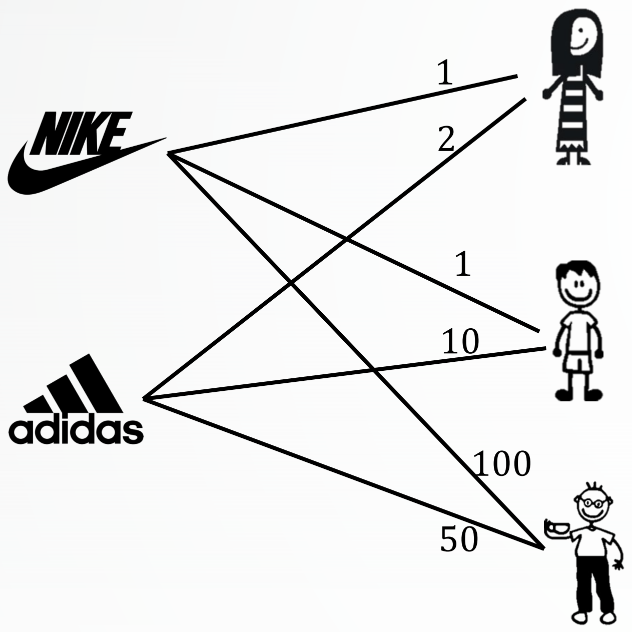
Competitive Analysis meets Stochastic Input: Secretary problems and Prophet inequalities

SHUCHI CHAWLA

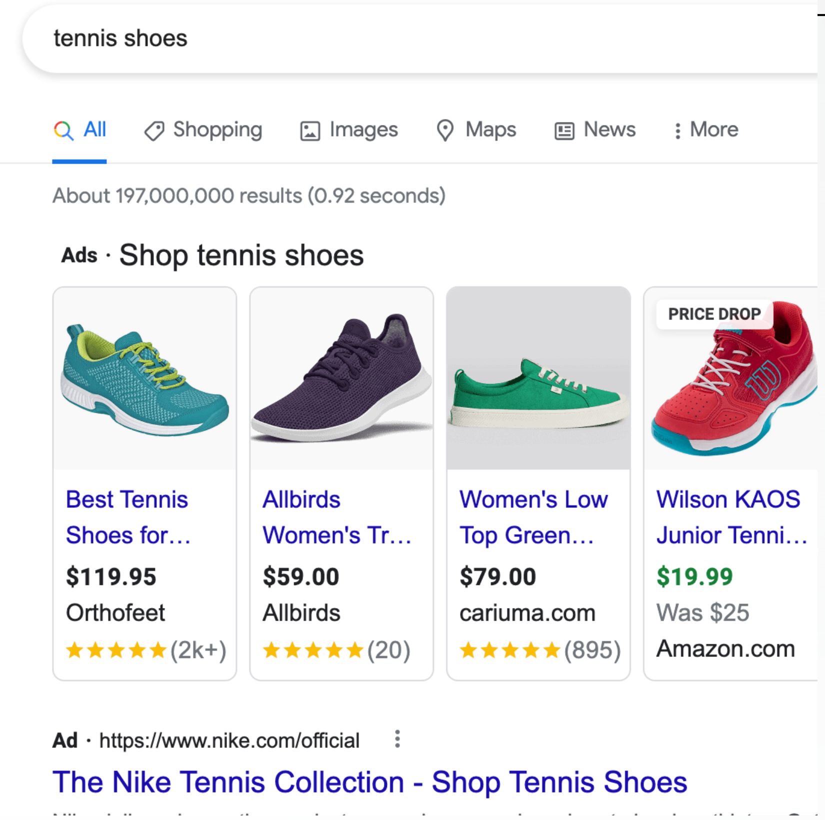
TEXAS
The University of Texas at Austin

Online maximum weight bipartite matching

Example 1: Matching advertisers to slots



Example 2: Uber driver accepting a customer



Competing against the hindsight optimum is hopeless in the worst case!

