Prophet Inequalities and Online Combinatorial Optimization

Vasilis Livanos

Theory Group
Department of Computer Science

livanos3@illinois.edu

April 21th, 2022
Overview

1. Prophets and Secretaries
   - The Secretary Problem
   - The Prophet Inequality Problem
   - Selecting Multiple Values

2. Online Combinatorial Optimization
   - Primer on Mathematical Programming
   - Online Contention Resolution Schemes

3. Equivalence via LP Duality

4. Variations and Open Problems
Consider \( n \) values \( v_1, \ldots, v_n \in \mathbb{R} \) arriving in random order.

Step \( i \): See value \( v_i \), immediately and irrevocably decide:

Select \( v_i \), or Skip.

Objective:

Maximize \( \Pr[\text{We select } v^*] \), where \( v^* = \max_{1 \leq i \leq n} v_i \).

Optimal strategy?
Consider $n$ values $v_1, \ldots, v_n \in \mathbb{R}$ arriving in random order.

Optimal strategy?
Consider $n$ values $v_1, \ldots, v_n \in \mathbb{R}$ arriving in random order.

Step $i$: See value $v_i$, **immediately** and **irrevocably** decide:
- Select $v_i$, or
- Skip.
Consider $n$ values $v_1, \ldots, v_n \in \mathbb{R}$ arriving in random order.

Step $i$: See value $v_i$, immediately and irrevocably decide:
- Select $v_i$, or
- Skip.

**Objective:** Maximize $\Pr [\text{We select } v^*]$, where $v^* = \max_{1 \leq i \leq n} v_i$. 
Consider $n$ values $v_1, \ldots, v_n \in \mathbb{R}$ arriving in random order.

Step $i$: See value $v_i$, immediately and irrevocably decide:
- Select $v_i$, or
- Skip.

Objective: Maximize $\Pr \left[ \text{We select } v^* \right]$, where $v^* = \max_{1 \leq i \leq n} v_i$.

Optimal strategy?
Secretary Problem (2/3)

Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^\ast$.

Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$.

Reject first $r$ values, for some $r$.

For $i > r$, accept first $v_i$ s.t. $v_i > \max_{1 \leq j \leq i-1} v_j$.

Example: Let $r = \frac{n}{2}$ and $v^\ast_2$ be the second-highest value. Then, $w.p. \frac{1}{2}$, $v^\ast_2$ is selected.

Thus, for $r = \frac{n}{2}$, $\Pr[\text{We select } v^\ast] \geq \frac{1}{4}$.

Optimal strategy: $r \approx n/e$. Then, $\Pr[\text{We select } v^\ast] \geq \frac{1}{e}$, and this bound is tight [Lin61; Dyn63].
Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$. 
Secretary Problem (2/3)

- Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$.
- Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$.
Secretary Problem (2/3)

- Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$.
- Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$.
  - Reject first $r$ values, for some $r$.
  - For $i > r$, accept first $v_i$ s.t. $v_i > \max_{1 \leq j \leq i-1} v_j$. 

Example: Let $r = \frac{n}{2}$ and $v^*_2$ be the second-highest value. Then, $v^*_2$ with probability $\frac{1}{2}$,

$\Pr[\text{we select } v^*_2] \geq \frac{1}{4}$.

Optimal strategy: $r \approx ne$. Then, $\Pr[\text{we select } v^*] \geq \frac{1}{e}$, and this bound is tight \cite{Lin61, Dyn63}. 

Prophet Inequalities and Online Combinatorial Optimization
April 21th, 2022 4 / 38
Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$. 

Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$. 

Reject first $r$ values, for some $r$. 

For $i > r$, accept first $v_i$ s.t. $v_i > \max_{1 \leq j \leq i-1} v_j$. 

Example: Let $r = \frac{n}{2}$ and $v^*_2$ be the second-highest value. Then,

1. w.p. $1/2$, 
2. w.p. $1/2$,
Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$.

Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$.

Reject first $r$ values, for some $r$. For $i > r$, accept first $v_i$ s.t. $v_i > \max_{1 \leq j \leq i-1} v_j$.

Example: Let $r = \frac{n}{2}$ and $v_2^*$ be the second-highest value. Then,

1. w.p. $1/2$,
2. w.p. $1/2$,

<table>
<thead>
<tr>
<th></th>
<th>$v^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_2^*$</td>
<td></td>
</tr>
</tbody>
</table>

In this case, we select $v^*$. Thus, for $r = \frac{n}{2}$,

$$\Pr[\text{We select } v^*] \geq \frac{1}{4}.$$
Secretary Problem (2/3)

- Never select $v_i$ if $v_i < \max_{1 \leq j \leq i-1} v_j$, because then certainly $v_i \neq v^*$.
- Decision at step $i$ can only depend on $\{v_1, \ldots, v_i\}$.
  - Reject first $r$ values, for some $r$.
  - For $i > r$, accept first $v_i$ s.t. $v_i > \max_{1 \leq j \leq i-1} v_j$.
- Example: Let $r = \frac{n}{2}$ and $v^*_2$ be the second-highest value. Then,
  - 1. w.p. 1/2,
  - 2. w.p. 1/2,
  
<table>
<thead>
<tr>
<th></th>
<th>$v^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v^*_2$</td>
<td></td>
</tr>
</tbody>
</table>

In this case, we select $v^*$. Thus, for $r = \frac{n}{2}$,

$$\Pr \left[ \text{We select } v^* \right] \geq \frac{1}{4}.$$ 

- Optimal strategy: $r \approx \frac{n}{e}$. Then, $\Pr \left[ \text{We select } v^* \right] \geq \frac{1}{e}$, and this bound is tight [Lin61; Dyn63].
Prophet Inequality Problem (1/4)

Non-random order?
Adversarial order
\[ \Rightarrow \]
Arbitrary values

Assume adversarial order, but
\( v_i \sim D_i \), where
\( D_i \) is known
\[ \Rightarrow \]
Prophet Inequality Problem
Prophet Inequality Problem (1/4)

- Non-random order?
Prophet Inequality Problem (1/4)

- Non-random order?
- \textit{Adversarial order} + \textit{Arbitrary values} \implies \Pr [\text{We select } v^*] \text{ arbitrarily small.}
Non-random order?

*Adversarial order + Arbitrary values* $\implies$ $\Pr \left[\text{We select } v^*\right]$ arbitrarily small.

Assume adversarial order, but $v_i \sim D_i$, where $D_i$ is known $\implies$ *Prophet Inequality Problem*. 
Given \( n \) r.v.'s \( X_1, \ldots, X_n \sim D_1, \ldots, D_n \). We see independent realizations of \( X_i \)'s in adversarial order.

Step \( i \):
1. Immediately and irrevocably decide 1 select realization of \( X_i \) and stop, or
2. Ignore realization of \( X_i \) and continue to step \( i + 1 \).

Compare against \( E[\max_{n i=1} X_i] \).

\[ \exists \text{ algorithm s.t. } E[\text{ALG}] \geq \frac{1}{2} E[\max_{n i=1} X_i], \] and no algorithm can achieve better competitive ratio [KS77].
Given $n$ r.v.'s $X_1, \ldots, X_n \sim D_1, \ldots, D_n$. We see independent realizations of $X_i$'s in adversarial order.
Given $n$ r.v.'s $X_1, \ldots, X_n \sim D_1, \ldots, D_n$. We see independent realizations of $X_i$'s in adversarial order.

Step $i$: immediately and irrevocably decide

1. select realization of $X_i$ and stop, or
2. ignore realization of $X_i$ and continue to step $i + 1$. 
Given $n$ r.v.'s $X_1, \ldots, X_n \sim D_1, \ldots, D_n$. We see independent realizations of $X_i$'s in adversarial order.

**Step $i$: immediately and irrevocably decide**

1. select realization of $X_i$ and stop, or
2. ignore realization of $X_i$ and continue to step $i + 1$.

Compare against $\mathbb{E} \left[ \max_{i=1}^n X_i \right]$. 
Given $n$ r.v.'s $X_1, \ldots, X_n \sim D_1, \ldots, D_n$. We see independent realizations of $X_i$'s in adversarial order.

Step $i$: immediately and irrevocably decide

1. select realization of $X_i$ and stop, or
2. ignore realization of $X_i$ and continue to step $i + 1$.

Compare against $\mathbb{E} \left[ \max_{i=1}^n X_i \right]$.

