Near Optimal Online Auctions

Jason D. Hartline
Microsoft Research – Silicon Valley

(Joint work with Avrim Blum)

October 28, 2004

Online Auction Problem

Online Auction Problem: [Bar-Yossef, Hildrum, Wu 2002]

- Seller has unlimited supply of an item (e.g., digital good).
- Bidders arrive one at a time, bid $b_1, b_2, \ldots \in [1, h]$
- Auctioneer decides sale price (or reject) before next bidder arrives.
- **Goal:** maximize auctioneer profit!

Online Auction Problem

Online Auction Problem: [Bar-Yossef, Hildrum, Wu 2002]

- Seller has unlimited supply of an item (e.g., digital good).
- Bidders arrive one at a time, bid $b_1, b_2, \ldots \in [1, h]$
- Auctioneer decides sale price (or reject) before next bidder arrives.
- **Goal:** maximize auctioneer profit!

Two Difficulties:

1. Bidders' values are private (need game-theory).
2. We do not know future (need online analysis).

Online Auction Problem

Online Auction Problem: [Bar-Yossef, Hildrum, Wu 2002]

- Seller has unlimited supply of an item (e.g., digital good).
- Bidders arrive one at a time, bid $b_1, b_2, \ldots \in [1, h]$
- Auctioneer decides sale price (or reject) before next bidder arrives.
- **Goal:** maximize auctioneer profit!

Two Difficulties:

1. Bidders' values are private (need game-theory).

   Truthful mechanism design: price offered bidder i not function of b_i.
2. We do not know future (need online analysis).

Online Auction Problem

Online Auction Problem: [Bar-Yossef, Hildrum, Wu 2002]

- Seller has unlimited supply of an item (e.g., digital good).
- Bidders arrive one at a time, bid $b_1, b_2, \ldots \in [1, h]$
- Auctioneer decides sale price (or reject) before next bidder arrives.
- **Goal:** maximize auctioneer profit!

Two Difficulties:

1. Bidders' values are private (need game-theory).
   Truthful mechanism design: price offered bidder i not function of b_i.
2. We do not know future (need online analysis).
   Online algorithm: price offered bidder i not function of future bids.

Online Auction Problem

Online Auction Problem: [Bar-Yossef, Hildrum, Wu 2002]

- Seller has unlimited supply of an item (e.g., digital good).
- Bidders arrive one at a time, bid $b_1, b_2, \ldots \in [1, h]$
- Auctioneer decides sale price (or reject) before next bidder arrives.
- **Goal:** maximize auctioneer profit!

Two Difficulties:

1. Bidders' values are private (need game-theory).

   Truthful mechanism design: price offered bidder i not function of b_i.

2. We do not know future (need online analysis).

   Online algorithm: price offered bidder i not function of future bids.

Conclusion: offer for bidder i based only on prior bids: $b_1, \ldots, b_{i-1}$.

Performance Guarantees?

Definition: $\mathrm{OPT} =$ optimal single price profit.

Performance Guarantees?

Definition: $\mathrm{OPT} =$ optimal single price profit.

Assumption: bids in $[1, h]$.

Performance Guarantees?

Definition: $\mathrm{OPT} =$ optimal single price profit.

Assumption: bids in $[1, h]$.

Goal: $\mathbf{E}[\text{Profit}] \geq \mathrm{OPT}/\beta - \gamma h$.

- $\beta =$ approximation ratio.
- $\gamma h =$ additive loss.

Overview

$\Longrightarrow$ 1. Standard Expert Learning Algorithm (Weighted Majority).

2. Online (Expert) Learning $\Rightarrow$ Online Auction
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$, given $[1,h]$.
   [Blum Kumar Rudra Wu 2003]

3. Kalai's Expert Learning Algorithm & Analysis.

4. Modifications of Kalai's Expert Algorithm for Online Auctions.
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon})$.

5. An Application: Attribute Auctions.

Online Learning

Expert Online Learning Problem:

In round i:

1. Each of k experts propose a strategy.
2. We choose an expert's strategy.
3. Find out how each strategy performed (payoff)

Goal: Obtain payoff close to single best expert overall (in hindsight).

Online Learning

Expert Online Learning Problem:

In round i:

1. Each of k experts propose a strategy.
2. We choose an expert's strategy.
3. Find out how each strategy performed (payoff)

Goal: Obtain payoff close to single best expert overall (in hindsight).

Weighted Majority Algorithm: (for round i)

Let h be maximum payoff. For expert j, let s_j be total payoff thus far.

Choose expert j's strategy with probability proportional to $2^{s_j/h}$.

Online Learning

Expert Online Learning Problem:

In round i:

1. Each of k experts propose a strategy.
2. We choose an expert's strategy.
3. Find out how each strategy performed (payoff)

Goal: Obtain payoff close to single best expert overall (in hindsight).

Weighted Majority Algorithm: (for round i)

Let h be maximum payoff. For expert j, let s_j be total payoff thus far.

Choose expert j's strategy with probability proportional to $2^{s_j/h}$.

Result: $\mathbf{E}[\text{payoff}] = \mathrm{OPT}/2 - O(h \log k)$.

Online Learning for Online Auctions

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

Online Learning for Online Auctions

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

1. Expert j's strategy: "use sale price p_j".

Online Learning for Online Auctions

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

1. Expert j's strategy: "use sale price p_j".

2. Best expert $\Rightarrow$ best single price profit.

Online Learning for Online Auctions

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Result: **E**[profit] $= \mathrm{OPT}/4 - O(h \log\log h)$.

