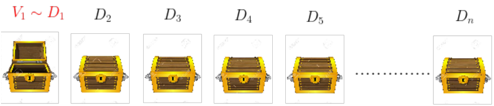
Prophet Inequalities with Limited Information - The Single Choice Problem and Beyond

Eshwar Ram Arunachaleswaran
WPE II Presentation
Dept. of Computing and Information Sciences
University of Pennsylvania

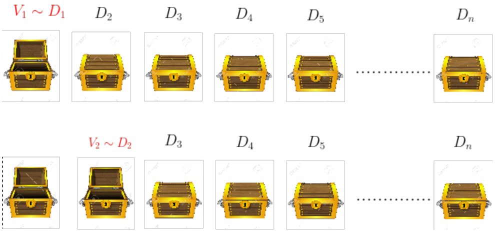
June 22, 2021

Outline of Talk

- Gambler sees a sequence of n non-negative values $V_1, V_2 \cdots V_n$
- Each value V_i is drawn independently from a distribution D_i



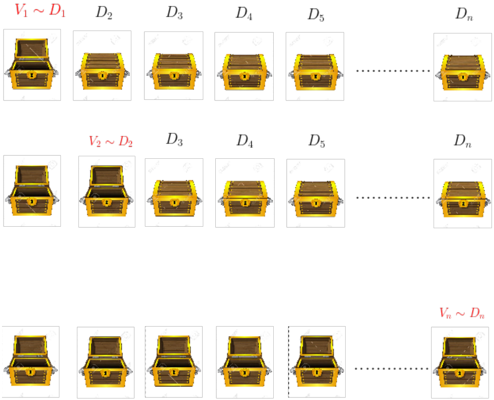
- Gambler sees a sequence of n non-negative values $V_1, V_2 \cdots V_n$
- Each value V_i is drawn independently from a distribution D_i
- Must accept or reject a value *irrevocably* on seeing it



- Objective of gambler is to maximize expected reward

- Objective of gambler is to maximize expected reward
- Can find an optimal strategy using backward induction, working backward from the last round

- Objective of gambler is to maximize expected reward
- Can find an optimal strategy using backward induction, working backward from the last round



- Objective of gambler is to maximize expected reward
- Can find an optimal strategy using backward induction, working backward from the last round
- Krengel et al.(1978) - Strategy that guarantees $1/2$ of the expected optimum reward

Introduction - Prophet Inequalities

- Objective of gambler is to maximize expected reward
- Can find an optimal strategy using backward induction, working backward from the last round
- Krengel et al.(1978) - Strategy that guarantees 1/2 of the expected optimum reward
- Samuel-Cahn (1984) - Same guarantee, but using a simple threshold strategy

Introduction - Prophet Inequalities

- Objective of gambler is to maximize expected reward
- Can find an optimal strategy using backward induction, working backward from the last round
- Krengel et al. (1978) - Strategy that guarantees $1/2$ of the expected optimum reward
- Samuel-Cahn (1984) - Same guarantee, but using a simple threshold strategy
- **All these strategies require non trivial knowledge of the distributions**

- Can the gambler achieve similar guarantees for the competitive ratio without knowing everything about the distribution?

- Can the gambler achieve similar guarantees for the competitive ratio without knowing everything about the distribution?
- In particular, is it sufficient to have a few samples, maybe even just one sample, from each distribution?

- Can the gambler achieve similar guarantees for the competitive ratio without knowing everything about the distribution?
- In particular, is it sufficient to have a few samples, maybe even just one sample, from each distribution?
- **Short Answer : Yes**

References I

Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg.
Prophet inequalities with limited information.
In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1358–1377. SIAM, 2014.

José Correa, Paul Dütting, Felix Fischer, and Kevin Schewior.
Prophet inequalities for iid random variables from an unknown distribution.
In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 3–17, 2019.

Aviad Rubinstein, Jack Z Wang, and S Matthew Weinberg.
Optimal single-choice prophet inequalities from samples.
Innovations in Theoretical Computer Science, 2020.

Some Basics First

Prophet Inequalities Problem Statement

- Given an environment $\mathcal{I} = \{[n], \mathcal{J}\}$ (for example $\mathcal{J}$ = matchings in a given graph)
- Observe sequence $V_1, V_2 \cdots V_n$ where $V_i \sim D_i$
- Accept or reject elements irrevocably, maintain an accepted set $A \in \mathcal{J}$.

Prophet Inequalities Generalized Problem Statement

- Given an environment $\mathcal{I} = \{[n], \mathcal{J}\}$ (for example $\mathcal{J}$ = matchings in a given graph)
- Observe sequence $V_1, V_2 \cdots V_n$ where $V_i \sim D_i$
- Accept or reject elements irrevocably, maintain an accepted set $A \in \mathcal{J}$.
- Competitive ratio α if expected reward is $\alpha.\mathbb{E}_{V \sim D}[\text{OPT}(V)]$

(comparison to the reward picked by an offline player - called the prophet)

Prophet Inequalities Generalized Problem Statement

- Given an environment $\mathcal{I} = \{[n], \mathcal{J}\}$ (for example $\mathcal{J}$ = matchings in a given graph)
- Observe sequence $V_1, V_2 \cdots V_n$ where $V_i \sim D_i$
- Accept or reject elements irrevocably, maintain an accepted set $A \in \mathcal{J}$.
- Competitive ratio α if expected reward is $\alpha.\mathbb{E}_{V\sim D}[\text{OPT}(V)]$ (comparison to the reward picked by an offline player - called the prophet)
- Gambler allowed to use randomized algorithms