$\exists$ algorithm s.t. $\mathbb{E}[\text{ALG}] \geq \frac{1}{2} \mathbb{E} \left[ \max_{i=1}^n X_i \right]$, and no algorithm can achieve better competitive ratio [KS77].
Let’s Play: Prophet Inequality Problem (3/4)

\[ X_1 = 2.74 \]
\[ X_2 = 3.75 \]
\[ X_3 = 2.81 \]
\[ X_4 = 5.66 \]

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022
Let’s Play: Prophet Inequality Problem (3/4)

- $X_1 = 2.74$

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022
Let’s Play: Prophet Inequality Problem (3/4)

\[ X_1 = 2.74 \]
\[ X_2 = 3.75 \]
Let's Play: Prophet Inequality Problem (3/4)

- $X_1 = 2.74$
- $X_2 = 3.75$
- $X_3 = 2.81$
Let’s Play: Prophet Inequality Problem (3/4)

- $X_1 = 2.74$
- $X_2 = 3.75$
- $X_3 = 2.81$
- $X_4 = 5.66$
Prophet Inequality Problem (4/4)

Simple problem in optimal stopping theory.

\[ \exists \text{ many algorithms for } \frac{1}{2} \text{-competitive ratio.} \]

A \( T \): “Fixed-Threshold” Algorithm

Set threshold \( T \) based on \( D_1, \ldots, D_n \). Accept the first \( X_i \geq T \).

What threshold to set?

1. Set \( T = \text{median of the distribution of } X^* \), i.e. \( \Pr[X^* \geq T] = \frac{1}{2} \) (assuming no point mass on \( T \)) [Sam84].

2. Set \( T = \frac{1}{2} \mathbb{E}[X^*] \) [KW12].
Prophet Inequality Problem (4/4)

- Simple problem in optimal stopping theory. ∃ many algorithms for 1/2-competitive ratio.

- **Fixed-Threshold** Algorithm
  - Set threshold $T$ based on $D_1, \ldots, D_n$. Accept the first $X_i \geq T$.
  - What threshold to set?
  - 1. Set $T = \text{median of the distribution of } X^*$, i.e. $\Pr[X^* \geq T] = \frac{1}{2}$ (assuming no point mass on $T$) [Sam84].
  - 2. Set $T = \frac{1}{2} \mathbb{E}[X^*]$ [KW12].
Simple problem in optimal stopping theory. Exists many algorithms for 1/2-competitive ratio.

\( A_T \): “Fixed-Threshold” Algorithm

Set threshold \( T \) based on \( D_1, \ldots, D_n \). Accept the first \( X_i \geq T \).

What threshold to set?
Simple problem in optimal stopping theory. ∃ many algorithms for 1/2-competitive ratio.

**A*T: “Fixed-Threshold” Algorithm**

Set threshold $T$ based on $D_1, \ldots, D_n$. Accept the first $X_i \geq T$.

What threshold to set?

1. Set $T =$ median of the distribution of $X^*$, i.e. $\Pr[X^* \geq T] = \frac{1}{2}$ (assuming no point mass on $T$) [Sam84].
Simple problem in optimal stopping theory. \exists many algorithms for 1/2-competitive ratio.

\[ A_T: \text{“Fixed-Threshold” Algorithm} \]

Set threshold \( T \) based on \( D_1, \ldots, D_n \). Accept the first \( X_i \geq T \).

What threshold to set?

1. Set \( T = \text{median of the distribution of } X^*, \) i.e. \( \Pr[X^* \geq T] = \frac{1}{2} \) (assuming no point mass on \( T \)) [Sam84].
2. Set \( T = \frac{1}{2} \mathbb{E}[X^*] \) [KW12].
Proof of the Prophet Inequality

Pr[\(X^* \geq T\)] = \(\frac{1}{2}\). Let \(\mathcal{E}_i\) be the event we “reach” the \(i\)-th random variable.
Proof of the Prophet Inequality

\[ \Pr[X^* \geq T] = \frac{1}{2}. \] Let \( \mathcal{E}_i \) be the event we “reach” the \( i \)-th random variable.

\[
\mathbb{E}[\text{ALG}] = T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot \mathbb{E}[(X_i - T)^+] 
\]
Proof of the Prophet Inequality

\( \Pr[X^* \geq T] = \frac{1}{2} \). Let \( \mathcal{E}_i \) be the event we “reach” the \( i \)-th random variable.

\[
\mathbb{E}[\text{ALG}] = T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot \mathbb{E}[(X_i - T)^+] \\
= T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr \left[ \max_{1 \leq j \leq i-1} X_j < T \right] \cdot \mathbb{E}[(X_i - T)^+] 
\]
Proof of the Prophet Inequality

\[ \Pr[X^* \geq T] = \frac{1}{2}. \]

Let \( \mathcal{E}_i \) be the event we “reach” the \( i \)-th random variable.

\[
\mathbb{E}[\text{ALG}] = T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot \mathbb{E}[(X_i - T)^+] \\
= T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr \left[ \max_{1 \leq j \leq i-1} X_j < T \right] \cdot \mathbb{E}[(X_i - T)^+] \\
\geq T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[X^* < T] \cdot \mathbb{E}[(X_i - T)^+] 
\]
Proof of the Prophet Inequality

Pr[X* ≥ T] = \frac{1}{2}. Let \mathcal{E}_i be the event we “reach” the i-th random variable.

E[ALG] = T \Pr[X^* ≥ T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot E[(X_i - T)^+]

= T \Pr[X^* ≥ T] + \sum_{i=1}^{n} \Pr \left[ \max_{1 ≤ j ≤ i-1} X_j < T \right] \cdot E[(X_i - T)^+]

≥ T \Pr[X^* ≥ T] + \sum_{i=1}^{n} \Pr[X^* < T] \cdot E[(X_i - T)^+]

= \frac{1}{2} T + \frac{1}{2} E \left[ \sum_{i=1}^{n} (X_i - T)^+ \right]
Proof of the Prophet Inequality

\( \Pr[X^* \geq T] = \frac{1}{2} \). Let \( \mathcal{E}_i \) be the event we “reach” the \( i \)-th random variable.

\[
\mathbb{E}[\text{ALG}] = T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot \mathbb{E}[(X_i - T)^+] \\
= T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr \left[ \max_{1 \leq j \leq i-1} X_j < T \right] \cdot \mathbb{E}[(X_i - T)^+] \\
\geq T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[X^* < T] \cdot \mathbb{E}[(X_i - T)^+] \\
= \frac{1}{2} T + \frac{1}{2} \mathbb{E} \left[ \sum_{i=1}^{n} (X_i - T)^+ \right] \\
\geq \frac{1}{2} T + \frac{1}{2} \mathbb{E} [(X^* - T)^+]
Proof of the Prophet Inequality

\[ \Pr[X^* \geq T] = \frac{1}{2}. \]

Let \( \mathcal{E}_i \) be the event we “reach” the \( i \)-th random variable.

\[
\mathbb{E}[\text{ALG}] = T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[\mathcal{E}_i] \cdot \mathbb{E}[(X_i - T)^+] \\
= T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr \left[ \max_{1 \leq j \leq i-1} X_j < T \right] \cdot \mathbb{E}[(X_i - T)^+] \\
\geq T \Pr[X^* \geq T] + \sum_{i=1}^{n} \Pr[X^* < T] \cdot \mathbb{E}[(X_i - T)^+] \\
= \frac{1}{2} T + \frac{1}{2} \mathbb{E} \left[ \sum_{i=1}^{n} (X_i - T)^+ \right] \\
\geq \frac{1}{2} T + \frac{1}{2} \mathbb{E} [(X^* - T)^+] \\
\geq \frac{1}{2} \mathbb{E}[X^*] \\
\]
\( \frac{1}{2} \) is Tight

Consider \( X_1 \) and \( X_2 \), where \( X_1 = 1 \) w.p. 1, and
\[
X_2 = \begin{cases} 
1 & \text{w.p. } \epsilon \\
0 & \text{w.p. } 1 - \epsilon,
\end{cases}
\]
for some small \( \epsilon > 0 \).

For every algorithm, \( E[\text{ALG}] = 1 \), regardless of which element it picks.

Expected value of the prophet is
\[
E[X^*] = 1 \epsilon + 1 \cdot (1 - \epsilon) = 2 - \epsilon.
\]
Consider $X_1$ and $X_2$, where

$$X_1 = 1 \text{ w.p. } 1, \text{ and } X_2 = \begin{cases} \frac{1}{\varepsilon} & \text{w.p. } \varepsilon \\ 0 & \text{w.p. } 1 - \varepsilon \end{cases},$$

for some small $\varepsilon > 0$. 

$\frac{1}{2}$ is Tight
\( \frac{1}{2} \) is Tight

- Consider \( X_1 \) and \( X_2 \), where

\[
X_1 = 1 \quad \text{w.p. 1, and } X_2 = \begin{cases} 
\frac{1}{\varepsilon} \quad \text{w.p. } \varepsilon \\
0 \quad \text{w.p. } 1 - \varepsilon
\end{cases}
\]

for some small \( \varepsilon > 0 \).