Incorporating data into the worst case model…

Purely worst-case

No information about input

Too pessimistic; Algorithms fine-tuned to unreasonable worst-case instances

Partial information a.k.a. semi-random models

- Coarse/limited info about input distribution
- Input distribution unknown but from a "nice" class
- Input distribution unknown but we have sample access
- Input is part stochastic and part adversarial

Purely stochastic

Full information about the input generation process

Too optimistic; Algorithms not robust to changes in model

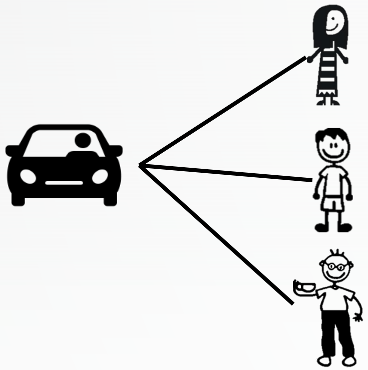
This talk

Ellen's talk

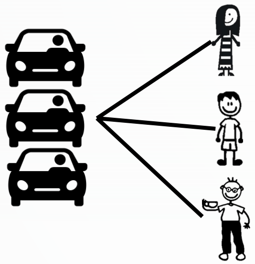
Anupam's talk

Online selection

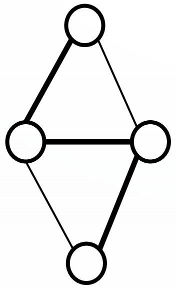
- n elements arrive in sequence; each with weight W_i.
- Algorithm makes irrevocable accept/reject decision for each element. $S \leftarrow$ accepted elements
- We require $S \in \mathcal{F}$ for a given downwards closed feasibility constraint $\mathcal{F}$.



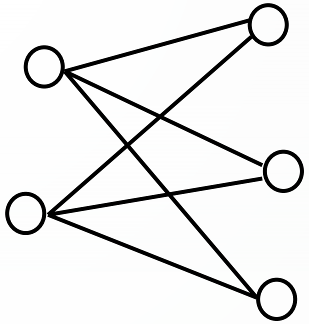
Single unit



k-unit

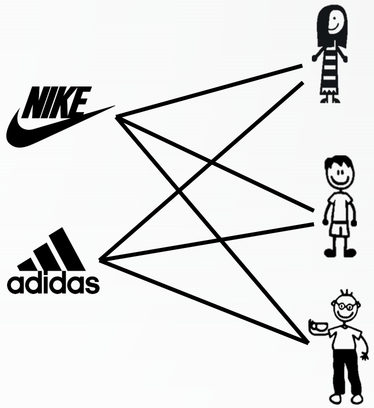


Matroids



Matching

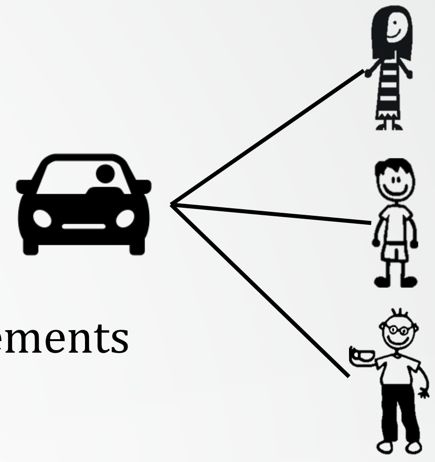
vertices known;
edges arrive over time



Online resource allocation

RHS vertices arrive over time

Online selection: semi-random models



- n elements arrive in sequence; each with weight W_i.
- Algorithm makes irrevocable accept/reject decision for each element. $S \leftarrow$ accepted elements
- We require $S \in \mathcal{F}$ for a given downwards closed feasibility constraint $\mathcal{F}$.

The Secretary Problem setting:
[Dynkin'63]

- Weights are adversarial
- Arrival order is uniformly random

Hindsight OPT = $\max_i W_i$

The Prophet Inequality setting:
[Krengel & Sucheston'77, Samuel-Cahn'84]

- Weights drawn from known distributions
- Arrival order is adversarial

Hindsight OPT = $\mathrm{E}\left[\max_i W_i\right]$

$$\text{Competitive Ratio} = \max_{\text{instances I}} \frac{\mathrm{E}_{\text{randomness in I}}[\text{Hindsight}-\text{OPT(I)}]}{\mathrm{E}_{\text{randomness in I, ALG}}[\text{ALG(I)}]}$$

Upshot: Unlike for the purely worst case, these models admit constant competitive ratios.