Application: (to online auctions) [Blum Kumar Rudra Wu 2003]

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Result: **E**[profit] $= \mathrm{OPT}/4 - O(h \log\log h)$.

Drawbacks:

- $h \log\log h$ additive loss.
- Must know h in advance.

Overview

1. Standard Expert Learning Algorithm (Weighted Majority).

2. Online (Expert) Learning $\Rightarrow$ Online Auction
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$, given $[1,h]$.
   [Blum Kumar Rudra Wu 2003]

$\Longrightarrow$ 3. Kalai's Expert Learning Algorithm & Analysis.

4. Modifications of Kalai's Expert Algorithm for Online Auctions.
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon})$.

5. An Application: Attribute Auctions.

Expert Learning via Hallucination

Kalai's Experts Algorithm:

1. (initialization) For each expert, j, hallucinate initial score, s_j, as:

$$h \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

Expert Learning via Hallucination

Kalai's Experts Algorithm:

1. (initialization) For each expert, j, hallucinate initial score, s_j, as:

$$h \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

Result: $\mathbf{E}[\text{profit}] = \mathrm{OPT}/2 - O(h \log k)$.

Kalai's Analysis

Definitions: (random variables)

$$M = \text{final } \max_j s_j$$

$$M_i = \text{change to } \max_j s_j \text{ in round } i$$

$$H = \text{maximum hallucination}$$

Kalai's Analysis

Definitions: (random variables)

M = final $\max_j s_j$

M_i = change to $\max_j s_j$ in round i

H = maximum hallucination

P = algorithm profit

P_i = profit in round i

Kalai's Analysis

Definitions: (random variables)

$M =$ final $\max_j s_j$	$P =$ algorithm profit
$M_i =$ change to $\max_j s_j$ in round i	$P_i =$ profit in round i
$H =$ maximum hallucination	

$$M = H + \sum_i M_i \qquad\qquad P = \sum_i P_i$$

Kalai's Analysis

Definitions: (random variables)

$$M = \text{final } \max_j s_j \qquad P = \text{algorithm profit}$$

$$M_i = \text{change to } \max_j s_j \text{ in round } i \qquad P_i = \text{profit in round } i$$

$$H = \text{maximum hallucination}$$

$$M = H + \textstyle\sum_i M_i \qquad P = \textstyle\sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k)$.

Kalai's Analysis

Definitions: (random variables)

M = final $\max_j s_j$

M_i = change to $\max_j s_j$ in round i

H = maximum hallucination

P = algorithm profit

P_i = profit in round i

$$M = H + \sum_i M_i$$

$$P = \sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2.$

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k).$

Proof:

- $\mathbf{E}[P] = \sum_i \mathbf{E}[P_i]$.

Kalai's Analysis

Definitions: (random variables)

M = final $\max_j s_j$

M_i = change to $\max_j s_j$ in round i

H = maximum hallucination

P = algorithm profit

P_i = profit in round i

$$M = H + \sum_i M_i$$

$$P = \sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k)$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2}\sum_i \mathbf{E}[M_i]}^{\text{from Lemma}}$.

Kalai's Analysis

Definitions: (random variables)

M = final $\max_j s_j$

M_i = change to $\max_j s_j$ in round i

H = maximum hallucination

P = algorithm profit

P_i = profit in round i

$$M = H + \sum_i M_i \qquad P = \sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k)$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.

Kalai's Analysis

Definitions: (random variables)

M = final $\max_j s_j$

M_i = change to $\max_j s_j$ in round i

H = maximum hallucination

P = algorithm profit

P_i = profit in round i

$$M = H + \sum_i M_i \qquad\qquad P = \sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k)$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}(\mathbf{E}[M] - \mathbf{E}[H])$.
- $\mathbf{E}[M] \geq \mathrm{OPT}$ and $\mathbf{E}[H] = O(h \log k)$.

Kalai's Analysis

Definitions: (random variables)

$$M = \text{final } \max_j s_j \qquad P = \text{algorithm profit}$$

$$M_i = \text{change to } \max_j s_j \text{ in round } i \qquad P_i = \text{profit in round } i$$

$$H = \text{maximum hallucination}$$

$$M = H + \sum_i M_i \qquad P = \sum_i P_i$$

Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/2 - O(h \log k)$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}$ and $\mathbf{E}[H] = O(h \log k)$.
- **Result:** $\mathbf{E}[P] \geq \frac{1}{2}\mathrm{OPT} - O(h \log k)$.

Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2.$

Proof: (after round i)

1. Tally expert "raw scores" (without hallucination) after round i.

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.

Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2.$

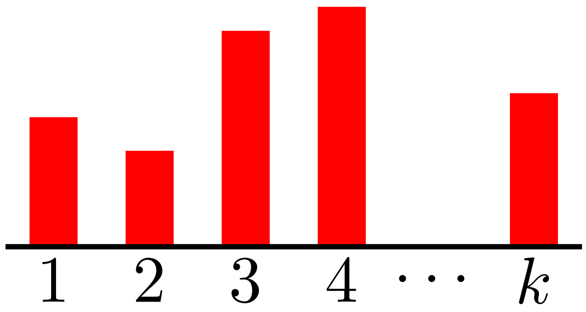
Proof: (after round i)