- For every algorithm, \( \mathbb{E}[\text{ALG}] = 1 \), regardless of which element it picks.
Consider $X_1$ and $X_2$, where

$$X_1 = 1 \text{ w.p. } 1,$$

and

$$X_2 = \begin{cases} 
\frac{1}{\varepsilon} & \text{w.p. } \varepsilon \\
0 & \text{w.p. } 1 - \varepsilon ,
\end{cases}$$

for some small $\varepsilon > 0$.

For every algorithm, $E[\text{ALG}] = 1$, regardless of which element it picks.

Expected value of the prophet is

$$E[X^*] = \frac{1}{\varepsilon} \cdot \varepsilon + 1 \cdot (1 - \varepsilon) = 2 - \varepsilon.$$
Natural generalization: Accept \( \leq k \) values, for given \( k \).

Compare against \( \text{OPT} = E \left[ \max_S: |S| \leq k \sum_{i \in S} X_i \right] \).

We differentiate between fixed-threshold and adaptive-threshold algorithms.

\[ A: \text{Adaptive-Threshold Algorithm} \]\n
\( \forall i \in [n] \), at step \( i \), set threshold \( T_i \), based on \( D_1, ..., D_n \) and \( X_1, ..., X_{i-1} \).

Accept every \( X_i \geq T_i \) until \( k \) values selected.
Natural generalization: Accept $\leq k$ values, for given $k$. 
Natural generalization: Accept $\leq k$ values, for given $k$.

Compare against $OPT = \mathbb{E} \left[ \max_{S: |S| \leq k} \sum_{i \in S} X_i \right]$. 

we differentiate between fixed-threshold and adaptive-threshold algorithms.

A: Adaptive-Threshold Algorithm

$\forall i \in [n]$, at step $i$, set threshold $T_i$, based on $D_1,...,D_n$ and $X_1,...,X_{i-1}$.

Accept every $X_i \geq T_i$ until $k$ values selected.
- Natural generalization: Accept $\leq k$ values, for given $k$.
- Compare against $OPT = \mathbb{E} \left[ \max_{|S| \leq k} \sum_{i \in S} X_i \right]$.
- We differentiate between fixed-threshold and adaptive-threshold algorithms.
Natural generalization: Accept $\leq k$ values, for given $k$.

Compare against $OPT = \mathbb{E} \left[ \max_{S: |S| \leq k} \sum_{i \in S} X_i \right]$.

We differentiate between fixed-threshold and adaptive-threshold algorithms.

**A: Adaptive-Threshold Algorithm**

$\forall i \in [n]$, at step $i$, set threshold $T_i$, based on $D_1, \ldots, D_n$ and $X_1, \ldots, X_{i-1}$. Accept every $X_i \geq T_i$ until $k$ values selected.
Fixed-Threshold Algorithm for $k$-Prophet

Simple fixed-threshold algorithm: $1 - O(\sqrt{\log k / k})$-competitive ratio.

Idea: Set threshold $T$ such that $E[|X_i \geq T|] = k - \delta$ for some $\delta$.

Use Hoeffding bound to show that, for $\delta = \sqrt{2k \log k}$, w.h.p.

$k - 2\delta \leq |X_i \geq T| \leq k$.

For fixed realizations, let $S_T = \{i \in [n] | X_i \geq T\}$. Then

$\sum_{i \in S_T} X_i = \sum_{i \in S_T} T + \sum_{i \in S_T} (X_i - T) = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T)$.

Simple algebra shows that

$E\left[\sum_{i \in S_T} X_i\right] \geq \left(1 - \frac{2\delta}{k}\right)OPT = \left(1 - \sqrt{\frac{2\log k}{k}}\right)OPT$. 

Prophet Inequalities and Online Combinatorial Optimization
April 21th, 2022 12 / 38
Simple fixed-threshold algorithm: $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

Idea:
Set threshold $T$ such that $E[|X_i \geq T|] = k - \delta$ for some $\delta$.

Use Hoeffding bound to show that, for $\delta = \sqrt{2k \log k}$, w.h.p.

$k - 2\delta \leq |X_i \geq T| \leq k$.

For fixed realizations, let $S_T = \{i \in [n] | X_i \geq T\}$. Then

$\sum_{i \in S_T} X_i = \sum_{i \in S_T} T + \left(\sum_{i \in S_T} (X_i - T)\right)$.

Simple algebra shows that $E\left[\sum_{i \in S_T} X_i\right] \geq \left(1 - 2\delta/k\right)OPT = \left(1 - \sqrt{2 \log k/k}\right)OPT$. 
Fixed-Threshold Algorithm for $k$-Prophet

- Simple fixed-threshold algorithm: $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.
- **Idea**: Set threshold $T$ s.t. $\mathbb{E}[|X_i \geq T|] = k - \delta$ for some $\delta$. 

\[
\mathbb{E}\left[\sum_{i \in S_T} X_i \right] \geq (1 - 2\delta) \cdot \text{OPT} = \left(1 - \sqrt{\frac{8 \log k}{k}}\right) \cdot \text{OPT}.
\]
Simple fixed-threshold algorithm: $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

**Idea:** Set threshold $T$ s.t. $\mathbb{E}[|X_i \geq T|] = k - \delta$ for some $\delta$.

Use Hoeffding bound to show that, for $\delta = \sqrt{2k \log k}$, w.h.p.

$$k - 2\delta \leq |X_i \geq T| \leq k.$$
Fixed-Threshold Algorithm for $k$-Prophet

- Simple fixed-threshold algorithm: $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

- **Idea:** Set threshold $T$ s.t. $E[|X_i \geq T|] = k - \delta$ for some $\delta$.

- Use Hoeffding bound to show that, for $\delta = \sqrt{2k \log k}$, w.h.p.

  \[ k - 2\delta \leq |X_i \geq T| \leq k. \]

- For fixed realizations, let $S_T = \{i \in [n] \mid X_i \geq T\}$. Then

  \[
  \sum_{i \in S_T} X_i = \sum_{i \in S_T} T + (X_i - T) = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T).
  \]
Fixed-Threshold Algorithm for $k$-Prophet

- Simple fixed-threshold algorithm: $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

- **Idea:** Set threshold $T$ s.t. $\mathbb{E}[|X_i \geq T|] = k - \delta$ for some $\delta$.

- Use Hoeffding bound to show that, for $\delta = \sqrt{2k \log k}$, w.h.p.

$$k - 2\delta \leq |X_i \geq T| \leq k.$$ 

For fixed realizations, let $S_T = \{i \in [n] \mid X_i \geq T\}$. Then

$$\sum_{i \in S_T} X_i = \sum_{i \in S_T} T + (X_i - T) = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T).$$

Simple algebra shows that

$$\mathbb{E} \left[ \sum_{i \in S_T} X_i \right] \geq \left(1 - \frac{2\delta}{k}\right) OPT = \left(1 - \sqrt{\frac{8 \log k}{k}}\right) OPT.$$
Adaptive-Threshold Algorithms

Adaptive-threshold algorithms can do better: $1 - \frac{1}{\sqrt{k}} + 3$ [Ala14], is asymptotically tight. Tight competitive ratio for every $k \geq 1$ [JMZ22] (complicated LP duality argument).
Adaptive-threshold algorithms can do better:

- $1 - \frac{1}{\sqrt{k+3}}$ [Ala14], is asymptotically tight.

- Tight competitive ratio for every $k \geq 1$ [JMZ22] (complicated LP duality argument).
General Feasibility Constraints

Given family $F$ of subsets of $n$-independent sets.

Select set $S$ of r.v.'s to maximize sum of values, subject to $S$ being independent.

Compare against $E\left[\max_{S \in F} \sum_{i \in S} X_i\right]$.

Examples:

1. Matroids: Uniform ($F = \{S \subseteq \{1, \ldots, n\} \mid |S| \leq k\}$), Graphic ($\{1, \ldots, n\} \rightarrow$ edges, $F \rightarrow$ forests), Vector ($\{1, \ldots, n\} \rightarrow$ vectors, $F \rightarrow$ lin. ind. vectors), etc.

2. Matchings: Given $G = (V, E)$, $\{1, \ldots, n\} \rightarrow E$ and $F \rightarrow$ matchings in $G$.

3. Knapsack: Given sizes $s_i \in \{0, 1\}$ for each $X_i$, $F = \{S \subseteq \{1, \ldots, n\} \mid \sum_{i \in S} s_i \leq 1\}$.

Matroid Prophet Inequality Theorem [KW12]

For every matroid $M$, $\exists$ an algorithm for the matroid prophet inequality problem that returns an independent set $S$ s.t. $E\left[\sum_{i \in S} X_i\right] \geq \frac{1}{2} \cdot E\left[\max_{S \in F} \sum_{i \in S} X_i\right]$.
General Feasibility Constraints

- Given family $\mathcal{F}$ of subsets of $[n]$ - independent sets.
General Feasibility Constraints

- Given family $\mathcal{F}$ of subsets of $[n]$ - *independent sets*.
- Select set $S$ of r.v.'s to maximize sum of values, subject to $S$ being independent.