Prophet Inequalities Generalized Problem Statement

- Given an environment $\mathcal{I} = \{[n], \mathcal{J}\}$ (for example $\mathcal{J}$ = matchings in a given graph)
- Observe sequence $V_1, V_2 \cdots V_n$ where $V_i \sim D_i$
- Accept or reject elements irrevocably, maintain an accepted set $A \in \mathcal{J}$.
- Competitive ratio α if expected reward is $\alpha.\mathbb{E}_{V \sim D}[\mathrm{OPT}(V)]$ (comparison to the reward picked by an offline player - called the prophet)
- Prophet inequalities for various settings - matchings, matroids, general set systems - in the full information setting

Prophet Inequalities Generalized Problem Statement

- Given an environment $\mathcal{I} = \{[n], \mathcal{J}\}$ (for example $\mathcal{J}$ = matchings in a given graph)
- Observe sequence $V_1, V_2 \cdots V_n$ where $V_i \sim D_i$
- Accept or reject elements irrevocably, maintain an accepted set $A \in \mathcal{J}$.
- Competitive ratio α if expected reward is $\alpha.\mathbb{E}_{V \sim D}[\mathrm{OPT}(V)]$ (comparison to the reward picked by an offline player - called the prophet)
- Prophet inequalities for various settings - matchings, matroids, general set systems - in the full information setting
- Many results have been extended to the limited information setting

Motivation - Online Mechanism Design Problems

- Suppose we wish to sell a single good to a pool of buyers arriving in some sequence

Motivation -Online Mechanism Design

- Suppose we wish to sell a single good to a pool of buyers arriving in some sequence
- Each buyer has a private value V_i, drawn from some distribution D_i
- Must decide in an online manner whether to sell to the newly arrived buyer

Motivation -Online Mechanism Design

- Suppose we wish to sell a single good to a pool of buyers arriving in some sequence
- Each buyer has a private value V_i, drawn from some distribution D_i
- Must decide in an online manner whether to sell to the newly arrived buyer
- Natural connection between prophet inequalities and optimal mechanism design

An Application -Online Mechanism Design

- Suppose we wish to sell a single good to a pool of buyers arriving in some sequence
- Each buyer has a private value V_i, drawn from some distribution D_i
- Must decide in an online manner whether to sell to the newly arrived buyer
- Natural connection between prophet inequalities and optimal mechanism design
- Results translate in both directions

Motivation -Online Mechanism Design

- Suppose we wish to sell a single good to a pool of buyers arriving in some sequence
- Each buyer has a private value V_i, drawn from some distribution D_i
- Must decide in an online manner whether to sell to the newly arrived buyer
- Natural connection between prophet inequalities and optimal mechanism design
- Results translate in both directions
- Combinatorial allocation problems motivated the generalized prophet inequalities problem

Outline of Talk

Outline of Talk

The Single Choice Problem

- Given samples $\{S_1, S_2 \cdots S_n\}$, where S_i is drawn independently from D_i, how well can the gambler do ?

Single Choice Problem

- Given samples $\{S_1, S_2 \cdots S_n\}$, where S_i is drawn independently from D_i, how well can the gambler do ?
- Gambler cannot do better than competitive ratio $1/2$, even with full knowledge of the distributions

Single Choice Problem

- Given samples $\{S_1, S_2 \cdots S_n\}$, where S_i is drawn independently from D_i, how well can the gambler do ?
- Gambler cannot do better than competitive ratio $1/2$, even with full knowledge of the distributions

Theorem (Rubinstein et al [RWW20])

There is a $1/2$*-competitive threshold based algorithm for the single sample single choice prophet inequality problem.*

Outline of Talk

- Consider the following two distributions:
- $$V_1 = 1 \text{ w.p. } 1$$
- $$V_2 = \begin{cases} \frac{1}{\varepsilon} \text{ w.p. } \varepsilon \\ 0 \text{ w.p. } 1 - \varepsilon \end{cases}$$

Upper Bounds - Single Choice Problem

- Consider the following two distributions:
- $$V_1 = 1 \text{ w.p. } 1$$
- $$V_2 = \begin{cases} \frac{1}{\varepsilon} \text{ w.p. } \varepsilon \\ 0 \text{ w.p. } 1 - \varepsilon \end{cases}$$
- Expected reward of the prophet (optimal reward) is $2 - \varepsilon$

Upper Bounds - Single Choice Problem

- Consider the following two distributions:
- $$V_1 = 1 \text{ w.p. } 1$$
- $$V_2 = \begin{cases} \frac{1}{\varepsilon} \text{ w.p. } \varepsilon \\ 0 \text{ w.p. } 1-\varepsilon \end{cases}$$
- Two possible strategies for the gambler - based on whether or not to accept the first value

- Consider the following two distributions:
- $$V_1 = 1 \text{ w.p. } 1$$
- $$V_2 = \begin{cases} \frac{1}{\varepsilon} \text{ w.p. } \varepsilon \\ 0 \text{ w.p. } 1-\varepsilon \end{cases}$$
- Two possible strategies for the gambler - based on whether or not to accept the first value
- Both strategies have expected reward 1

Outline of Talk

1/2-Competitive Strategy

Theorem (Rubinstein et al [RWW20])

There is a 1/2-competitive threshold based algorithm for the single sample single choice prophet inequality problem.

- **Algorithm:** Set threshold $\tau = \max_i S_i$, accept any value that is at least τ.

1/2-Competitive Strategy : Proof

1/2-Competitive Strategy : Proof

- **Key Observation 1:** The set $\{V_i, S_i\}$ is a set of two independent draws $\{Y_i, Z_i\}$ from the distribution D_i. WLOG, let $Y_i > Z_i$.