1. Tally expert "raw scores" (without hallucination) after round i.

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2.$

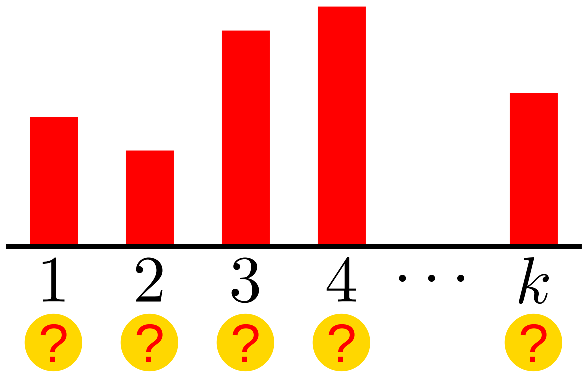
Proof: (after round i)

1. Tally expert "raw scores" (without hallucination) after round i.

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

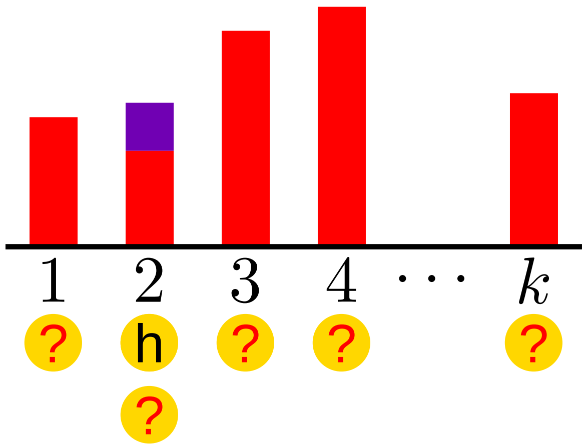
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

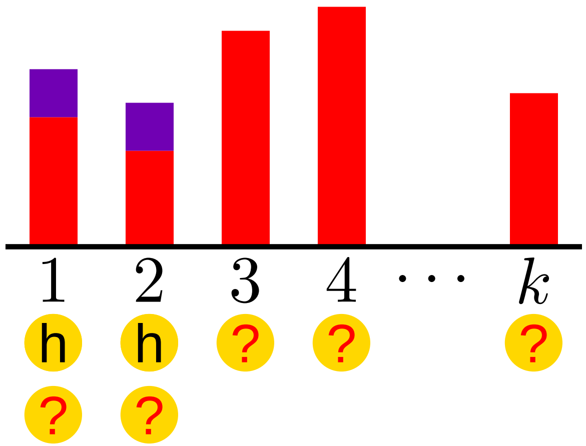
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

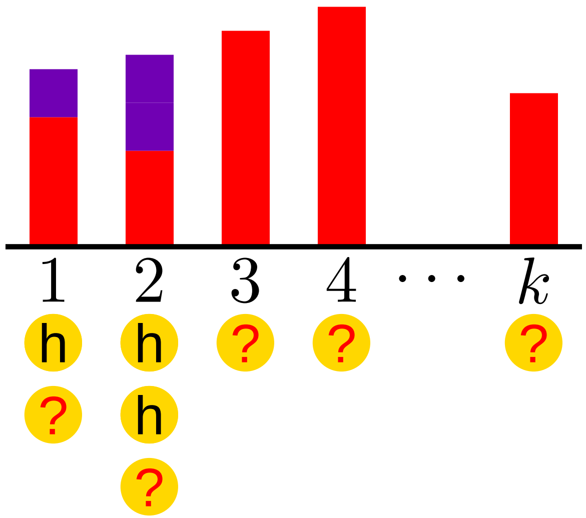
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2.$

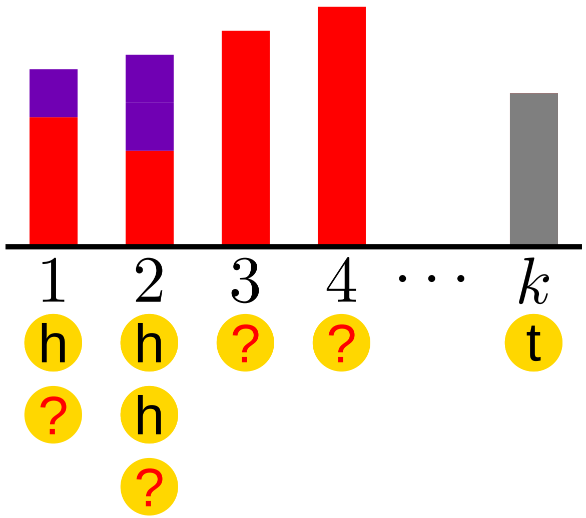
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

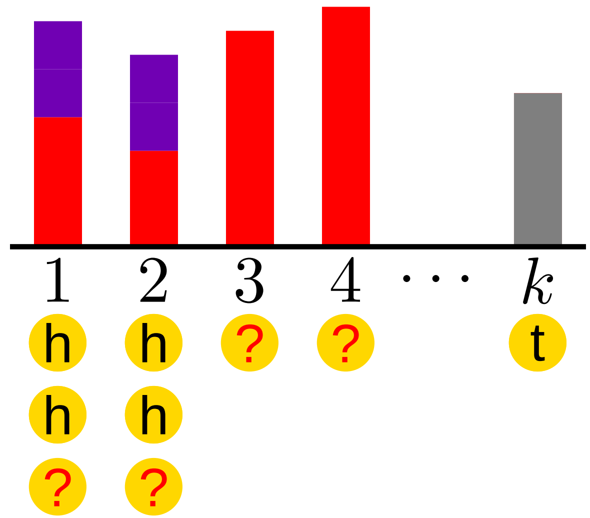
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