Rest of this talk

- Prophet inequalities
  - Contention Resolution Schemes
  - Combinatorial approaches
  - Online resource allocation
- Secretary problem
  - Explore and exploit
  - Learning duals
  - Learning the primal
- Some extensions

Prophet Inequality for single unit

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

- Hindsight-OPT = $\mathrm{E}\left[\max_i W_i\right] \;=\; \sum_i x_i\, \mathrm{E}[W_i \mid \text{OPT selects } i]$
- Let $x_i = \Pr[\text{OPT selects } i]$

Idea: try to mimic the optimal probabilities of selection.

[Chawla Hartline Malec Sivan'10, Alaei'11]

- When element 1 arrives, accept w.p. x_1
  - Set acceptance threshold t_1 such that $\Pr[W_1 \geq t_1] = x_1$. Note: $E[W_1 | W_1 \geq t_1] \geq \mathrm{E}[W_1 \mid \text{OPT selects } 1]$
- When element 2 arrives, accept w.p. x_2. (Set threshold t_2 such that $\Pr[W_2 \geq t_2] = x_2$.)
- And so on…

$$\text{ALG} = \sum_i \Pr[\text{ALG reaches } i] . x_i . E[W_i | W_i \geq t_i]$$

??

Prophet Inequality for single unit

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

- Hindsight-OPT = $\mathrm{E}\left[\max_i W_i\right] = \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$
- Let $x_i = \Pr[\text{OPT selects } i]$

Idea: try to mimic the optimal probabilities of selection.

[Chawla Hartline Malec Sivan'10, Alaei'11]

- When element 1 arrives, accept w.p. $x_1/2$
  - Set acceptance threshold t_1 such that $\Pr[W_1 \geq t_1] = x_1/2.$ Note: $E[W_1 | W_1 \geq t_1] \geq \mathrm{E}[W_1 \mid \text{OPT selects } 1]$
- When element 2 arrives, accept w.p. $x_2/2.$ (Set threshold t_2 such that $\Pr[W_2 \geq t_2] = x_2/2.$)
- And so on…

$$\text{ALG} = \sum_i \Pr[\text{ALG reaches } i] \,.\, x_i/2.\, E[W_i | W_i \geq t_i] \;\; \geq \frac{1}{4} \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$$

??

$= 1 - \Pr[\text{a previous element was accepted}] \geq 1 - \Sigma_i x_i/2 \geq 1/2$

Prophet Inequality for single unit

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

- Hindsight-OPT = $\mathrm{E}\left[\max_i W_i\right] = \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$
- Let $x_i = \Pr[\text{OPT selects } i]$

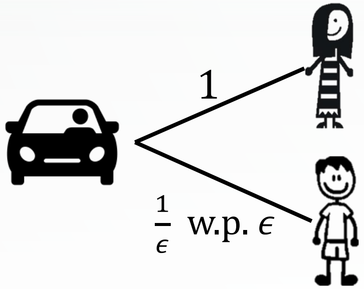
Idea: try to mimic the optimal probabilities of selection.

[Chawla Hartline Malec Sivan'10, Alaei'11]

Slightly better approach:

- Accept each element i with probability exactly $x_i/2$
  - Compute probability of reaching element i $\leftarrow \alpha_i$
  - Set acceptance threshold t_i such that $\Pr[W_i \geq t_i] = x_i/2\alpha_i$.

Note: $\alpha_i \geq 1/2$, so, $E[W_i | W_i \geq t_i] \geq \mathrm{E}[W_i \mid \text{OPT selects } i]$



$$\text{ALG} = \sum_i \underbrace{\Pr[\text{ALG reaches } i]}_{\alpha_i} \cdot \frac{x_i}{2\alpha_i} \cdot E[W_i | W_i \geq t_i] \geq \frac{1}{2} \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$$

Tight!

Prophet Inequality for matchings

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

- Hindsight-OPT = E[max weight matching] $= \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$
- Let $x_i = \Pr[\text{OPT selects } i]$

Idea: try to mimic the optimal probabilities of selection.