- **Key Observation 1:** The set $\{V_i, S_i\}$ is a set of two independent draws $\{Y_i, Z_i\}$ from the distribution D_i. WLOG, let $Y_i > Z_i$.
- **Key Observation 2:** Based on an independent, unbiased coin toss, either $V_i = Y_i$, $S_i = Z_i$ or vice-versa

1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i\in[n]}$
- We will show a competitive ratio of $1/2$ for this fixed draw

1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots > X_{2n}$

1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots > X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct

1/2-Competitive Strategy : Proof

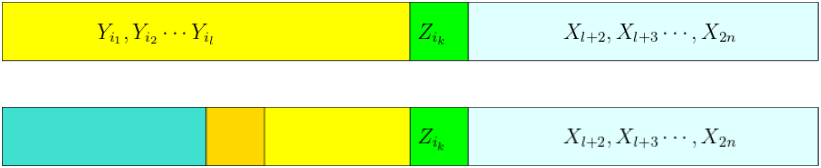
Penn
UNIVERSITY of PENNSYLVANIA

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots > X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct

$X_1, X_2 \cdots X_l$	X_{l+1}	$X_{l+2}, X_{l+3} \cdots, X_{2n}$
$Y_{i_1}, Y_{i_2} \cdots Y_{i_l}$	Z_{i_k}	$X_{l+2}, X_{l+3} \cdots, X_{2n}$

1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots > X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- Observe that the prophet picks the largest value.

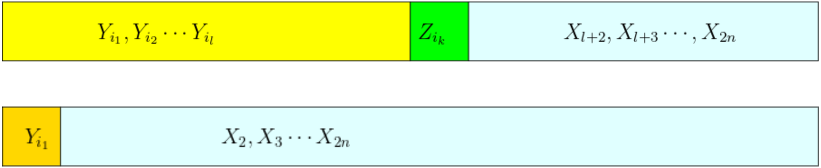


1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} >$ .... where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the prophet gets X_1 ?

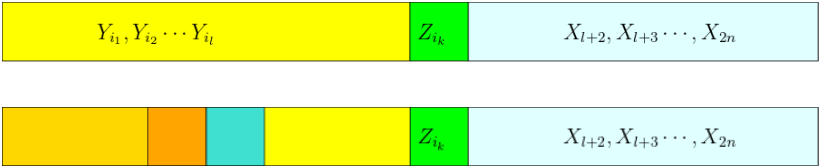
1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the prophet gets X_1 ?
- With probability 1/2 (i.e., Y_{i_1} is set as the i_1-th value)



1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- Observe that the prophet picks the largest value.
- The gambler gets at least the smallest value that is larger than the largest sample.

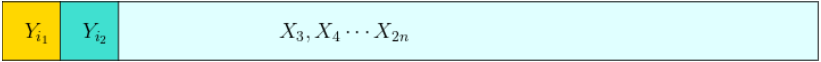


1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots$. where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the gambler gets X_1 ?

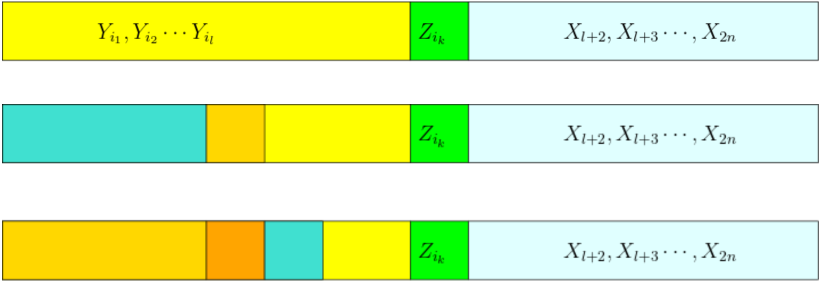
1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots$. where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the gambler gets X_1 ?
- With probability 1/4 (i.e., Y_{i_1} is set as the i_1-th value and Y_{i_2} is set as the i_2-th value)



1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots$. where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- More generally, what is the probability that the prophet (gambler) gets X_j, for $j < l$?
- With probability $1/2^j$ $(1/2^{j+1})$



1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- Note that the first sample as well as first value must appear by X_{l+1}

1/2-Competitive Strategy : Proof

- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots$. where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- $$\mathbb{E}[\text{Prophet's Reward}] = \left(\sum_{i=1}^{l} \frac{X_i}{2^i}\right) + \frac{X_{l+1}}{2^l}$$
- $$\mathbb{E}[\text{Gambler's Reward}] \geq \left(\sum_{i=1}^{l} \frac{X_i}{2^{i+1}}\right) + \frac{X_l}{2^{l+1}}$$

Summary - Single Choice Problem

- Given samples $\{S_1, S_2 \cdots S_n\}$, where S_i is drawn independently from D_i, how well can the gambler do ?
- Gambler cannot do better than competitive ratio $1/2$, even with full knowledge of the distributions
- Rubinstein et al. : Simple Threshold based strategy is 1/2-Competitive
- Simple algorithm that matches the full information competitive ratio as well as the upper bound, all with a single sample
- This algorithm is a special case of a more general algorithm by Azar et al., which achieves a competitive ratio of $1 - O\left(\frac{1}{\sqrt{k}}\right)$ for the k-choice problem.

The k Choice Problem

- Given samples $\{S_1, S_2 \cdots S_n\}$, where S_i is drawn independently from D_i, how well can the gambler do ?
- Azar et al. (2013), showed a $1 - O\left(\frac{1}{\sqrt{k}}\right)$-competitive algorithms
- Asymptotically comparable to the upper bound
- Uses the largest $k - 2\sqrt{k}$ samples to set k thresholds

The k Choice Problem

- Azar et al. (2013), showed a $1 - O\left(\frac{1}{\sqrt{k}}\right)$-competitive algorithms
- Uses the largest $k - 2\sqrt{k}$ samples to set k thresholds
- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots.$ where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the prophet (gambler) gets X_j, for $j < l$?