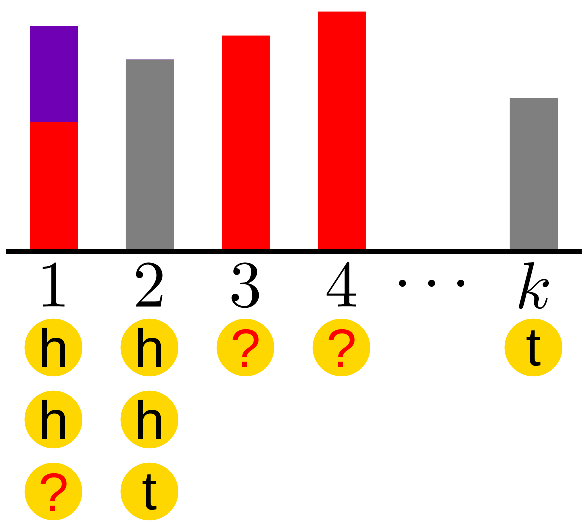
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

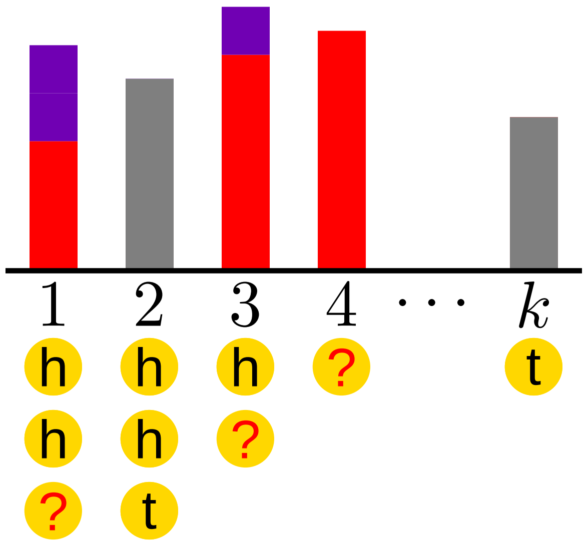
Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Proof:

1. Tally expert scores up to including round i (without hallucination).
2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

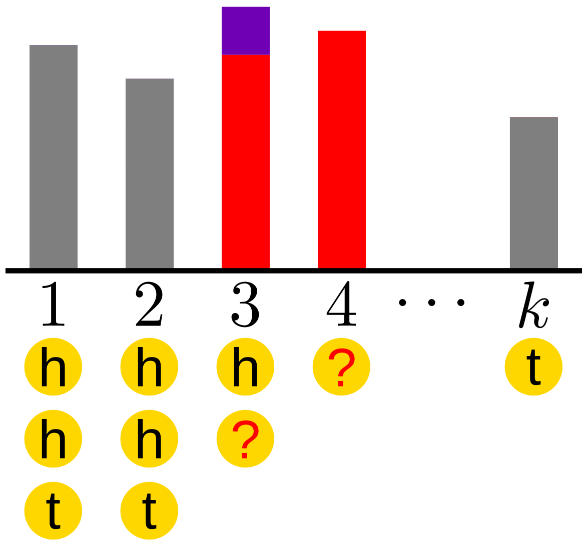
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

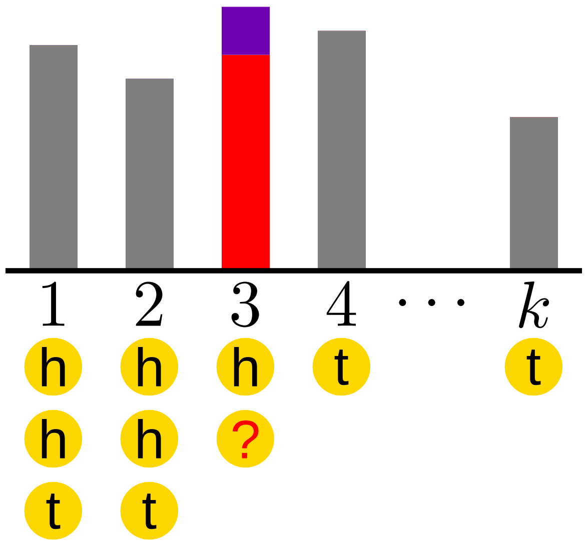
Proof:

1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.



Proof of Lemma

Recall Lemma: $\mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Proof:

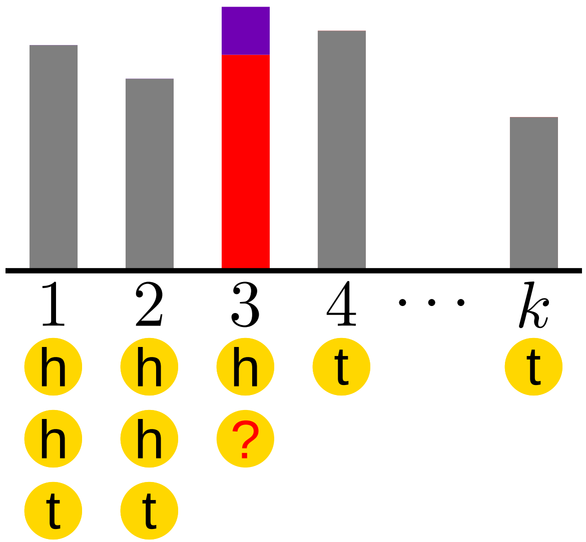
1. Tally expert scores up to including round i (without hallucination).

2. Re-hallucinate (repeat until one expert is left):

   Flip coin of expert with lowest score

   - **Heads:** add h to expert's score.
   - **Tails:** discard expert.