[Chawla Hartline Malec Sivan'10, Alaei'11]

Simple "collision-avoidance" algorithm:

- When element i arrives, if feasible to accept, then accept w.p. $x_i/3$
- $\Pr[i \text{ remains unblocked}] \geq 1 - \Pr[i\text{'s first end point is matched}] - \Pr[i\text{'s second end point is matched}] \geq 1 - \frac{1}{3} - \frac{1}{3} = \frac{1}{3}$

$$\text{ALG} = \sum_i \Pr[i \text{ remains unblocked when reached}] \cdot \frac{x_i}{3} \cdot E[W_i \mid W_i \geq t_i] \;\geq\; \frac{1}{9} \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$$

$\geq 1/3$

A general approach: OCRS

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

- Let $x_i = \Pr[\text{OPT selects } i]$
- Hindsight-OPT $= \sum_i x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$

Idea: try to mimic the optimal probabilities of selection.

c - Online Contention Resolution Scheme:

[Chekuri Vondrak Zenklusen'14, Feldman Svensson Zenklusen'16]

- Online procedure for determining the probability of accepting an element that arrives, if unblocked.
- Goal: Accept each element i with probability $y_i \coloneqq c.x_i$
- Show: Each element remains unblocked with probability $\geq c$.

$$\text{ALG} = \sum_i y_i . E[W_i | W_i \geq t_i] \geq \sum_i c.x_i \, \mathrm{E}[W_i \mid \text{OPT selects } i]$$

c-OCRS $\Longrightarrow$ c-competitive Prophet Inequality

OCRSs exist for many set systems. k-unit: $(1 - 1/\sqrt{k+3})$–OCRS [Alaei'11]; General matroids: ½-OCRS [Feldman Svensson Zenklusen'16].

[Lee-Singla'18]: Prophet Inequalities and OCRS are essentially equivalent

Combinatorial approaches

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

[Samuel-Cahn'84]: 2-competitive single-unit Prophet Inequality

- Find a threshold t such that $\Pr[\exists i \text{ with } W_i \geq t] = \frac{1}{2}$.
- Pick the first element that exceeds t ← Price for "selling" the unit

Alternatively: Set $t = \frac{1}{2}\text{OPT}$

Another alternative: pick any value between the two!

t if the unit sells

$W_i - t$ if the unit is sold to i

Proof approach: break up the reward earned into "seller's revenue" and "buyer's utility" [Feldman Gravin Lucier'15]

$OPT \leq t + \max_i (W_i - t)^+$ whereas $ALG \geq t.\Pr[\text{unit sells}] + \sum_i (W_i - t)^+ . \Pr[\text{unit didn't sell before } i]$

$$\geq t.\Pr[\text{unit sells}] + \sum_i (W_i - t)^+ . \Pr[\text{unit doesn't sell}]$$

$$\geq \frac{1}{2} t + \frac{1}{2} \sum_i (W_i - t)^+$$

$\Longrightarrow$ 2-approximation

Combinatorial approaches

Model:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

[Samuel-Cahn'81]: 2-competitive single-unit Prophet Inequality

- Find a threshold t such that $\Pr[\exists i \text{ with } W_i \geq t] = \frac{1}{2}$.
- Pick the first element that exceeds t

Alternatively: Set $t = \frac{1}{2}\mathrm{OPT}$

Another alternative: pick any value between the two!

Extensions to k-units with static thresholds:

- [Hajiaghayi Kleinberg Sandholm'07] pick t such that $\mathrm{E}[\#i \text{ with } W_i \geq t] \approx k - \sqrt{k \log k}$.