The k Choice Problem

- Azar et al. (2013), showed a $1 - O\left(\frac{1}{\sqrt{k}}\right)$-competitive algorithms
- Uses the largest $k - 2\sqrt{k}$ samples to set k thresholds
- Fix any draw of samples and values $\{Y_i, Z_i\}_{i \in [n]}$
- Let these quantities be $X_1 > X_2 \cdots X_{2n}$
- Equivalently, they are $Y_{i_1} > Y_{i_2} \cdots Y_{i_l} > Z_{i_k} > \ldots$. where $k \in \{1, 2, \cdots l\}$ and $\{i_1, i_2, \cdots i_l\}$ are all distinct
- What is the probability that the prophet (gambler) gets X_j, for $j < l$?
- Bounding the height of a negatively correlated random walks used to compare probabilities

Outline of Talk

Outline of Talk

The Secretary Problem

- There are n values $V_1, V_2 \cdots V_n$ which are presented in uniformly random order - $V_{i_1}, V_{i_2}, \cdots V_{i_n}$
- Once again, must choose irrevocably whether or not to accept the j-th value V_{i_j}
- Objective is to maximize the probability of selecting the maximum value

The Secretary Problem

- There are n values $V_1, V_2 \cdots V_n$ which are presented in uniformly random order - $V_{i_1}, V_{i_2}, \cdots V_{i_n}$
- Once again, must choose irrevocably whether or not to accept the j-th value V_{i_j}
- Objective is to maximize the probability of selecting the maximum value

Theorem

There is an algorithm that accepts the maximum value with probability $1/e$ for the single choice secretary problem. Additionally, the probability $1/e$ is optimal.

The Secretary Problem

- There are n values $V_1, V_2 \cdots V_n$ which are presented in uniformly random order - $V_{i_1}, V_{i_2}, \cdots V_{i_n}$
- Once again, must choose irrevocably whether or not to accept the j-th value V_{i_j}
- Objective is to maximize the probability of selecting the maximum value

Theorem

There is an algorithm that accepts the maximum value with probability $1/e$ for the single choice secretary problem. Additionally, the probability $1/e$ is optimal.

- **Algorithm :** Observe the first $1/e$ fraction of values and note down the maximum, accept the first value outside this set exceeding the noted maximum

The Secretary Problem

- There are n values $V_1, V_2 \cdots V_n$ which are presented in uniformly random order - $V_{i_1}, V_{i_2}, \cdots V_{i_n}$
- Once again, must choose irrevocably whether or not to accept the j-th value V_{i_j}
- Objective is to maximize the probability of selecting the maximum value
- Just like prophet inequalities, the problem can be generalized to selecting more than one element
- Constant (or better) probability of selecting the optimal set for multi-choice problems - eg: matchings, k-Choice, matroids etc.

The Secretary Problem

- There are n values $V_1, V_2 \cdots V_n$ which are presented in uniformly random order - $V_{i_1}, V_{i_2}, \cdots V_{i_n}$
- Once again, must choose irrevocably whether or not to accept the j-th value V_{i_j}
- Objective is to maximize the probability of selecting the maximum value
- Just like prophet inequalities, the problem can be generalized to selecting more than one element
- Constant (or better) probability of selecting the optimal set for multi-choice problems - eg: matchings, k-Choice, matroids etc.
- **Are prophets easier than secretaries**?

Outline of Talk

Reducing Prophets to Secretaries

Theorem (Azar et al. [AKW14])

Any α-competitive order-oblivious algorithm $\mathcal{A}_S$ for the secretary problem in environment $\mathcal{I} = \{[n], \mathcal{J}\}$ yields a α-competitive algorithm $\mathcal{A}_P$ for the corresponding **single sample** *prophet inequality problem in the same environment.*

Order-Oblivious Algorithms

- $\mathcal{A}_S$ picks a threshold index k before starting the sequence (potentially using random bits) and only observes the first k values $A = \{v_{i_1}, v_{i_2} \cdots v_{i_k}\}$ in the sequence.

Order-Oblivious Algorithms

- $\mathcal{A}_S$ picks a threshold index k before starting the sequence (potentially using random bits) and only observes the first k values $A = \{v_{i_1}, v_{i_2} \cdots v_{i_k}\}$ in the sequence.
- $\mathcal{A}_S$ assumes only that the set A is a uniformly random subset of size k of the set $\{v_i\}_{i \in [n]}$ of n values, while proving the competitive ratio.

Order-Oblivious Algorithms- Example

- Single choice problem
- Select $k = \text{Binomial}(n, 1/2)$ and set threshold as max of first k values, accept first value after k values that beats this threshold

Order-Oblivious Algorithms- Example

- Single choice problem
- Select $k = \text{Binomial}(n, 1/2)$ and set threshold as max of first k values, accept first value after k values that beats this threshold
- **Claim :** This algorithm picks the maximum value with probability $1/4$ (under order-oblivious analysis)

Order-Oblivious Algorithms- Example

- Single choice problem
- Select $k = \text{Binomial}(n, 1/2)$ and set threshold as max of first k values, accept first value after k values that beats this threshold
- **Claim :** This algorithm picks the maximum value with probability $1/4$ (under order-oblivious analysis)
- Consider the following construction of the first k values - each value is included independently with probability $1/2$

Order-Oblivious Algorithms- Example

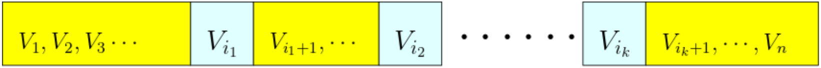
- Single choice problem
- Select $k = \text{Binomial}(n, 1/2)$ and set threshold as max of first k values, accept first value after k values that beats this threshold
- **Claim :** This algorithm picks the maximum value with probability $1/4$ (under order-oblivious analysis)
- Consider the following construction of the first k values - each value is included independently with probability $1/2$
- Thus, with probability $1/4$, the second largest element is in the first k values and the largest value is in the second part.