3. Remaining expert j: best and still has coin.



Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Proof of Lemma (cont)

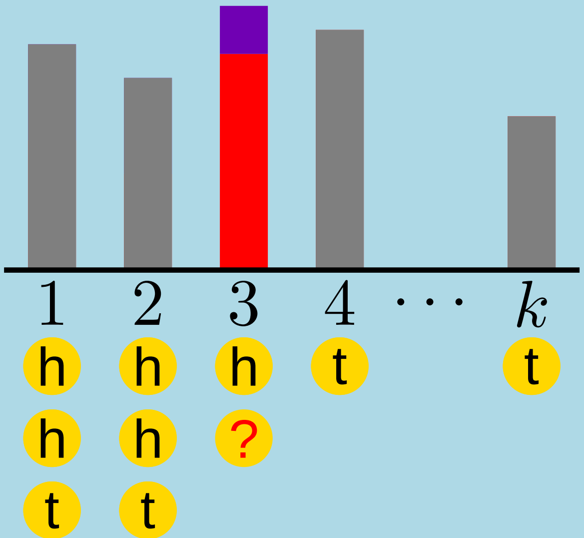
Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.	**Case 2:** j's coin flips tails.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.

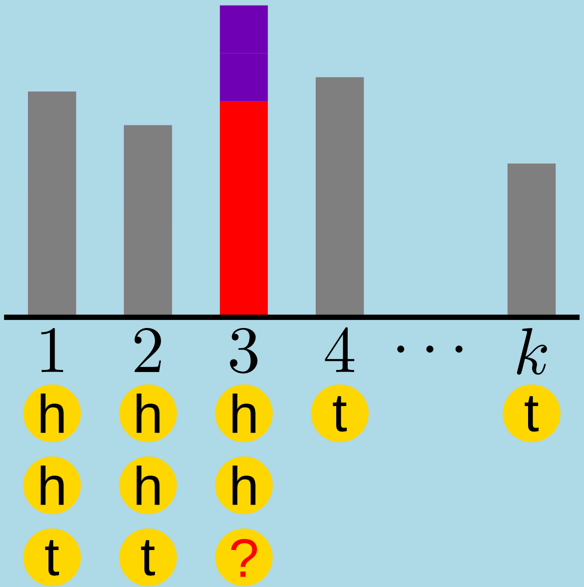


Case 2: j's coin flips tails.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.

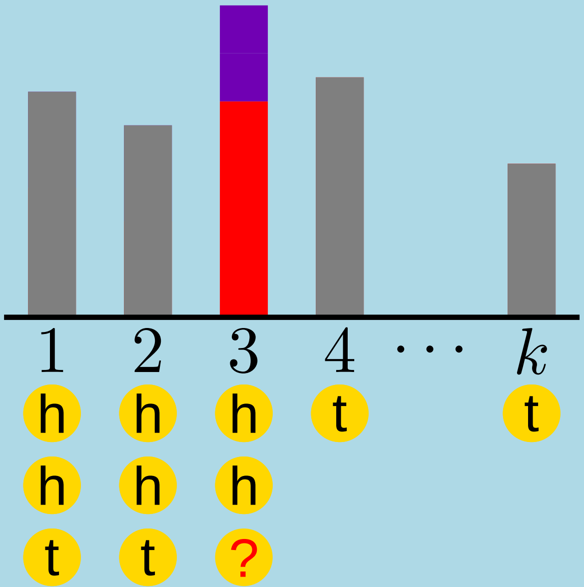


Case 2: j's coin flips tails.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



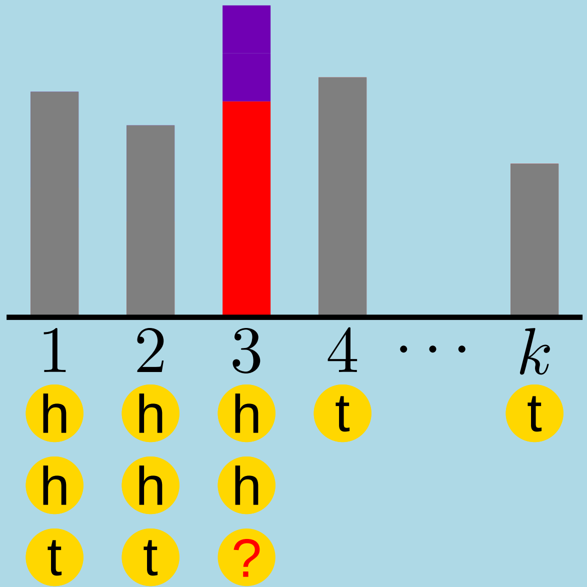
Recall: j's payoff $\leq h$.

Case 2: j's coin flips tails.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



Recall: j's payoff $\leq h$.

$\Rightarrow$ j best before round i.

$\Rightarrow$ algorithm chose j.

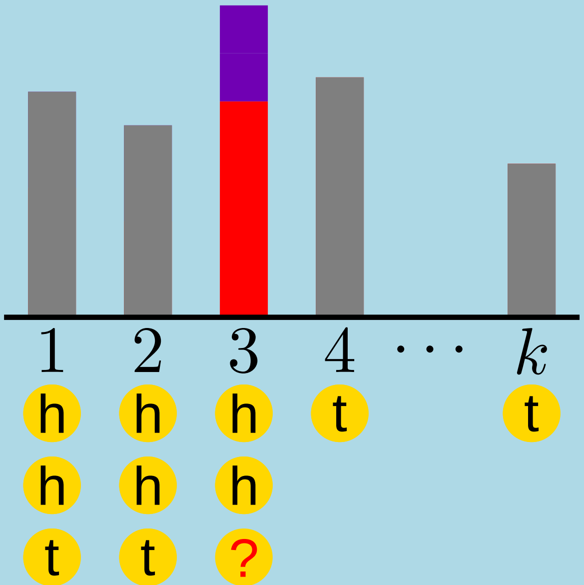
$\Rightarrow$ $P_i = M_i$.