Matroids: Uniform ($\mathcal{F} = \{S \subseteq [n] || |S| \leq k\}$), Graphic ($[n] \rightarrow \text{edges}$, $\mathcal{F} \rightarrow \text{forests}$), Vector ($[n] \rightarrow \text{vectors}$, $\mathcal{F} \rightarrow \text{lin. ind. vectors}$), etc.

Matchings: Given $G = (V, E)$, $[n] \rightarrow E$ and $\mathcal{F} \rightarrow \text{matchings in } G$.

Knapsack: Given sizes $s_i \in [0, 1]$ for each $X_i$, $\mathcal{F} = \{S \subseteq [n] \mid \sum_{i \in S} s_i \leq 1\}$.

Matroid Prophet Inequality Theorem [KW12]

For every matroid $M$, $\exists$ an algorithm for the matroid prophet inequality problem that returns an independent set $S$ s.t.

$$E\left[\sum_{i \in S} X_i\right] \geq \frac{1}{2} \cdot E\left[\max_{S \in \mathcal{F}} \sum_{i \in S} X_i\right].$$
General Feasibility Constraints

- Given family $\mathcal{F}$ of subsets of $[n]$ - independent sets.
- Select set $S$ of r.v.'s to maximize sum of values, subject to $S$ being independent.
- Compare against $\mathbb{E} \left[ \max_{S \in \mathcal{F}} \sum_{i \in S} X_i \right]$.
General Feasibility Constraints

- Given family $\mathcal{F}$ of subsets of $[n]$ - independent sets.
- Select set $S$ of r.v.'s to maximize sum of values, subject to $S$ being independent.
- Compare against $\mathbb{E} \left[ \max_{S \in \mathcal{F}} \sum_{i \in S} X_i \right]$.
- Examples:

  1. **Matroids**: Uniform ($\mathcal{F} = \{ S \subseteq [n] \mid |S| \leq k \}$), Graphic ([n] → edges, $\mathcal{F}$ → forests), Vector ([n] → vectors, $\mathcal{F}$ → lin. ind. vectors), etc.
  2. **Matchings**: Given $G = (V, E)$, $[n] \rightarrow E$ and $\mathcal{F}$ → matchings in $G$.
  3. **Knapsack**: Given sizes $s_i \in [0, 1]$ for each $X_i$, $\mathcal{F} = \{ S \subseteq [n] \mid \sum_{i \in S} s_i \leq 1 \}$. 
General Feasibility Constraints

- Given family $\mathcal{F}$ of subsets of $[n]$ - independent sets.
- Select set $S$ of r.v.'s to maximize sum of values, subject to $S$ being independent.
- Compare against $\mathbb{E} \left[ \max_{S \in \mathcal{F}} \sum_{i \in S} X_i \right]$.
- Examples:
  1. Matroids: Uniform ($\mathcal{F} = \{S \subseteq [n] \mid |S| \leq k\}$), Graphic ($[n] \to$ edges, $\mathcal{F} \to$ forests), Vector ($[n] \to$ vectors, $\mathcal{F} \to$ lin. ind. vectors), etc.
  2. Matchings: Given $G = (V, E)$, $[n] \to E$ and $\mathcal{F} \to$ matchings in $G$.
  3. Knapsack: Given sizes $s_i \in [0, 1]$ for each $X_i$, $\mathcal{F} = \{S \subseteq [n] \mid \sum_{i \in S} s_i \leq 1\}$.

Matroid Prophet Inequality Theorem [KW12]

For every matroid $\mathcal{M}$, $\exists$ an algorithm for the matroid prophet inequality problem that returns an independent set $S$ s.t.

$$
\mathbb{E} \left[ \sum_{i \in S} X_i \right] \geq \frac{1}{2} \cdot \mathbb{E} \left[ \max_{S \in \mathcal{F}} \sum_{i \in S} X_i \right].
$$
1 Prophets and Secretaries
   - The Secretary Problem
   - The Prophet Inequality Problem
   - Selecting Multiple Values

2 Online Combinatorial Optimization
   - Primer on Mathematical Programming
   - Online Contention Resolution Schemes

3 Equivalence via LP Duality

4 Variations and Open Problems
Given set $N$ of elements, $|N| = n$, and set function $f: 2^N \rightarrow \mathbb{R}$.

Goal: Optimize function $f$ under constraints.

Arguments of $f$: Variables.

Examples:
- $\min \sum_{i \in N} w_i x_i$
- $\max \sum_{e \in E(G)} w_e x_e$

s.t. $\sum_{i \in N} x_i \geq 1$ or $\sum_{e \in \delta(u)} x_e \leq 1$, $\forall u \in V(G)$

$x_i \in \{0, 1\}, \forall i \in N$

$x_e \in \{0, 1\}, \forall e \in E(G)$
Given set $N$ of elements, $|N| = n$, and set function $f : 2^N \rightarrow \mathbb{R}$.

**Goal:** Optimize function $f$ under constraints.

**Arguments of $f$: Variables.**
Given set \( N \) of elements, \(|N| = n\), and set function \( f : 2^N \to \mathbb{R} \).

**Goal:** Optimize function \( f \) under constraints.

Arguments of \( f \): **Variables**.

**Examples:**

\[
\begin{align*}
\text{min} & \quad \sum_{i \in N} w_i x_i \\
\text{s.t.} & \quad \sum_{i \in N} x_i \geq 1
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E(G)} w_e x_e \\
\text{s.t.} & \quad \sum_{e \in \delta(u)} x_e \leq 1, \quad \forall u \in V(G)
\end{align*}
\]

\( x_i \in \{0, 1\} \), \( \forall i \in N \)  

\( x_e \in \{0, 1\} \), \( \forall e \in E(G) \)
Linear Programs

How to solve IPs? ⇒ Relax constraints to continuous variables.

Most common relaxation of IPs: Linear Program (LP).

Constraint ⇒ Intersection of half-spaces ≡ Convex polytope $P$.

Examples:

\[
\begin{align*}
\min & \sum_{i \in N} w_i x_i \\
\text{s.t.} & \sum_{i \in N} x_i \geq 1 \quad \text{or} \quad \sum_{e \in \delta(u)} x_e \leq 1, \quad \forall u \in V(G) \\
& x_i \in [0, 1], \quad \forall i \in N \\
& x_e \in [0, 1], \quad \forall e \in E(G)
\end{align*}
\]
How to solve IPs? \(\Longrightarrow\) \textit{Relax} constraints to continuous variables.
Linear Programs

- How to solve IPs? \( \implies \) Relax constraints to continuous variables.
- Most common relaxation of IPs: Linear Program (LP).
- Constraint \( \implies \) Intersection of half-spaces \( \equiv \) Convex polytope \( \mathcal{P} \).
How to solve IPs? \( \Rightarrow \) \textit{Relax} constraints to continuous variables.

Most common relaxation of IPs: Linear Program (LP).

Constraint \( \Rightarrow \) Intersection of half-spaces \( \equiv \) Convex polytope \( \mathcal{P} \).

Examples:

\[
\begin{align*}
\text{min} & \sum_{i \in N} w_i x_i \\
\text{s.t.} & \sum_{i \in N} x_i \geq 1, \\
& x_i \in [0, 1], \; \forall i \in N
\end{align*}
\]

\[
\begin{align*}
\text{max} & \sum_{e \in E(G)} w_e x_e \\
\text{s.t.} & \sum_{e \in \delta(u)} x_e \leq 1, \; \forall u \in V(G) \\
& x_e \in [0, 1], \; \forall e \in E(G)
\end{align*}
\]
Rounding IPs

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in \{0, 1\}^n
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in [0, 1]^n
\end{align*}
\]

\[\implies x^* \text{ optimal solution of LP} \implies ???\]
Rounding IPs

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in \{0, 1\}^n
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in [0, 1]^n
\end{align*}
\]

\[\Rightarrow x^* \text{ optimal solution of LP} \Rightarrow ???\]

- **Round** \(x^*\) to obtain solution for IP. **Non-trivial!**
Rounding IPs

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in \{0, 1\}^n
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
x & \in [0, 1]^n
\end{align*}
\]

\[\implies x^* \text{ optimal solution of LP} \implies ???\]

- *Round* \(x^*\) to obtain solution for IP. **Non-trivial!**
- **Attempt 1:** Independently set \(y_i^* = 1\) w.p. \(x_i^*\), and 0 otherwise.
Rounding IPs

\[
\begin{align*}
\text{max} \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b \\
& x \in \{0, 1\}^n
\end{align*}
\] \implies

\[
\begin{align*}
\text{max} \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b \\
& x \in [0, 1]^n
\end{align*}
\]

\implies x^* \text{ optimal solution of LP} \implies ???