Reducing Prophets to Secretaries

Theorem

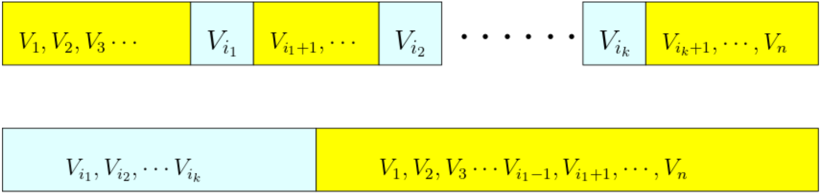
Any α*-competitive order-oblivious algorithm* $\mathcal{A}_S$ *for the secretary problem in environment* $\mathcal{I} = \{[n], \mathcal{J}\}$ *yields a* α*-competitive algorithm* $\mathcal{A}_P$ *for the corresponding* **single sample** *prophet inequality problem in the same environment.*

Proof

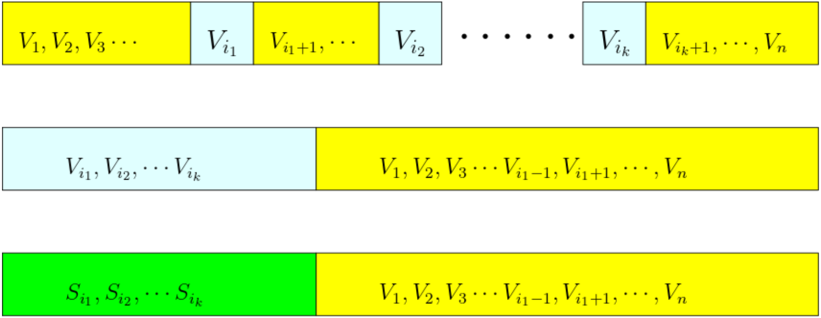


$V_1, V_2, V_3 \cdots$
V_{i_1}
$V_{i_1+1}, \cdots$
V_{i_2}
$\cdots\cdots$
V_{i_k}
$V_{i_k+1}, \cdots, V_n$

Proof


$V_1, V_2, V_3 \cdots$
V_{i_1}
$V_{i_1+1}, \cdots$
V_{i_2}
$\cdots\cdots$
V_{i_k}
$V_{i_k+1}, \cdots, V_n$
$V_{i_1}, V_{i_2}, \cdots V_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$

Proof


$V_1, V_2, V_3 \cdots$
V_{i_1}
$V_{i_1+1}, \cdots$
V_{i_2}
V_{i_k}
$V_{i_k+1}, \cdots, V_n$
$V_{i_1}, V_{i_2}, \cdots V_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$
$S_{i_1}, S_{i_2}, \cdots S_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$

Proof

- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$

- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$
- Select k in the same manner as $\mathcal{A}_S$. Select a uniformly random subset $K = \{i_1, i_2 \cdots i_k\}$ of $[n]$ of size k - the first k elements of X are the samples $S_{i_1}, S_{i_2}, \cdots S_{i_k}$.

Proof

- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$
- Select k in the same manner as $\mathcal{A}_S$. Select a uniformly random subset $K = \{i_1, i_2 \cdots i_k\}$ of $[n]$ of size k - the first k elements of X are the samples $S_{i_1}, S_{i_2}, \cdots S_{i_k}$.
- The rest of the sequence X is constructed in an online manner - observe each value V_i, if $i \in K$, ignore it.

Proof

- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$
- Select k in the same manner as $\mathcal{A}_S$. Select a uniformly random subset $K = \{i_1, i_2 \cdots i_k\}$ of $[n]$ of size k - the first k elements of X are the samples $S_{i_1}, S_{i_2}, \cdots S_{i_k}$.
- The rest of the sequence X is constructed in an online manner - observe each value V_i, if $i \in K$, ignore it.
- If $i \notin K$, add V_i as the next element of X

Proof

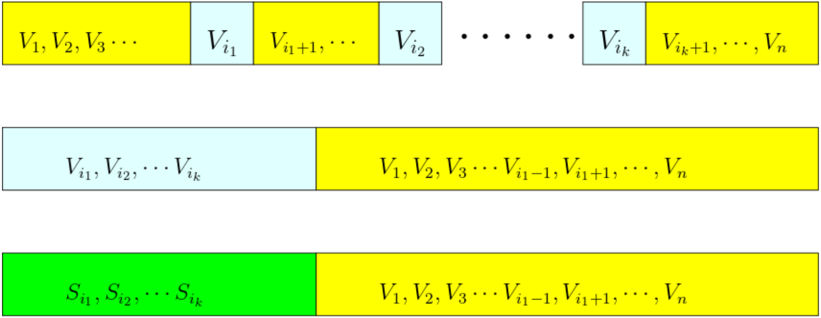
- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$
- Select k in the same manner as $\mathcal{A}_S$. Select a uniformly random subset $K = \{i_1, i_2 \cdots i_k\}$ of $[n]$ of size k - the first k elements of X are the samples $S_{i_1}, S_{i_2}, \cdots S_{i_k}$.
- The rest of the sequence X is constructed in an online manner - observe each value V_i, if $i \in K$, ignore it.
- If $i \notin K$, add V_i as the next element of X

Proof

- We construct an instance of the secretary problem - $X = X_1, X_2, \cdots X_n$
- Select k in the same manner as $\mathcal{A}_\mathcal{S}$. Select a uniformly random subset $K = \{i_1, i_2 \cdots i_k\}$ of $[n]$ of size k - the first k elements of X are the samples $S_{i_1}, S_{i_2}, \cdots S_{i_k}$.
- The rest of the sequence X is constructed in an online manner - observe each value V_i, if $i \in K$, ignore it.
- If $i \notin K$, add V_i as the next element of X
- **Run algorithm $\mathcal{A}_\mathcal{S}$ on X**

