-1} - \{i\})}_{= -r_i} \le \cancel{f(X_{i-1}) - f(X_{i-1}}\ \underbrace{f(X_{i-1} \cup \{i\}) - f(X_{i-1})}_{a_i}$$

Thus $-r_i \le a_i \rightarrow 0 \le a_i + r_i$. $\square$

Historical note: There were really complicated 0.41 and 0.42 ... algs. This however is able to achieve 1/2-approx.

Analysis

Let OPT denote the optimal set and

$$OPT_i = X_i \cup (OPT \cap \{i+1 \dots n\})$$

So...

$$OPT_0 = OPT \qquad OPT_n = X_n = \hat{X}_n$$

OPT_i is a sliding window from OPT to X_i.

Again we state lemma first then prove later.

<u>Lemma</u>: $\forall i = 1 \dots n$

diff what OPT did and what ALG did on ith step.

$$\mathbb{E}\{f(OPT_{i-1}) - f(OPT_i)\}$$
$$\leq \frac{1}{2}\mathbb{E}[(f(X_i) - f(X_{i-1})) + (f(\hat{X}_i) - f(\hat{X}_{i-1}))]$$

<u>Theorem</u>: Double greedy is a $\frac{1}{2}$-approx for non monotone submodular maximization.

Proof: Lemma telescoping sum...

$$\sum_{i=1}^{n} \mathbb{E}[f(OPT_{i-1}) - f(OPT_i)]$$
$$\leq \frac{1}{2}\sum_{i=1}^{n} \mathbb{E}[(f(X_i) - f(X_{i-1})) + (f(\hat{X}_i) - f(\hat{X}_{i-1}))] \quad \text{lemma}$$
$$= \frac{1}{2}\Big(\mathbb{E}[f(X_n)] - \underbrace{\mathbb{E}[f(X_0)]}_{\substack{\text{constant} \\ =0 \\ \text{since } f(X_0) = f(\emptyset) = 0.}} + \mathbb{E}[f(\hat{X}_n)] - \underbrace{\mathbb{E}[f(\hat{X}_0)]}_{\substack{\text{constant} \\ \geq 0 \\ \text{since } = f(E)}}\Big)$$
$$\leq \frac{1}{2}\Big(\mathbb{E}[f(X_n)] + \underbrace{\mathbb{E}[f(\hat{X}_n)]}_{=f(X_n)}\Big)$$
$$= \mathbb{E}[f(X_n)]$$

But $\sum_{i=1}^{n} \mathbb{E}[f(OPT_{i-1}) - f(OPT_i)]$ $\cancel{\mathbb{E}[f(OPT_0)]}$ is ...

$$= \underbrace{\mathbb{E}[f(OPT_0)]}_{\text{constant}} - \underbrace{\mathbb{E}[f(OPT_n)]}_{=X_n}$$
$$= f(OPT) - \mathbb{E}[f(X_n)]$$

thus ~~for~~ we have...

$$f(OPT) - \cancel{\mathbb{E}}\ \mathbb{E}[f(X_n)] \leq \mathbb{E}[f(X_n)]$$

$$f(OPT) \leq 2\,\mathbb{E}[f(X_n)]$$

$$\mathbb{E}[f(X_n)] \geq \frac{1}{2} f(OPT). \quad \square$$

Now to prove the lemma:

Lemma: $\forall i = 1 \dots n$

$$\mathbb{E}[f(OPT_{i-1}) - f(OPT_i)]$$
$$\leq \frac{1}{2} \mathbb{E}\left[(f(X_i) - f(X_{i-1})) + (f(\hat{X}_i) - f(\hat{X}_{i-1})) \right]$$

Proof: You literally check every if statement for every iteration for cases where $i \notin OPT$ and $i \in OPT$: Focus on $\underline{i \notin OPT}$.

<u>Case</u>: $a_i \geq 0,\ r_i < 0$: i added, no expectation needed

Observe then that...

- $X_i = X_{i-1} \cup \{i\}$
- $\hat{X}_i = \hat{X}_{i-1}$
- $OPT_i = OPT_{i-1} \cup \{i\}$ ← because you added i.
- $OPT_{i-1} \subseteq \hat{X}_i - \{i\}$ ← Since $i \notin OPT$, ~~$i \notin \hat{X}$~~ $i \notin OPT_{i-1}$

Thus by submodularity.

$$f(\underbrace{\hat{X}_{i-1}}) - f(\hat{X}_{i-1} - \{i\}) \le f(OPT_{i-1} \cup \{i\}) - f(OPT_{i-1})$$

$= (\hat{X}_{i-1} - \{i\}) \cup \{i\}$ since $OPT_{i-1} \subseteq \hat{X}_{i-1} - \{i\}$.