Case 2: j's coin flips tails.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



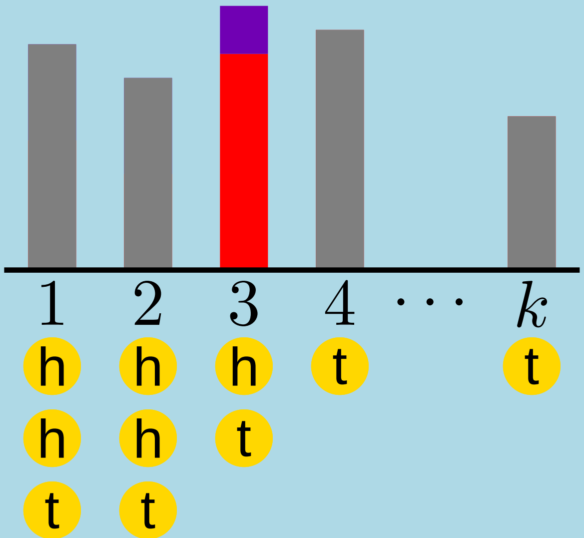
Recall: j's payoff $\leq h$.

$\Rightarrow j$ best before round i.

$\Rightarrow$ algorithm chose j.

$\Rightarrow P_i = M_i$.

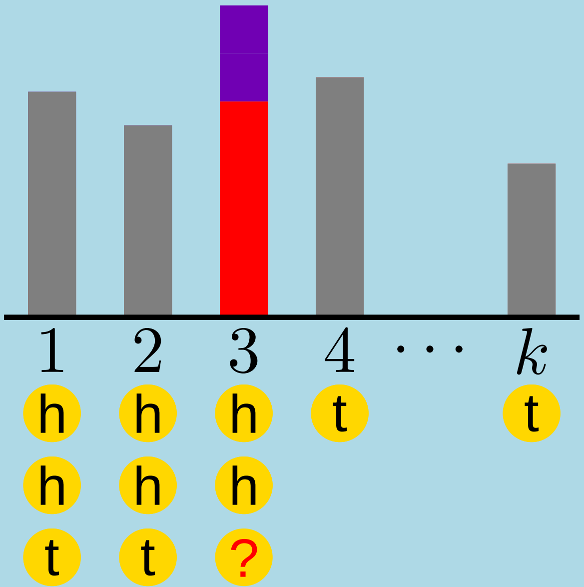
Case 2: j's coin flips tails.



Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



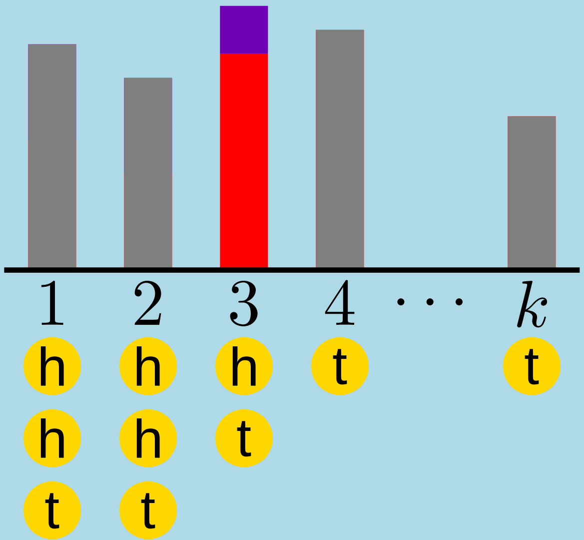
Recall: j's payoff $\leq h$.

$\Rightarrow$ j best before round i.

$\Rightarrow$ algorithm chose j.

$\Rightarrow$ $P_i = M_i$.

Case 2: j's coin flips tails.