Proof

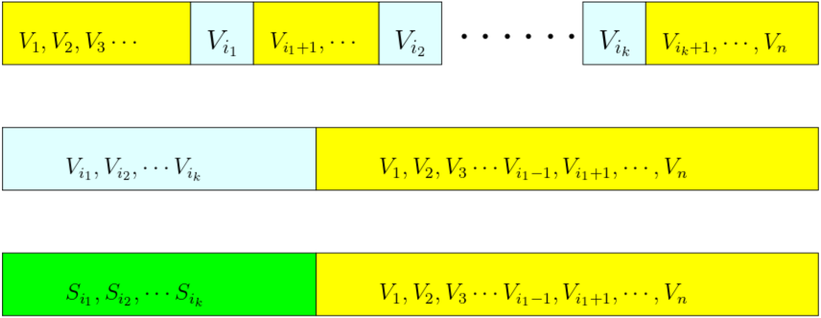
- **Observation 1:** Our algorithm picks a feasible subset of values


$V_1, V_2, V_3 \cdots$
V_{i_1}
$V_{i_1+1}, \cdots$
V_{i_2}
$\cdots\cdots$
V_{i_k}
$V_{i_k+1}, \cdots, V_n$
$V_{i_1}, V_{i_2}, \cdots V_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$
$S_{i_1}, S_{i_2}, \cdots S_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$

Proof

- **Observation 1:** Our algorithm picks a feasible subset of values
- **Observation 2:** The expected value of the maximum feasible subset of X is equal to the expected value of the maximum subset of V

Proof


$V_1, V_2, V_3 \cdots$
V_{i_1}
$V_{i_1+1}, \cdots$
V_{i_2}
$\cdots\cdots$
V_{i_k}
$V_{i_k+1}, \cdots, V_n$
$V_{i_1}, V_{i_2}, \cdots V_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$
$S_{i_1}, S_{i_2}, \cdots S_{i_k}$
$V_1, V_2, V_3 \cdots V_{i_1-1}, V_{i_1+1}, \cdots, V_n$

Proof

- **Observation 1:** Our algorithm picks a feasible subset of values
- **Observation 2:** The expected value of the maximum feasible subset of X is equal to the expected. value of the maximum subset of V
- Thus the guarantee of $\mathcal{A}_s$ translates into a prophet inequality

Corollaries

- $O(\log\log(\text{rank}))$-competitive factor algorithm for matroid constraints
- $1/8$-competitive factor algorithm for graphic matroids
- $\frac{1}{12\sqrt{3}}$-competitive factor algorithm for laminar matroids
- $1/16$-competitive factor algorithm for transversal matroids
- **Note:** All the above are single sample prophet inequality problems

Outline of Talk

IID Prophet Inequalities

- Consider the single choice problem, but with all distributions identical

IID Prophet Inequalities

- Consider the single choice problem, but with all distributions identical
- If the gambler knows this distribution, Correa et al.(2017) showed an algorithm with 0.745-competitive ratio

IID Prophet Inequalities

- Consider the single choice problem, but with all distributions identical
- If the gambler knows this distribution, Correa et al. showed an algorithm with 0.745-competitive ratio
- This result is optimal, due to an impossibility result of Hill and Kertz(1982)

IID Prophet Inequalities

- Consider the single choice problem, but with all distributions identical
- If the gambler knows this distribution, Correa et al. showed an algorithm with 0.745-competitive ratio
- This result is optimal, due to an impossibility result of Hill and Kertz
- What if the distribution is not known to the gambler?

- $1/e$- Competitive Algorithm, based on the secretary problem

Theorem

There exists a $1/e$-competitive algorithm for the unknown IID prophet problem.

Unknown IID Prophet Inequalities

- $1/e$- Competitive Algorithm, based on the secretary problem
- $1/e$ upper bound, based on the construction of a pathological distribution for any fixed algorithm

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ competitive ratio for the unknown IID prophet inequality problem.

Unknown IID Prophet Inequalities

- $1/e$- Competitive Algorithm, based on the secretary problem
- $1/e$ upper bound, based on the construction of a pathological distribution for any fixed algorithm
- Improves to $1 - 1/e \approx 0.632$ with $n - 1$ samples

Unknown IID Prophet Inequalities - Summary

- $1/e$- Competitive Algorithm, based on the secretary problem
- $1/e$ upper bound, based on the construction of a pathological distribution for any fixed algorithm
- Improves to $1 - 1/e \approx 0.632$ with $n - 1$ samples
- Improves all the way to $0.745 - \varepsilon$ with $O_\varepsilon(n)$ samples

Outline of Talk

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ *competitive ratio for the unknown IID prophet inequality problem.*

- **Key Idea :** Use the fact that $1/e$ is the optimal probability of selecting the max element in the secretary problem

Unknown IID Prophet Inequalities

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ competitive ratio for the unknown IID prophet inequality problem.

- **Key Idea :** Use the fact that $1/e$ is the optimal probability of selecting the max element in the secretary problem
- Need to restrict the class of algorithms to secretary-like algorithms

Unknown IID Prophet Inequalities

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ competitive ratio for the unknown IID prophet inequality problem.

- For a fixed algorithm $\mathcal{A}$, design distribution F such that $\mathcal{A}$ only uses ordinal information on F

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ competitive ratio for the unknown IID prophet inequality problem.