- Round \( x^* \) to obtain solution for IP. **Non-trivial!**
- Attempt 1: Independently set \( y_i^* = 1 \) w.p. \( x_i^* \), and 0 otherwise. \( y^* \) may be infeasible!
Independently set $y^*_i = 1$ w.p. $x^*_i$. Let $R(x^*) = \{ i \in \mathbb{N} | y^*_i = 1 \}$. $R(x^*)$: Set of active elements.

“Drop” elements from $R$ to get $S \subseteq R$ with $S \in F$.

Contention Resolution Scheme (informally) [CVZ11]

A $c$-selectable Contention Resolution Scheme (CRS) is an algorithm which receives a point $x \in P$ as input and returns an independent set $S \in F$ which contains every $i \in \mathbb{N}$ with probability at least $c \cdot x_i$.

$c$-selectable CRS $\Rightarrow$ $c$-approximate (integer) solution to a linear $f$.

$c = \min_{i \in \mathbb{N}} \Pr \left[ i \in S \mid i \in R \right]$.
Attempt 2: Independently set $y_i^* = 1$ w.p. $x_i^*$. Let $R(x^*) = \{i \in \mathbb{N} \mid y_i^* = 1\}$.
Attempt 2: Independently set $y_i^* = 1$ w.p. $x_i^*$. Let $R(x^*) = \{ i \in N \mid y_i^* = 1 \}$.

$R$: Set of active elements.
Attempt 2: Independently set $y_i^* = 1$ w.p. $x_i^*$. Let $R(x^*) = \{ i \in N \mid y_i^* = 1 \}$.

$R$: Set of active elements.

"Drop" elements from $R$ to get $S \subseteq R$ with $S \in \mathcal{F}$. 

Contention Resolution Scheme (informally) [CVZ11]

A $c$-selectable Contention Resolution Scheme (CRS) is an algorithm which receives a point $x \in P$ as input and returns an independent set $S \in \mathcal{F}$ which contains every $i \in N$ with probability at least $c \cdot x_i$.

$c$-selectable CRS $\Rightarrow$ $c$-approximate (integer) solution to a linear $f$.

$c = \min_{i \in N} \Pr \left[ i \in S \mid i \in R \right]$. 

Prophet Inequalities and Online Combinatorial Optimization
Attempt 2: Independently set $y_i^* = 1$ w.p. $x_i^*$. Let $R(x^*) = \{i \in N \mid y_i^* = 1\}$.

$R$: Set of active elements.

“Drop” elements from $R$ to get $S \subseteq R$ with $S \in \mathcal{F}$.

Contention Resolution Scheme (informally) [CVZ11]

A $c$-selectable Contention Resolution Scheme (CRS) is an algorithm which receives a point $x \in \mathcal{P}$ as input and returns an independent set $S \in \mathcal{F}$ which contains every $i \in N$ with probability at least $c \cdot x_i$. 
Attempt 2: Independently set \( y_i^* = 1 \) w.p. \( x_i^* \). Let 
\[ R(x^*) = \{ i \in N \mid y_i^* = 1 \}. \]

\( R \): Set of active elements.

“Drop” elements from \( R \) to get \( S \subseteq R \) with \( S \in \mathcal{F} \).

Contention Resolution Scheme (informally) [CVZ11]

A \( c \)-selectable Contention Resolution Scheme (CRS) is an algorithm which receives a point \( x \in \mathcal{P} \) as input and returns an independent set \( S \in \mathcal{F} \) which contains every \( i \in N \) with probability at least \( c \cdot x_i \).

- \( c \)-selectable CRS \( \implies \) \( c \)-approximate (integer) solution to a linear \( f \).
Attempt 2: Independently set $y_i^* = 1$ w.p. $x_i^*$. Let $R(x^*) = \{i \in N \mid y_i^* = 1\}$. 

$R$: Set of active elements.

“Drop” elements from $R$ to get $S \subseteq R$ with $S \in \mathcal{F}$.

**Contention Resolution Scheme (informally) [CVZ11]**

A $c$-selectable Contention Resolution Scheme (CRS) is an algorithm which receives a point $x \in \mathcal{P}$ as input and returns an independent set $S \in \mathcal{F}$ which contains every $i \in N$ with probability at least $c \cdot x_i$.

- $c$-selectable CRS $\implies$ $c$-approximate (integer) solution to a linear $f$.
- $c = \min_{i \in N} \Pr [i \in S \mid i \in R]$.
Online CRSs

Problem: Round $x$ in specific order - adversarial, random, etc.

Solution: Online Contention Resolution Schemes (OCRSs) [FSZ16]

Example (Single item):
Let $R'$ contain every $i \in R$ independently w.p. $1/2$.

$$\sum_{i \in R'} x_i = \frac{1}{2} \sum_{i \in R} x_i \leq \frac{1}{2} = \Rightarrow R' = \emptyset \text{ w.p. } \geq \frac{1}{2}.$$
Online CRSs

- **Problem:** Round $x$ in specific order - adversarial, random, etc.
Online CRSs

- **Problem:** Round $x$ in specific order - adversarial, random, etc.
- **Solution:** Online Contention Resolution Schemes (OCRSs) [FSZ16]!
**Online CRSs**

- **Problem:** Round $x$ in specific order - adversarial, random, etc.
- **Solution:** Online Contention Resolution Schemes (OCRSs) [FSZ16]!
- **Example (Single item):** Let $R'$ contain every $i \in R$ independently w.p. $1/2$.

\[ \sum_{i \in R'} x_i = \frac{1}{2} \leq \frac{1}{2} = \Rightarrow R' = \emptyset \text{ w.p. } \geq \frac{1}{2}. \]
Online CRSs

- **Problem:** Round $x$ in specific order - adversarial, random, etc.
- **Solution:** Online Contention Resolution Schemes (OCRSs) [FSZ16]!
- **Example (Single item):** Let $R'$ contain every $i \in R$ independently w.p. $1/2$.

$$\sum_{i \in R'} x_i = \frac{1}{2} \sum_{i \in R} x_i \leq \frac{1}{2} \implies R' = \emptyset \text{ w.p. } \geq \frac{1}{2}.$$
Problem: Round $x$ in specific order - adversarial, random, etc.

Solution: Online Contention Resolution Schemes (OCRSs) [FSZ16]!

Example (Single item): Let $R'$ contain every $i \in R$ independently w.p. 1/2.

\[ \sum_{i \in R'} x_i = \frac{1}{2} \sum_{i \in R} x_i \leq \frac{1}{2} \implies R' = \emptyset \text{ w.p. } \geq \frac{1}{2}. \]

\[
\Pr [i \in S \mid i \in R] = \Pr [i \in R' \mid i \in R] \cdot \Pr [1, \ldots, i - 1 \notin R'] \\
\geq \Pr [i \in R' \mid i \in R] \cdot \Pr [R' = \emptyset] \\
= \frac{1}{4}.
\]
Let $q_i = \Pr \left[ i \in R' | i \in R \right]$ and $r_i = \Pr \left[ 1, \ldots, i-1 \right. \in R' \left. \right]$. Before, $q_i = r_i = \frac{1}{2}$, for all $i$.

Idea: Ensure $r_i \cdot q_i = \frac{1}{2}$.

Initially, $r_1 = 1 \Rightarrow q_1 = \frac{1}{2}$. Notice $r_{i+1} = r_i (1 - q_i x_i)$ $\Leftrightarrow$ $r_i - r_{i+1} = r_i q_i x_i$.

Set $q_{i+1} = \frac{1}{2} r_{i+1}$. Sum up (1) to get $r_{i+1} = r_1 - i \sum_{j=1}^{x_i} x_i^2 \geq \frac{1}{2}$.

Tight: Consider $x_1 = 1 - \varepsilon$ and $x_2 = \varepsilon$ for small $\varepsilon > 0$.

$\Pr \left[ 2 \in S | 2 \in R \right] = 1 - x_1 \Pr \left[ 1 \in S | 1 \in R \right] = \frac{1}{2} - \Pr \left[ 1 \in S | 1 \in R \right] + o(1)$.\
Optimal OCRS for Single Item

- Let $q_i = \Pr [i \in R' \mid i \in R]$ and $r_i = \Pr [1, \ldots, i - 1 \notin R']$. Before, $q_i = r_i = 1/2$, for all $i$. 

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022
Let $q_i = \Pr [i \in R' \mid i \in R]$ and $r_i = \Pr [1, \ldots, i - 1 \notin R']$. Before, $q_i = r_i = 1/2$, for all $i$.

**Idea:** Ensure $r_i \cdot q_i = \frac{1}{2}$. 
Let \( q_i = \Pr [i \in R' \mid i \in R] \) and \( r_i = \Pr [1, \ldots, i - 1 \notin R'] \). Before, \( q_i = r_i = 1/2 \), for all \( i \).