  $\Longrightarrow 1 - \Theta(\sqrt{\log k / k})$ asymptotically
- [Chawla Lykouris Devanur'21] pick t such that $\mathrm{E}[\text{fraction of units unsold}] = \Pr[\text{all units sold out}]$

  $\Longrightarrow 1 - \Theta(\sqrt{\log k / k})$ for all k

Extension to matroids: "Balanced" thresholds [Kleinberg Weinberg'12]

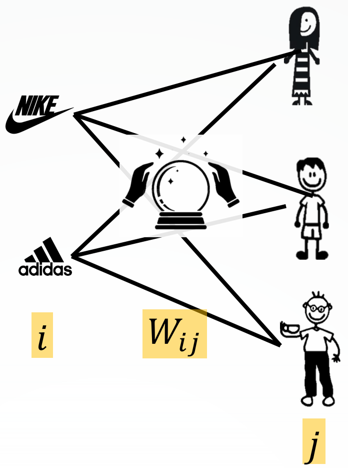
- Set $t_i = \frac{1}{2}$. the expected "opportunity cost" of accepting i.
- 2-approximation for general matroids

Benefits of a single static threshold:

- One parameter to learn
- Robustness to errors
- Nice fairness & incentive properties

Downside: not always optimal

Online resource allocation



Model:

- Shoppers arrive in fixed but arbitrary order
- Weights of all edges incident on a shopper are revealed at once
- Weights drawn from known distributions: $W_{ij} \sim D_{ij}$

- Can use the matching OCRS as before

  But can potentially do much better!

[Feldman Gravin Lucier'15]: pricing-based algorithm

- Set a price for item i, $t_i = \frac{1}{2}$E[contribution of i to OPT]
- When shopper j arrives, assign to it the available item that maximizes $W_{ij} - t_i$ ← Economic interpretation: shoppers maximize their utility

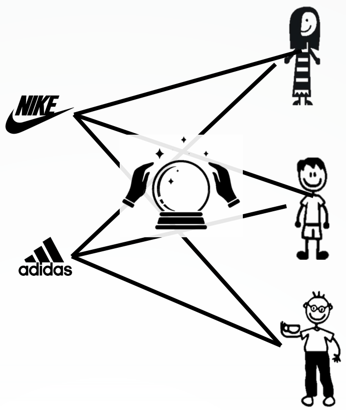
- Suppose in OPT, i is assigned to $j^*(i)$.
- Item i's contribution to the algorithm $\geq t_i . \mathbb{I}$[item i sells] + $\left(W_{i j^*(i)} - t_i\right) . \mathbb{I}$[item i doesn't sell]
- Taking expectations, i's contribution $\geq t_i$. Pr[i sells] + t_i. Pr[i doesn't sell] $= \frac{1}{2}$E[contribution of i to OPT]

$\Longrightarrow$ 2-approximation

Online resource allocation

- Can use the matching OCRS as before

  But can potentially do much better!



Model:

- Shoppers arrive in fixed but arbitrary order
- Weights of all edges incident on a shopper are revealed at once
- Weights drawn from known distributions: $W_{ij} \sim D_{ij}$

[Feldman Gravin Lucier'15]: pricing-based 2-approximation algorithm

Can extend these ideas to shoppers purchasing bundles of items

- XOS; MPH hierarchy [Feldman Gravin Lucier'15, Dutting Feldman Kesselheim Lucier'17]
- subadditive values [Dutting Kesselheim Lucier'20]
- intervals or paths in networks [Chawla Miller Teng'19]

With large item multiplicities and other structure on weights, dual prices provide a good approximation

[Chawla Devanur Holroyd Karlin Martin Sivan'17]

Rest of this talk

- Prophet inequalities
  - Contention Resolution Schemes
  - Combinatorial approaches
  - Online resource allocation

- Secretary problem
  - Explore and exploit
  - Learning duals
  - Learning the primal

- Some extensions

Secretary Problem

Model:
- Elements arrive in uniformly random order
- Weights are adversarial

Basic idea: use the first few elements as a sample to "learn" the instance. [Dynkin'63]

- Phase 1 (explore): Reject the first $n/2$ elements; Let $t =$ maximum weight observed
- Phase 2 (exploit): Among the remaining, pick the first element i with $W_i \geq t$

How long to explore?

What to learn?