- For a fixed algorithm $\mathcal{A}$, design distribution F such that $\mathcal{A}$ only uses ordinal information on F
- Set up the support of F so that the maximum element contributes a $1 - o(1)$ fraction of the expected optimum

Value Oblivious Algorithms

Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$*, there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$*, there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$*,*

$$\Pr[\mathcal{A} \textit{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

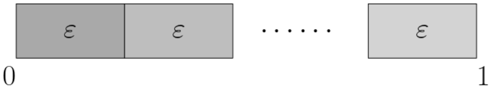
Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$, *there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$, *there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 1$:



□

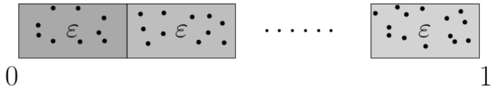
Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$, *there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$, *there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 1$:



□

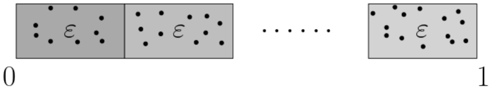
Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$, *there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$, *there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 1$:



□

At least one interval has an infinite number of points. Call these points S_1.

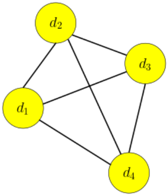
Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$, *there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$, *there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 2$: Consider the complete graph on the vertex set S_1.



Value Oblivious Algorithms

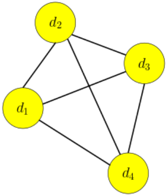
Lemma (Correa et al [CDFS19])

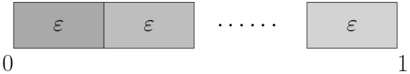
For any $\varepsilon > 0$, there exists an infinite subset $S \subset \mathbb{N}$ such that : for all $i \in [n]$, there exists $p_i \in [0,1]$ such that for distinct $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 2$: Consider the complete graph on the vertex set S_1. Color the edge (u, w) where $u < w$ with the corresponding color.





Value Oblivious Algorithms

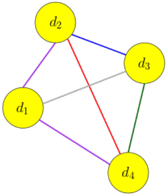
Lemma (Correa et al [CDFS19])

For any $\varepsilon > 0$, there exists an infinite subset $S \subset \mathbb{N}$ such that : for all $i \in [n]$, there exists $p_i \in [0,1]$ such that for distinct $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 2$: Consider the complete graph on the vertex set S_1. Color the edge (u, w) where $u < w$ with the corresponding color.



Value Oblivious Algorithms

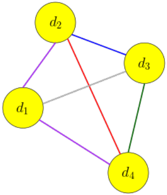
Lemma (Correa et al [CDFS19])

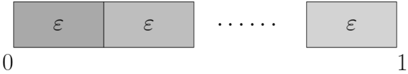
For any $\varepsilon > 0$, there exists an infinite subset $S \subset \mathbb{N}$ such that : for all $i \in [n]$, there exists $p_i \in [0,1]$ such that for distinct $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 2$: Consider the complete graph on the vertex set S_1. Color the edge (u, w) where $u < w$ with the corresponding color.
We want a monochromatic infinite clique.





Theorem (Ramsay)

Let H be a d-uniform infinite complete hypergraph whose edges coloured with c colours. Then, H must have a monochromatic d-uniform infinite complete sub-hypergraph.

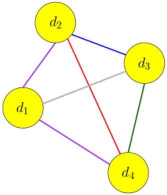
Lemma (Correa et al [CDFS19])

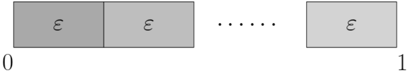
For any $\varepsilon > 0$, *there exists an infinite subset* $S \subset \mathbb{N}$ *such that : for all* $i \in [n]$, *there exists* $p_i \in [0,1]$ *such that for distinct* $v_1, v_2, \cdots v_i \in S$,

$$\Pr[\mathcal{A} \text{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For $i = 2$: Consider the complete graph on the vertex set S_1. Color the edge (u, w) where $u < w$ with the corresponding color.
Call the monochromatic infinite clique S_2





Lemma (Correa et al [CDFS19])

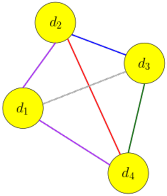
For any $\varepsilon > 0$, there exists an infinite subset $S \subset \mathbb{N}$ such that : for all $i \in [n]$, there exists $p_i \in [0,1]$ such that for distinct $v_1, v_2, \cdots v_i \in S$,

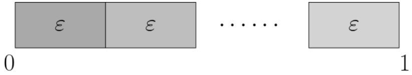
$$\Pr[\mathcal{A} \textit{ accepts } v_i | v_i > \max\{v_1, v_2 \cdots v_{i-1}\}] \in (p_i - \varepsilon, p_i + \varepsilon]$$

Proof.

For general i : Consider the complete graph on the vertex set S_{i-1}. Color the edge $(d_1, d_2, \cdots d_i)$ where $d_i > \{d_1, d_2, \cdots d_j\}$ with the corresponding color.

Let S_i be an infinite monochromatic clique





A $1/e$ Upper Bound

Theorem (Correa et al [CDFS19])

No algorithm can do better than $1/e$ competitive ratio for the unknown IID prophet inequality problem.

- For a fixed algorithm $\mathcal{A}$, design distribution F such that $\mathcal{A}$ only uses ordinal information on F
- Set up the support of F so that the maximum element contributes a $1 - o(1)$ fraction of the expected optimum
- Thus, $\mathcal{A}$ cannot do any better than $1/e$ for the distribution F

Outline of Talk

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- **Proof :** Set aside the first $o(1)$ fraction of values to be used as samples in the (completely) unknown IID prophet problem

Beating the $1/e$ Bound

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- **Proof :** Set aside the first $o(1)$ fraction of values to be used as samples in the (completely) unknown IID prophet problem
- Expected maximum of the remaining values is $1 - o(1)$ times the expected maximum of all the values

Beating the $1/e$ Bound

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- Already know that n samples suffice for $1/2$-competitive ratio. Can we do better, given that the distributions are identical?