**Idea:** Ensure \( r_i \cdot q_i = \frac{1}{2} \).

Initially, \( r_1 = 1 \Rightarrow q_1 = \frac{1}{2} \). Notice

\[
  r_{i+1} = r_i (1 - q_i x_i) \iff r_i - r_{i+1} = r_i q_i x_i. \tag{1}
\]
Optimal OCRS for Single Item

- Let $q_i = \Pr [i \in R' \mid i \in R]$ and $r_i = \Pr [1, \ldots, i - 1 \notin R']$. Before, $q_i = r_i = 1/2$, for all $i$.

- **Idea:** Ensure $r_i \cdot q_i = \frac{1}{2}$.

- Initially, $r_1 = 1 \implies q_1 = \frac{1}{2}$. Notice

  $$r_{i+1} = r_i (1 - q_i x_i) \iff r_i - r_{i+1} = r_i q_i x_i. \tag{1}$$

- Set $q_{i+1} = \frac{1}{2 r_{i+1}}$. Sum up (1) to get

  $$r_{i+1} = r_1 - \sum_{j=1}^{i} \frac{x_i}{2} \geq \frac{1}{2}.$$
Let $q_i = \Pr[i \in R' \mid i \in R]$ and $r_i = \Pr[1, \ldots, i - 1 \not\in R']$. Before, $q_i = r_i = 1/2$, for all $i$.

**Idea:** Ensure $r_i \cdot q_i = \frac{1}{2}$.

Initially, $r_1 = 1 \implies q_1 = \frac{1}{2}$. Notice

$$r_{i+1} = r_i (1 - q_i x_i) \iff r_i - r_{i+1} = r_i q_i x_i. \quad (1)$$

Set $q_{i+1} = \frac{1}{2r_{i+1}}$. Sum up (1) to get

$$r_{i+1} = r_1 - \sum_{j=1}^{i} \frac{x_i}{2} \geq \frac{1}{2}.$$ 

**Tight:** Consider $x_1 = 1 - \varepsilon$ and $x_2 = \varepsilon$ for small $\varepsilon > 0$. 

$$\Pr[2 \in S \mid 2 \in R] = 1 - x_1 \Pr[1 \in S \mid 1 \in R] = 1 - \Pr[1 \in S \mid 1 \in R] + o(1).$$
Greedy OCRSs

Almighty adversary: Knows $R$ in advance + any randomness of our algorithm.

Idea: Select a priori a subfamily $F' \subseteq F$ of feasible sets based on $x_i$. Greedily select $i$ if $\min R, x_i \in F'$.

Previous OCRS not greedy - $q_i$ depended on order. Better than $1/4$? Yes! $\exists 1/e$-selectable greedy OCRS for single item and it is tight [Liv21].

Algorithm includes $\{i\} \in F'$ w.p. $1 - e^{-x_i} x_i$. Extends to partition and transversal matroids.
Greedy OCRSs

- Almighty adversary: Knows $R$ in advance + any randomness of our algorithm.

Previous OCRS not greedy - $q_i$ depended on order.

Better than $1/4$? Yes!

$\exists 1/e$-selectable greedy OCRS for single item and it is tight [Liv21].

Algorithm includes $\{i\} \in F'$ w.p. $1 - e^{-xi}$. Extends to partition and transversal matroids.
Almighty adversary: Knows $R$ in advance + any randomness of our algorithm.

Idea: Select a priori a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of feasible sets based on $x$. Greedily select $i$ if

1. $i \in R$, and
2. $S_{i-1} + i \in \mathcal{F}'$.

$\implies$ Greedy OCRS.
Almighty adversary: Knows $R$ in advance + any randomness of our algorithm.

Idea: Select a priori a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of feasible sets based on $x$. Greedily select $i$ if

1. $i \in R$, and
2. $S_{i-1} + i \in \mathcal{F}'$.

$\implies$ Greedy OCRS.

Previous OCRS not greedy - $q_i$ depended on order.
Almighty adversary: Knows $R$ in advance + any randomness of our algorithm.

Idea: Select a priori a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of feasible sets based on $x$. Greedily select $i$ if

1. $i \in R$, and
2. $S_{i-1} + i \in \mathcal{F}'$.

$\implies$ Greedy OCRS.

Previous OCRS not greedy - $q_i$ depended on order.

Better than $1/4$? Yes! $\exists$ $1/e$-selectable greedy OCRS for single item and it is tight [Liv21].
Greedy OCRSs

- **Almighty adversary:** Knows $R$ in advance + any randomness of our algorithm.
- **Idea:** Select a priori a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of feasible sets based on $x$. Greedily select $i$ if
  1. $i \in R$, and
  2. $S_{i-1} + i \in \mathcal{F}'$.

  $\implies$ *Greedy OCRS.*

- Previous OCRS not greedy - $q_i$ depended on order.
- Better than $1/4$? Yes! $\exists$ $1/e$-selectable greedy OCRS for single item and it is tight [Liv21].
- Algorithm includes $\{i\} \in \mathcal{F}'$ w.p. $\frac{1-e^{-x_i}}{x_i}$. Extends to partition and transversal matroids.
Overview

1. Prophets and Secretaries
   - The Secretary Problem
   - The Prophet Inequality Problem
   - Selecting Multiple Values

2. Online Combinatorial Optimization
   - Primer on Mathematical Programming
   - Online Contention Resolution Schemes

3. Equivalence via LP Duality

4. Variations and Open Problems
From OCRS to Prophet Inequality

Assume a \( c \)-selectable OCRS \( \pi \) for some \( F \). Let

\[
x_i = \Pr \left[ i \in \arg \max_{I \in F} \sum_{j \in I} X_j \right],
\]

and

\[
v_i(x_i) = \mathbb{E} \left[ X_i | X_i's \text{ value is in its top } x_i \text{ quantile} \right] \text{ (ex-ante PI)}.
\]

Then, \( x \in P \), and

\[
\text{OPT} = \mathbb{E} \left[ \max_{I \in F} \sum_{j \in I} X_j \right] \leq \sum_{i \in N} x_i v_i(x_i).
\]

Algorithm:
- Run \( \pi \) on \( x \) to get \( S \).
- Accept \( X_i \) iff \( i \in S \).

\( \pi \) is \( c \)-selectable \( \Rightarrow \mathbb{E} [\text{ALG}] \geq c \sum_{i \in N} x_i v_i(x_i) \geq c \cdot \text{OPT} \).

Essentially a reduction to Bernoulli r.v.'s.
Assume a $c$-selectable OCRS $\pi$ for some $\mathcal{F}$. Let

$$x_i = \Pr \left[ i \in \arg \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right],$$

and

$$v_i(x_i) = \mathbb{E} \left[ X_i \mid X_i \text{’s value is in its top } x_i \text{ quantile} \right] \quad \text{(ex-ante PI)}.$$

Then, $x \in \mathcal{P}$, and $OPT = \mathbb{E} \left[ \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right] \leq \sum_{i \in \mathcal{N}} x_i v_i(x_i)$. 
Assume a \( c \)-selectable OCRS \( \pi \) for some \( \mathcal{F} \). Let 
\[
x_i = \Pr \left[ i \in \arg \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right],
\]
and 
\[
v_i(x_i) = \mathbb{E} \left[ X_i \mid X_i \text{'s value is in its top } x_i \text{ quantile} \right] \quad \text{(ex-ante PI)}.
\]

Then, \( \mathbf{x} \in \mathcal{P} \), and 
\[
OPT = \mathbb{E} \left[ \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right] \leq \sum_{i \in \mathcal{N}} x_i v_i(x_i).
\]

**Algorithm:** Run \( \pi \) on \( \mathbf{x} \) to get \( S \). Accept \( X_i \) iff \( i \in S \).
Assume a $c$-selectable OCRS $\pi$ for some $F$. Let $x_i = \Pr \left[ i \in \arg \max_{I \in F} \sum_{j \in I} X_j \right]$, and

$$v_i(x_i) = \mathbb{E} [X_i \mid X_i \text{’s value is in its top } x_i \text{ quantile}] \quad (\text{ex-ante PI}).$$

Then, $x \in P$, and $OPT = \mathbb{E} \left[ \max_{I \in F} \sum_{j \in I} X_j \right] \leq \sum_{i \in N} x_i v_i(x_i)$.

**Algorithm:** Run $\pi$ on $x$ to get $S$. Accept $X_i$ iff $i \in S$.

$\pi$ is $c$-selectable $\implies \mathbb{E} [\text{ALG}] \geq c \sum_{i \in N} x_i v_i(x_i) \geq c \cdot OPT$. 
Assume a $c$-selectable OCRS $\pi$ for some $\mathcal{F}$. Let $x_i = \Pr \left[ i \in \arg \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right]$, and

$$v_i(x_i) = \mathbb{E} [X_i \mid X_i \text{’s value is in its top } x_i \text{ quantile}] \quad \text{(ex-ante PI)}.$$ 

Then, $x \in \mathcal{P}$, and $\text{OPT} = \mathbb{E} \left[ \max_{I \in \mathcal{F}} \sum_{j \in I} X_j \right] \leq \sum_{i \in N} x_i v_i(x_i)$.