How to exploit?

$$\Pr[\text{largest weight picked}] \geq \Pr\left[t = W_{(2)}, \text{and}, W_{(1)} \text{ appears in the second half}\right] \geq \frac{1}{4}$$

$\Longrightarrow$ 4-competitive algorithm

Multiple alternations of explore and exploit?

k-unit secretary: explore for $n/\text{poly}(k)$ steps $\Longrightarrow 1 - O(1/\text{poly}(k))$ approximation

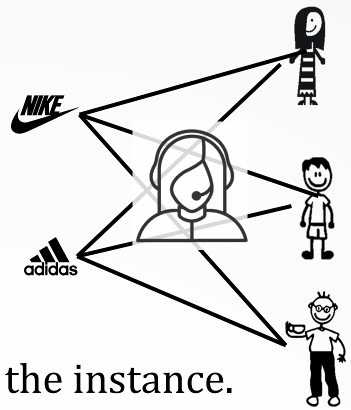
Improved k-unit secretary: geometrically increasing explore/exploit phases; in each phase, exploit using the threshold learned in previous phases $\Longrightarrow 1 - O(1/\sqrt{k})$ approximation [Kleinberg'05]

Rank-k matroid: greedily pick largest feasible set crossing a single threshold $\Longrightarrow O(\log k)$ approx. [Babaioff Immorlica Kleinberg'07]

Best known: $O(\log \log k)$ [Lachish'15, Feldman Svensson Zenklusen'16]

[Dughmi'21]: Connection between matroid secretary and matroid OCRS $\longrightarrow$ $O(1)$??

Online Resource Allocation



Model:

- Shoppers arrive in uniformly random order
- Weights of all edges incident on a shopper are revealed at once
- Weights are adversarial

Basic idea: use the first few elements as a sample to “learn” the instance.

Primal program:

$\max \sum_{i,j} x_{i,j} W_{i,j}$ subject to:

$\sum_j x_{i,j} \leq 1$ for all shoppers i

$\sum_i x_{i,j} \leq k_j$ for all items j

$x_{i,j} \geq 0$ for all i and j

Dual program:

$\min \sum_j k_j t_j + \sum_i u_i$ subject to:

$u_i \geq W_{i,j} - t_j$ for all i, j

$u_i, p_j \geq 0$ for all i, j

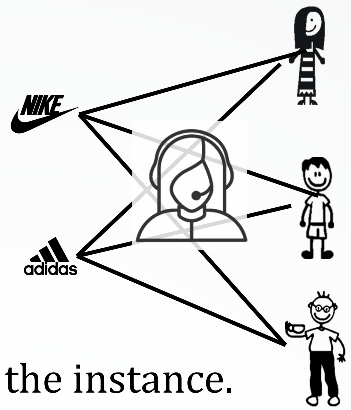
Given the “correct” dual:

- Set t_j as the price for j.
- Every shopper, on arrival, should choose the item maximizing $W_{i,j} - t_j$

Dual-learning algorithm [Devanur Hayes’09, Agarwal Wang Ye’14]:

- Solve the dual program over the first ϵn samples with scaled down capacities to learn the dual prices
- Exploit using dual prices
- Concentration bounds $\Longrightarrow$ learned duals are close to the optimal dual

Online Resource Allocation



Model:

- Shoppers arrive in uniformly random order
- Weights of all edges incident on a shopper are revealed at once
- Weights are adversarial

Basic idea: use the first few elements as a sample to "learn" the instance.

Primal program:

$\max \sum_{i,j} x_{i,j} W_{i,j}$ subject to:

$\sum_j x_{i,j} \leq 1$ for all shoppers i

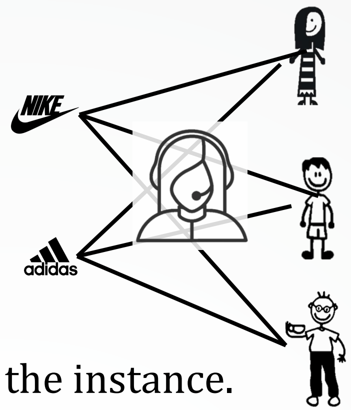
$\sum_i x_{i,j} \leq k_j$ for all items j

$x_{i,j} \geq 0$ for all i and j

Primal-learning algorithm [Kesselheim Radke Tonnis Vocking'14]:

- At every step, solve the primal with appropriately scaled down capacities.
- Round the component corresponding to shopper i
- If the match suggested by the primal is feasible, include it in solution.