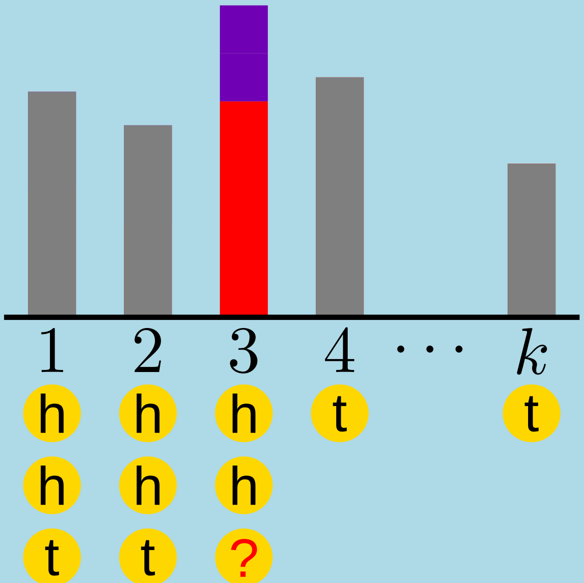


At least: $P_i \geq 0$.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



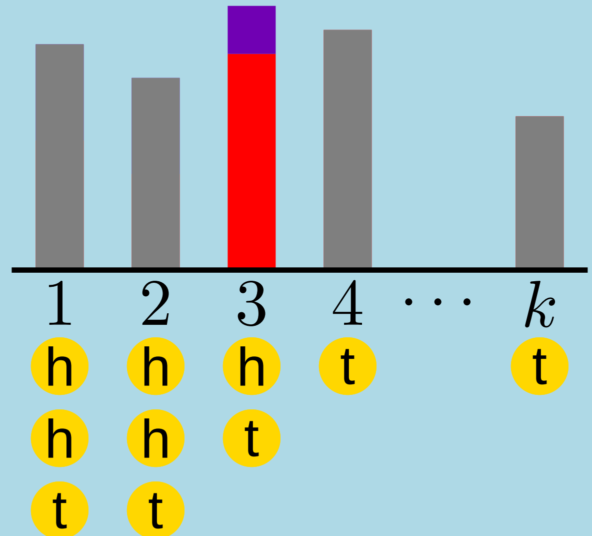
Recall: j's payoff $\leq h$.

$\Rightarrow$ j best before round i.

$\Rightarrow$ algorithm chose j.

$\Rightarrow$ $P_i = M_i$.

Case 2: j's coin flips tails.



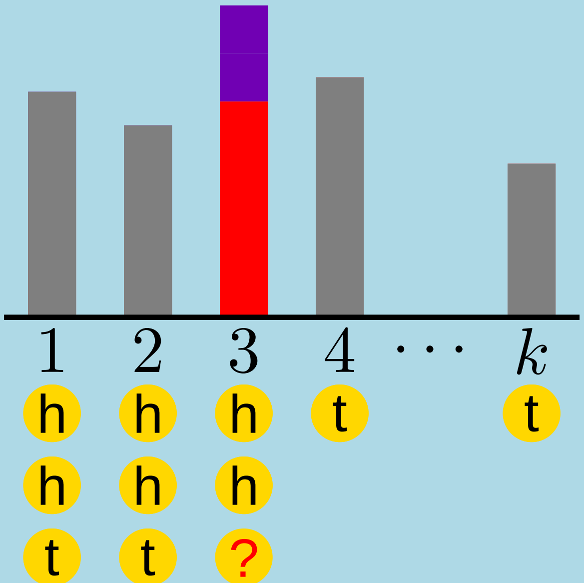
At least: $P_i \geq 0$.

$\forall j,\ \mathbf{E}[P_i \mid j \text{ best}] \geq \frac{1}{2}\mathbf{E}[M_i \mid j \text{ best}]$.

Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



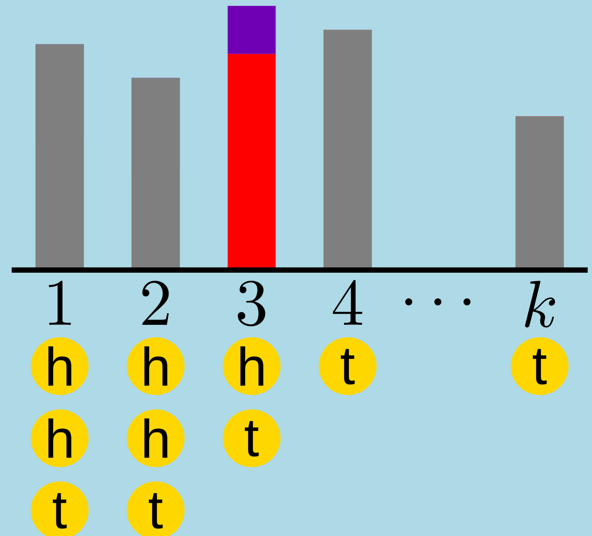
Recall: j's payoff $\leq h$.

$\Rightarrow j$ best before round i.

$\Rightarrow$ algorithm chose j.

$\Rightarrow P_i = M_i$.

Case 2: j's coin flips tails.



At least: $P_i \geq 0$.

$\forall j$, $\mathbf{E}[P_i \mid j \text{ best}] \geq \frac{1}{2}\mathbf{E}[M_i \mid j \text{ best}] \Rightarrow \mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Overview

1. Standard Expert Learning Algorithm (Weighted Majority).

2. Online (Expert) Learning $\Rightarrow$ Online Auction
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$, given $[1, h]$.
   [Blum Kumar Rudra Wu 2003]

3. Kalai's Expert Learning Algorithm & Analysis.

$\Longrightarrow$ 4. Modifications of Kalai's Expert Algorithm for Online Auctions.
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon})$.

5. An Application: Attribute Auctions.

Adaptation to Online Auction

Recall Application: (to online auctions)

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Adaptation to Online Auction

Recall Application: (to online auctions)

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Modification to Kalai:

1. (initialization) For each expert, j, hallucinate initial score, s_j, as

$$p_j \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

Adaptation to Online Auction

Recall Application: (to online auctions)

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$ experts)

Modification to Kalai:

1. (initialization) For each expert, j, hallucinate initial score, s_j, as

   $$p_j \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

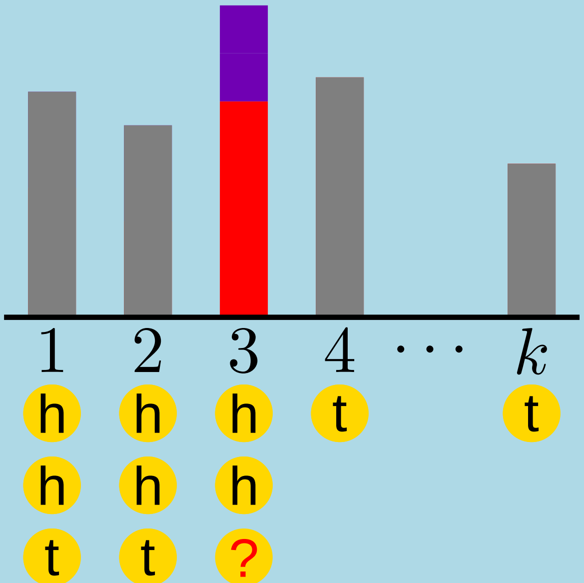
Analysis:

- Lemma still holds.