Algorithm: Run $\pi$ on $x$ to get $S$. Accept $X_i$ iff $i \in S$.

$\pi$ is $c$-selectable $\implies \mathbb{E} [\text{ALG}] \geq c \sum_{i \in N} x_i v_i(x_i) \geq c \cdot \text{OPT}$.

Essentially a reduction to Bernoulli r.v.’s.
Let $\Phi$ be the set of all deterministic online algorithms.

$$\phi : 2^N \times 2^N \times N \rightarrow \{0, 1\} \in \Phi \iff \phi(A, B, i) = 1$$

only for $B \subseteq A, i \not\in A$ and $B + i \in F$.

$A$: Set of elements seen before $i$.

$B$: Set of elements selected before $i$.

$$\phi(A, B, i) = 1 \Rightarrow \text{algorithm selects } i.$$
Let $\Phi$ be the set of all \textit{deterministic online algorithms}.

$$\phi : 2^N \times 2^N \times N \to \{0, 1\} \in \Phi \iff \phi(A, B, i) = 1$$

only for $B \subseteq A, i \notin A$ and $B + i \in \mathcal{F}$. 
Let $\Phi$ be the set of all \textit{deterministic online algorithms}.

$$\phi : 2^N \times 2^N \times N \rightarrow \{0, 1\} \in \Phi \iff \phi(A, B, i) = 1$$

only for $B \subseteq A$, $i \notin A$ and $B + i \in \mathcal{F}$.

- $A$: Set of elements seen before $i$.
- $B$: Set of elements selected before $i$.
- $\phi(A, B, i) = 1 \implies$ algorithm selects $i$. 

$q_i, \phi = \Pr[i \in S | ALG = \phi]$. 

$$\max_{\lambda, c, \mu} \sum_{\phi \in \Phi} q_i, \phi \lambda \phi \geq c \cdot x_i \forall i \in N \\
\sum_{i \in N} q_i, \phi y_i \leq \mu \forall \phi \in \Phi \\
\sum_{\phi \in \Phi} \lambda \phi = 1 \\
\sum_{i \in N} x_i y_i = 1 \\
\lambda \phi \geq 0 \forall \phi \in \Phi \\
y_i \geq 0 \forall i \in N$$
Let \( \Phi \) be the set of all \textit{deterministic online algorithms}.

\[ \phi : 2^N \times 2^N \times N \rightarrow \{0, 1\} \in \Phi \iff \phi(A, B, i) = 1 \]

only for \( B \subseteq A, i \notin A \) and \( B + i \in \mathcal{F} \).

- \( A \): Set of elements seen before \( i \).
- \( B \): Set of elements selected before \( i \).
- \( \phi(A, B, i) = 1 \implies \) algorithm selects \( i \).

Let \( q_{i,\phi} = Pr[i \in S \mid \text{ALG} = \phi] \).
Let $\Phi$ be the set of all \textit{deterministic online algorithms}.

$$
\phi : 2^N \times 2^N \times N \rightarrow \{0, 1\} \in \Phi \iff \phi(A, B, i) = 1
$$

only for $B \subseteq A$, $i \notin A$ and $B + i \in \mathcal{F}$.

- $A$: Set of elements seen before $i$.
- $B$: Set of elements selected before $i$.
- $\phi(A, B, i) = 1 \implies$ algorithm selects $i$.

Let $q_{i,\phi} = \Pr [i \in S \mid \text{ALG} = \phi]$.

\[
\begin{align*}
\max_{\lambda, c} & \quad c \\
\text{s.t.} & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \quad \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \quad \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi
\end{align*}
\]

\[
\begin{align*}
\min_{y, \mu} & \quad \mu \\
\text{s.t.} & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \quad \sum_{i \in N} x_i y_i = 1 \\
& \quad y_i \geq 0 \quad \forall i \in N
\end{align*}
\]
From (ex-ante) Prophet Inequality to OCRS (2/3)

<table>
<thead>
<tr>
<th>max $\lambda, c$</th>
<th>$\min y, \mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>s.t. $\sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N$</td>
<td>s.t. $\sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi$</td>
</tr>
<tr>
<td>$\sum_{\phi \in \Phi} \lambda_{\phi} = 1$</td>
<td>$\sum_{i \in N} x_i y_i = 1$</td>
</tr>
<tr>
<td>$\lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi$</td>
<td>$y_i \geq 0 \quad \forall i \in N$</td>
</tr>
</tbody>
</table>

If Primal has value $\geq c = \Rightarrow \exists c$-selectable OCRS. By strong LP duality, suffices to show Dual has value $\geq c$. Show that $\forall y \geq 0$ s.t. $\sum_{i \in N} x_i y_i = 1$, $\exists \phi \in \Phi$ s.t. $\sum_{i \in N} q_{i,\phi} y_i \geq \mu$. 

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022 26 / 38
From (ex-ante) Prophet Inequality to OCRS (2/3)

\[
\begin{align*}
\max_{\lambda,c} & \quad c \\
\text{s.t.} & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \quad \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \quad \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi
\end{align*}
\]

\[
\begin{align*}
\min_{y,\mu} & \quad \mu \\
\text{s.t.} & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \quad \sum_{i \in N} x_i y_i = 1 \\
& \quad y_i \geq 0 \quad \forall i \in N
\end{align*}
\]

- If PRIMAL has value \( \geq c \) \( \implies \) \( \exists \) c–selectable OCRS.
From (ex-ante) Prophet Inequality to OCRS (2/3)

\[
\begin{align*}
\max_{\lambda, c} & \quad c \\
s.t. & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \quad \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \quad \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi \\
\end{align*}
\]

\[
\begin{align*}
\min_{y, \mu} & \quad \mu \\
s.t. & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \quad \sum_{i \in N} x_i y_i = 1 \\
& \quad y_i \geq 0 \quad \forall i \in N \\
\end{align*}
\]

- If **Primal** has value \( \geq c \) \( \iff \exists \) \( c \)-selectable OCRS.
- By strong LP duality, suffices to show **Dual** has value \( \geq c \). Show that \( \forall y \geq 0 \) s.t. \( \sum_{i \in N} x_i y_i = 1 \),

  \[
  \exists \phi \in \Phi \text{ s.t. } \sum_{i \in N} q_{i,\phi} y_i \geq c.
  \]
\[
\begin{align*}
\max_{\lambda, c} & \quad c \\
\text{s.t.} \quad & \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi
\end{align*}
\]
\[
\begin{align*}
\min_{y, \mu} & \quad \mu \\
\text{s.t.} \quad & \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \sum_{i \in N} x_i y_i = 1 \\
& y_i \geq 0 \quad \forall i \in N
\end{align*}
\]
\[
\begin{align*}
\max_{\lambda, c} & \quad c \\
\text{s.t.} & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \quad \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \quad \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi \\
\end{align*}
\]

\[
\begin{align*}
\min_{y, \mu} & \quad \mu \\
\text{s.t.} & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \quad \sum_{i \in N} x_i y_i = 1 \\
& \quad y_i \geq 0 \quad \forall i \in N \\
\end{align*}
\]

- Consider Bernoulli PI instance where \( X_i = y_i \) w.p. \( x_i \) and 0 otherwise.
From (ex-ante) Prophet Inequality to OCRS (3/3)

\[
\begin{align*}
\max_{\lambda, c} & \quad c \\
\text{s.t.} & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
& \quad \sum_{\phi \in \Phi} \lambda_{\phi} = 1 \\
& \quad \lambda_{\phi} \geq 0 \quad \forall \phi \in \Phi
\end{align*}
\]

\[
\begin{align*}
\min_{y, \mu} & \quad \mu \\
\text{s.t.} & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
& \quad \sum_{i \in N} x_i y_i = 1 \\
& \quad y_i \geq 0 \quad \forall i \in N
\end{align*}
\]

- Consider Bernoulli PI instance where \( X_i = y_i \) w.p. \( x_i \) and 0 otherwise.
- \( x \in \mathcal{P} \) and \( \sum_{i \in N} x_i y_i = 1 \implies \) Value of ex-ante PI is 1.
Consider Bernoulli PI instance where $X_i = y_i$ w.p. $x_i$ and 0 otherwise.

- $x \in P$ and $\sum_{i \in N} x_i y_i = 1 \implies$ Value of ex-ante PI is 1.
- Assuming $c$-competitive (ex-ante) PI $\implies \exists \phi \in \Phi$ s.t. $E[\phi] \geq c$. 
Consider Bernoulli PI instance where $X_i = y_i$ w.p. $x_i$ and 0 otherwise.

**x** ∈ $\mathcal{P}$ and $\sum_{i \in N} x_i y_i = 1 \implies$ Value of ex-ante PI is 1.

Assuming $c$-competitive (ex-ante) PI $\implies \exists \phi \in \Phi$ s.t. $\mathbb{E} [\phi] \geq c$.