Online Resource Allocation



Model:

- Shoppers arrive in uniformly random order
- Weights of all edges incident on a shopper are revealed at once
- Weights are adversarial

Basic idea: use the first few elements as a sample to "learn" the instance.

Primal program:

$\max \sum_{i,j} x_{i,j} W_{i,j}$ subject to:

$\sum_j x_{i,j} \leq 1$ for all shoppers i

$\sum_i x_{i,j} \leq 1$ for all items j

$x_{i,j} \geq 0$ for all i and j

Primal-learning algorithm [Kesselheim Radke Tonnis Vocking'13]:

- Reject the first n/e requests.
- At subsequent requests i :
  - Find optimal matching over shoppers $\{1, \dots, i\}$; Say i is matched to $j^*(i)$
  - If $j^*(i)$ is available, match i to it.

Analysis in two parts:

Part 1: For any i, the expected weight of $(i, j^*(i))$ is at least OPT/n.

$\frac{i}{n} OPT \times \frac{1}{i}$

Part 2: The probability that $j^*(i)$ is blocked is small:

- The probability that $j^*(i)$ is matched to $i' < i$ is at most $1/i'$.
- Net "unblocking" probability $\geq \prod_{i'=\frac{n}{e}}^{i-1} \left(1 - \frac{1}{i'}\right) \approx \frac{\frac{n}{e}}{i} \geq \frac{1}{e}$

$\Rightarrow$ total contribution of $\frac{1}{e}\left(1 - \frac{1}{e}\right)$ OPT

A recap of techniques

Secretary Problem:

- Elements arrive in uniformly random order
- Weights are adversarial

Prophet Inequality:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

Secretary Problem techniques:

– Explore and exploit
– Learning duals
– Learning the primal

Prophet Inequality techniques:

– Contention Resolution Schemes
– Combinatorial approaches: balanced prices
– Online resource allocation: balanced prices; dual prices

Some extensions

Secretary Problem:

- Elements arrive in uniformly random order
- Weights are adversarial

Prophet Inequality:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

Many possible variants:

- I.i.d. weights [Correa Foncea Hoeksma Ossterwijk Vredeveld'17]
- Correlated weight distributions [Chawla Malec Sivan'15, Immorlica Singla Waggoner'20]
- Unknown distributions but with sample access [Azar Kleinberg Weinberg'14, Correa Dutting Fischer Schewior'19, Rubinstein Wang Weinberg'20]
- Best/constrained order prophet inequality [Chawla Hartline Malec Sivan'10, Agrawal Sethuraman Zhang'20, Peng Tang'22, Arsenis Drosis Kleinberg'21]
- Non-uniform distribution or corruption over orderings [Kesselheim Kleinberg Niazadeh'15, Bradac Gupta Singla Zuzic'20]
- Prophet secretary: known weight distributions AND random order of arrival [Esfandiari Hajiaghayi Liaghat Monemizadeh'15, Azar Chiplunkar Kaplan'18]
- Non-linear objectives [Feldman Zenklusen'15, Rubinstein Singla'17]
- Stochastic departures [Kessel Shameli Saberi Wajc'22]

Some extensions

Secretary Problem:

- Elements arrive in uniformly random order
- Weights are adversarial

Prophet Inequality:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

Secretary/prophet models for other optimization problems:

- Bin packing [Kenyon'96]
- Online Steiner tree [Garg Gupta Leonardi Sankowski'08]
- Set cover; facility location [Grandoni Gupta Leonardi Miettinen Sankowski Singh'08]
- Online independent set [Gobel Hoefer Kesselheim Schleiden Vocking'14]
- k-server [Dehghani Ehsani Hajiaghayi Liaghat Seddighin'17]

Stochastic probing [Guha Munagala'07, Gupta Nagarajan'13, Gupta Nagarajan Singla'16, '17]

Price of information (Pandora's box) problems [Kleinberg Waggoner Weyl'16, Singla'18, Chawla Gergatsouli Teng Tzamos Zhang'20]

Secretary Problem:

- Elements arrive in uniformly random order
- Weights are adversarial

Prophet Inequality:

- Elements arrive in fixed but arbitrary order
- Weights are drawn from known distributions: $W_i \sim D_i$

Questions?