$$= f(OPT_i) - f(OPT_{i-1})$$

However $f(\hat{X}_{i-1}) - f(\hat{X}_{i-1} - \{i\}) = -r_i$ thus

$(*) \rightarrow$ $f(OPT_{i-1}) - f(OPT_i) \le r_i < 0$

$* -1$

Since by assumption $r_i < 0$. Meanwhile $a_i \ge 0$ thus

$$\begin{aligned} 0 &\le \tfrac{1}{2} a_i \\ &= \tfrac{1}{2}\left(f(X_i \cup \{i\}) - f(X_{i-1})\right) \\ &= \tfrac{1}{2}\left((f(X_i \cup \{i\}) - f(X_{i-1})) + \underbrace{(f(\hat{X}_i) - f(\hat{X}_{i-1}))}_{=0 \text{ since } \hat{X}_i = \hat{X}_{i-1}}\right) \end{aligned}$$

Thus we have ...

$$f(OPT_{i-1}) - f(OPT_i) \le \tfrac{1}{2}(\ldots)$$

Case 2: Suppose $a_i < 0$ and $r_i \ge 0$ i.e. i removed from consideration. Observe

- $X_i = X_{i-1}$ $\xrightarrow{\text{implies}}$ • $OPT_i = OPT_{i-1}$ ~~since~~
- $\hat{X}_i = \hat{X}_{i-1} - \{i\}$

Assuming $i \notin OPT$ ...

$$(**)$$

$$\mathbb{E}\left[f(OPT_{i-1}) - f(OPT_i)\right] = 0 \leq \tfrac{1}{2} r_i$$

$$= \tfrac{1}{2}\left(f(\hat{x}_{i-1} - \{i\}) - f(\hat{x}_{i-1})\right)$$

$$= \tfrac{1}{2}\left(\underbrace{(f(x_i) - f(x_{i-1}))}_{=0 \text{ since } x_i = x_{i-1}} + (f(\hat{x}_{i-1} - \{i\}) - f(\hat{x}_{i-1}))\right)$$

as required...

Case 3: $a_i \geq 0$ and $r_i \geq 0$ i.e coin tossed and expectation is required.

Observe that w/ prob $\frac{r_i}{r_i + a_i}$ we fall into case 2

w/prob $\frac{a_i}{a_i + r_i}$ we fall into case 1...

$$\mathbb{E}\left[f(OPT_{i-1}) - f(OPT_i)\right]$$

from $(*)$

from $(**)$ $\leq 0 \cdot \frac{r_i}{a_i + r_i} + (-r_i) \cdot \frac{a_i}{a_i + r_i}$

$$= -\frac{a_i r_i}{a_i + r_i}$$

Meanwhile the RHS has...

$$\tfrac{1}{2} \mathbb{E}\left[(f(x_i) - f(x_{i-1})) + (f(\hat{x}_i) - f(\hat{x}_{i-1}))\right]$$

$$= \tfrac{1}{2} \mathbb{E}\left[f(x_i) - f(x_{i-1})\right] + \tfrac{1}{2} \mathbb{E}\left[f(\hat{x}_i) - f(\hat{x}_{i-1})\right]$$

Observe from previous two cases...

$$\mathbb{E}[f(X_i) - f(X_{i-1})] = \frac{a_i^2}{a_i + r_i}$$

$= a_i$ w.p. $\frac{a_i}{a_i + r_i}$ since $X_i = X_{i-1} \cup \{i\}$

$= 0$ w.p. $\frac{r_i}{a_i + r_i}$ since $X_i = X_{i-1}$

$$\mathbb{E}[f(\hat{X}_i) - f(\hat{X}_{i-1})] = \frac{r_i^2}{a_i + r_i}$$

$= 0$ w.p. $\frac{a_i}{a_i + r_i}$ Since $X_i = X_{i-1} \cup \{i\}$ $\rightarrow \hat{X}_i = \hat{X}_{i-1}$

$= r_i$ w.p. $\frac{r_i}{a_i + r_i}$ Since $\hat{X}_i = \hat{X}_{i-1} - \{i\}$ if r_i flipped.

This means...

$$\frac{1}{2}\left(\mathbb{E}[f(X_i) - f(X_{i-1})] + \mathbb{E}[f(\hat{X}_i) - f(\hat{X}_{i-1})]\right)$$

$$= \frac{1}{2} \cdot \frac{a_i^2}{a_i + r_i} + \frac{1}{2} \cdot \frac{r_i^2}{a_i + r_i}$$

$$\geq \frac{a_i r_i}{a_i + r_i} \iff a_i^2 + r_i^2 \geq 2a_i r_i$$

$$\iff a_i^2 - 2a_i r_i + r_i^2 \geq 0 \checkmark$$

$$= \mathbb{E}[f(OPT_{i-1}) - f(OPT_i)]$$

As required. □

Congrats now you have to do this agum for i.d OPT oh well...

Why is a $\frac{1}{2}$ approximation important?

Theorem (Feige, Mirakni, Vondrak 2007):

No $(\frac{1}{2}+\epsilon)$-approximation for maximizing nonmonotone submodular functions exists for the oracle access model.

How do we even compute $f(S)$ anyway?

- Assume polytime oracle that provides value.

The way to prove such a result is to find two submodular functions w/ OPT factor 2 away from eachother Then show its like superpolynomial time to distinguish b/w the two.