But, $\mathbb{E} [\phi] = \sum_{i \in N} q_{i,\phi} y_i$, by linearity of expectation.

Thus, $\sum_{i \in N} q_{i,\phi} y_i \geq c$. 

---

**From (ex-ante) Prophet Inequality to OCRS (3/3)**

\[
\begin{align*}
\text{max} & \quad \lambda, c \\
\text{s.t.} & \quad \sum_{\phi \in \Phi} q_{i,\phi} \lambda_{\phi} \geq c \cdot x_i \quad \forall i \in N \\
\sum_{\phi \in \Phi} \lambda_{\phi} &= 1 \\
\lambda_{\phi} &\geq 0 \quad \forall \phi \in \Phi
\end{align*}
\]

\[
\begin{align*}
\text{min} & \quad y, \mu \\
\text{s.t.} & \quad \sum_{i \in N} q_{i,\phi} y_i \leq \mu \quad \forall \phi \in \Phi \\
\sum_{i \in N} x_i y_i &= 1 \\
y_i &\geq 0 \quad \forall i \in N
\end{align*}
\]
Overview

1. Prophets and Secretaries
   - The Secretary Problem
   - The Prophet Inequality Problem
   - Selecting Multiple Values

2. Online Combinatorial Optimization
   - Primer on Mathematical Programming
   - Online Contention Resolution Schemes

3. Equivalence via LP Duality

4. Variations and Open Problems
Prophet Secretary and I.I.D. Setting

Problem:

∃ \frac{1}{2} - 1 - e^{−1} \text{ competitive algorithm}\ [Esf+15].

∃ \frac{1}{2} - 1 - e^{−1} + d \text{ for small } d > 0\ [ACK17; CSZ18] \text{ but doesn't yield OCRS.}

I.I.D. Prophet Inequality Problem:

∃ \approx 0.7451 \text{ competitive ratio algorithm and it's tight}\ [Cor+17].
Prophet Secretary problem: Prophet Inequality problem + random order. $\exists 1 - \frac{1}{e}$-competitive algorithm [Esf+15].
Prophet Secretary problem: Prophet Inequality problem + random order. \( \exists 1 - \frac{1}{e} \)-competitive algorithm [Esf+15]. \( \exists 1 - \frac{1}{e} + d \) for small \( d > 0 \) [ACK17; CSZ18] but doesn’t yield OCRS.
Prophet Secretary problem: Prophet Inequality problem + random order. $\exists 1 - \frac{1}{e}$-competitive algorithm [Esf+15]. $\exists 1 - \frac{1}{e} + d$ for small $d > 0$ [ACK17; CSZ18] but doesn’t yield OCRS.

I.I.D. Prophet Inequality problem: $\exists \approx 0.7451$-competitive ratio algorithm and it’s tight [Cor+17].
Interesting Open Problems

1. $k$-selectable greedy OCRS for matroids.
2. Prophet for i.i.d. $X_i$'s - better than $1 - \Theta(\sqrt[4]{k})$?
3. Optimal OCRSs for matching constraints.

and more...

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022

30 / 38
1. 1/e-selectable greedy OCRS for matroids.
Interesting Open Problems

1. $1/e$-selectable greedy OCRS for matroids.

2. $k$-Prophet for i.i.d. $X_i$’s - better than $1 - O\left(\frac{1}{\sqrt{k}}\right)$?
Interesting Open Problems

1. $1/e$-selectable greedy OCRS for matroids.
2. $k$-Prophet for i.i.d. $X_i$’s - better than $1 - O\left(\frac{1}{\sqrt{k}}\right)$?
3. Optimal OCRSs for matching constraints.

and more...
QUESTIONS?
One can give a simple fixed-threshold algorithm for this setting, which achieves a \(1 - O\left(\sqrt{\log k / k}\right)\) competitive ratio.

Idea: Select a threshold \(T\) such that the expected number of values \(\geq T\) are \(k - \delta\) for some \(\delta\).

Since the realizations of the \(X_i\)'s are independent, for an appropriately chosen \(\delta\), one can show that the number of realizations that are at least \(T\) are between \(k - 2\delta\) and \(k\), with high probability (Hoeffding bound).

For fixed realizations, let \(S_T = \{i \in [n] | X_i \geq T\}\). Then:

\[
\sum_{i \in S_T} X_i = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T)
\]

Since \(|S_T| \geq k - 2\delta\), our revenue is at least \((k - 2\delta)T\).
One can give a simple fixed-threshold algorithm for this setting, which achieves a $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

**Idea:** Select a threshold $T$ such that the expected number of values $\geq T$ are $k - \delta$ for some $\delta$.

Since the realizations of the $X_i$’s are independent, for an appropriately chosen $\delta$, one can show that the number of realizations that are at least $T$ are between $k - 2\delta$ and $k$, with high probability (Hoeffding bound).

For fixed realizations, let $S_T = \{i \in [n] \mid X_i \geq T\}$. Then

$$\sum_{i \in S_T} X_i = \sum_{i \in S_T} T + (X_i - T) = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T).$$
One can give a simple fixed-threshold algorithm for this setting, which achieves a $1 - O\left(\sqrt{\frac{\log k}{k}}\right)$-competitive ratio.

**Idea:** Select a threshold $T$ such that the expected number of values $\geq T$ are $k - \delta$ for some $\delta$.

Since the realizations of the $X_i$’s are independent, for an appropriately chosen $\delta$, one can show that the number of realizations that are at least $T$ are between $k - 2\delta$ and $k$, with high probability (Hoeffding bound).

For fixed realizations, let $S_T = \{i \in [n] | X_i \geq T\}$. Then

$$\sum_{i \in S_T} X_i = \sum_{i \in S_T} T + (X_i - T) = T \cdot |S_T| + \sum_{i \in S_T} (X_i - T).$$

Since $|S_T| \geq k - 2\delta$, our revenue is at least $(k - 2\delta)T$. 

Prophet Inequalities and Online Combinatorial Optimization

April 21th, 2022
Let $S^*$ be the optimal set selected by the prophet. Then

$$\text{OPT} = \sum_{i \in S^*} X_i \leq \sum_{i \in S^*} T_i + (X_i - T_i) \leq kT + n \sum_{i=1}^{T} (X_i - T_i),$$

Since $|S_T| \leq k$, we accepted every value that was at least $T$. Thus

$$\sum_{i \in S_T} (X_i - T_i) = n \sum_{i=1}^{T} (X_i - T_i) \geq \text{OPT} - kT \geq \delta k \text{OPT} - 2 \delta k T.$$ 

For $\delta = \sqrt{2k \log k}$, we get

$$\sum_{i \in S_T} X_i \geq \left(1 - 2\delta k\right) \text{OPT} = \left(1 - \sqrt{8 \log k} / k\right) \text{OPT}.$$
Let $S^*$ be the optimal set selected by the prophet. Then

$$OPT = \sum_{i \in S^*} X_i \leq \sum_{i \in S^*} T + (X_i - T) \leq kT + \sum_{i=1}^{n} (X_i - T),$$
Let $S^*$ be the optimal set selected by the prophet. Then

$$OPT = \sum_{i \in S^*} X_i \leq \sum_{i \in S^*} T + (X_i - T) \leq kT + \sum_{i=1}^{n} (X_i - T),$$

Since $|S_T| \leq k$, we accepted every value that was at least $T$. Thus

$$\sum_{i \in S_T} (X_i - T) = \sum_{i=1}^{n} (X_i - T) \geq OPT - kT \geq \frac{k - 2\delta}{k} (OPT - kT)$$

$$= \left(1 - \frac{2\delta}{k}\right) OPT - (k - 2\delta) T.$$
Fixed-Threshold Algorithm for $k$-Prophet (Proof 2/2)

- Let $S^*$ be the optimal set selected by the prophet. Then

$$OPT = \sum_{i \in S^*} X_i \leq \sum_{i \in S^*} T + (X_i - T) \leq kT + \sum_{i=1}^{n} (X_i - T),$$

- Since $|S_T| \leq k$, we accepted every value that was at least $T$. Thus

$$\sum_{i \in S_T} (X_i - T) = \sum_{i=1}^{n} (X_i - T) \geq OPT - kT \geq \frac{k - 2\delta}{k} (OPT - kT)$$

$$= \left(1 - \frac{2\delta}{k}\right) OPT - (k - 2\delta)T.$$  

- For $\delta = \sqrt{2k \log k}$, we get

$$\sum_{i \in S_T} X_i \geq \left(1 - \frac{2\delta}{k}\right) OPT = \left(1 - \sqrt{\frac{8 \log k}{k}}\right) OPT.$$
<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Conference/Journal</th>
<th>Year</th>
<th>Pages/Publication Details</th>
</tr>
</thead>
</table>