Beating the $1/e$ Bound

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- **Approach 1 :** Independently simulate the other $n-1$ values using samples

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples

A $1-1/e$-Competitive Algorithm

- First, a $1-1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- **Key Idea 1:** The events $V_i > \tau_i$ are independent

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n - 1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- **Key Idea 1:** The events $V_i > \tau_i$ are independent
- On reaching the i-th value, probability of accepting it is $1/n$

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- **Key Idea 1:** The events $V_i > \tau_i$ are independent
- On reaching the i-th value, probability of accepting it is $1/n$
- **Key Idea 2 :** Conditioned upon accepting a value, its expected value is $\mathbb{E}[V_{\max}]$
- Consider the event that some value is accepted. This event happens with probability $\left(1 - \left(1 - \frac{1}{n}\right)^n\right)$

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- **Key Idea 1:** The events $V_i > \tau_i$ are independent
- On reaching the i-th value, probability of accepting it is $1/n$
- **Key Idea 2 :** Conditioned upon accepting a value, its expected value is $\mathbb{E}[V_{\max}]$
- Consider the event that some value is accepted. This event happens with probability $\left(1 - \left(1 - \frac{1}{n}\right)^n\right)$
- Putting the pieces together , the expected reward is

$$\left(1 - \left(1 - \frac{1}{n}\right)^n\right) \sum_{i=1}^{n} \frac{1}{n} . \mathbb{E}[V_{\max}]$$

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- Putting the pieces together , the expected reward is

$$\left(1 - \left(1 - \frac{1}{n}\right)^n\right) \sum_{i=1}^{n} \frac{1}{n} . \mathbb{E}[V_{\max}] \geq \left(1 - \frac{1}{e}\right) \mathbb{E}[V_{\max}]$$

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n-1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- Can achieve the same competitive ratio with $n-1$ samples by "recycling" rejected values as samples

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n - 1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- Can achieve the same competitive ratio with $n - 1$ samples by "recycling" rejected values as samples
- Maintain a set of $n - 1$ samples S
- Set threshold $\tau_i = \max S$. If V_i is rejected, swap in V_i into S with a uniformly random element

A $1 - 1/e$-Competitive Algorithm

- First, a $1 - 1/e$-approx algorithm with $n(n-1)$ samples
- For each $i \in [n]$, use $n - 1$ fresh samples to set a threshold τ_i as their maximum value. Accept V_i if $V_i > \tau_i$
- Can achieve the same competitive ratio with $n - 1$ samples by "recycling" rejected values as samples
- Maintain a set of $n - 1$ samples S
- Set threshold $\tau_i = \max S$. If V_i is rejected, swap in V_i into S with a uniformly random element
- **Claim :** Conditioned on arriving at the i-th value, the distribution of S is that of $n - 1$ "fresh" samples.

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- **Approach 1 :** Independently simulate the other $n-1$ values using samples

Beating the $1/e$ Bound

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- **Approach 1 :** Independently simulate the other $n-1$ values using samples
- Similar approach used by Azar et al. for bipartite matching

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- **Approach 1 :** Use multiple samples to generate independence between some suitable set of events
- **Approach 2 :** Use samples to approximate statistics used by the full information algorithm

Beating the $1/e$ Bound

- **Corollary:** Cannot do better than $1/e$ with $o(n)$ samples
- How to use $\Omega(n)$ samples to improve competitive ratio?
- **Approach 1 :** Use multiple samples to generate independence between some suitable set of events
- **Approach 2 :** Use samples to approximate statistics used by the full information algorithm
- Used by Runinstein et al [RWW20] to show a comptetive ratio of $0.745 - \varepsilon$ using $O_\varepsilon(n)$ samples

Outline of Talk

Conclusion

- Many prophet problems can be solved with a single sample, or a few samples from each distribution

Conclusion

- Many prophet problems can be solved with a single sample, or a few samples from each distribution
- Competitive ratios of these algorithms often match that of the full information algorithms

Conclusion

- Many prophet problems can be solved with a single sample, or a few samples from each distribution
- Competitive ratios of these algorithms often match that of the full information algorithms
- Also give simpler algorithms for the full information setting

Open Problems

- Gap between best competitive ratio for full information versus single sample for the matroid problem : $1/2$ versus $O(\log\log(\text{rank}))$

Open Problems

- Gap between best competitive ratio for full information versus single sample for the matroid problem : $1/2$ versus $O(\log\log(\text{rank}))$
- Want an exact ratio for Single Sample k-choice, the single sample algorithm's ratio is $1 - O\left(\frac{1}{\sqrt{k}}\right)$, while the full information algorithm is $1 - \frac{1}{\sqrt{k+3}}$

Open Problems

- Gap between best competitive ratio for full information versus single sample for the matroid problem : $1/2$ versus $O(\log\log(\text{rank}))$
- Want an exact ratio for Single Sample k-choice, the single sample algorithm's ratio is $1 - O\left(\frac{1}{\sqrt{k}}\right)$, while the full information algorithm is $1 - \frac{1}{\sqrt{k+3}}$
- Want to know the best possible ratio for the IID prophet problem with n samples (gap between 0.632 algorithm and 0.745 upper bound)

Open Problems

- Gap between best competitive ratio for full information versus single sample for the matroid problem : $1/2$ versus $O(\log\log(\text{rank}))$
- Want an exact ratio for Single Sample k-choice, the single sample algorithm's ratio is $1 - O\left(\frac{1}{\sqrt{k}}\right)$, while the full information algorithm is $1 - \frac{1}{\sqrt{k+3}}$
- Want to know the best possible ratio for the IID prophet problem with n samples (gap between 0.648 algorithm and 0.745 upper bound)
- Unknown IID Prophet Inequalities beyond the single choice problem

Thanks!