Recall Proof of Lemma (cont)

Recall: Expert j is best and has coin to flip.

Case 1: j's coin flips heads.



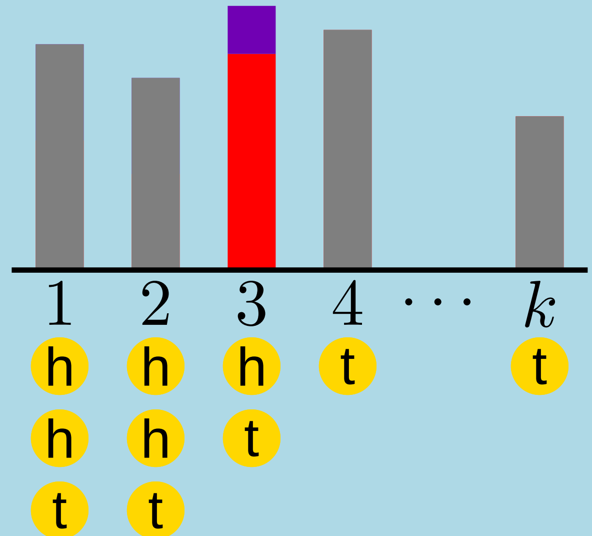
Recall: $\boxed{j\text{'s payoff} \leq 2^j}$

$\Rightarrow$ j best before round i.

$\Rightarrow$ algorithm chose j.

$\Rightarrow$ $P_i = M_i$.

Case 2: j's coin flips tails.



At least: $P_i \geq 0$.

$\forall j,\ \mathbf{E}[P_i \mid j \text{ best}] \geq \frac{1}{2}\mathbf{E}[M_i \mid j \text{ best}] \Rightarrow \mathbf{E}[P_i] \geq \mathbf{E}[M_i]/2$.

Adaptation to Online Auction

Recall Application: (to online auctions)

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$)

Modification to Kalai:

1. (initialization) For each expert, j, halucinate initial score as

$$p_j \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

Analysis:

- Lemma still holds.

Adaptation to Online Auction

Recall Application: (to online auctions)

1. Expert j's strategy: "use sale price p_j".
2. Best expert $\Rightarrow$ best single price profit.
3. For bids in $[1, h]$ use $p_j = 2^j$. ($k = \log h$)

Modification to Kalai:

1. (initialization) For each expert, j, halucinate initial score as

$$p_j \times \text{ number of heads flipped in a row.}$$

2. Run deterministic "go with best expert" algorithm.

Analysis:

- Lemma still holds.
- Theorem improves: $\mathbf{E}[\text{payoff}] \geq \mathrm{OPT}/4 - h$.

Theorem Re-analysis

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}_{\text{expert}}$

Theorem Re-analysis

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2}\sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}_{\text{expert}} \geq \frac{1}{2}\,\mathrm{OPT}$.

Theorem Re-analysis

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}_{\text{expert}} \geq \frac{1}{2}\,\mathrm{OPT}$.
- $\mathbf{E}[H] \leq \sum_j \mathbf{E}[j\text{'s hallucination}] \leq 2h$.

Theorem Re-analysis

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2}\sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2}\left(\mathbf{E}[M] - \mathbf{E}[H]\right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}_{\text{expert}} \geq \frac{1}{2}\,\mathrm{OPT}.$
- $\mathbf{E}[H] \leq \sum_j \mathbf{E}[j\text{'s hallucination}] \leq 2h$.
  - $\mathbf{E}[\text{number of heads in a row}] = 1.$ (geometric distribution)
  - $\mathbf{E}[j\text{'s hallucination}] = p_j \times \mathbf{E}[\text{number of heads}] = p_j = 2^j$.

Theorem Re-analysis

Theorem: $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Proof:

- $\mathbf{E}[P] = \overbrace{\sum_i \mathbf{E}[P_i] \geq \frac{1}{2} \sum_i \mathbf{E}[M_i]}^{\text{from Lemma}} = \frac{1}{2} \left(\mathbf{E}[M] - \mathbf{E}[H] \right)$.
- $\mathbf{E}[M] \geq \mathrm{OPT}_{\text{expert}} \geq \frac{1}{2} \mathrm{OPT}$.
- $\mathbf{E}[H] \leq \sum_j \mathbf{E}[j\text{'s hallucination}] \leq 2h$.
  - $\mathbf{E}[\text{number of heads in a row}] = 1$. (geometric distribution)
  - $\mathbf{E}[j\text{'s hallucination}] = p_j \times \mathbf{E}[\text{number of heads}] = p_j = 2^j$.
- **Result:** $\mathbf{E}[P] \geq \mathrm{OPT}/4 - h$.

Summary & Extensions

1. Near Optimal Online Auction: (up to constant factors)
   - $\mathbf{E}[P] \geq (1-\epsilon)\,\mathrm{OPT} - O(h/\epsilon)$.
   - No need to know bid range in advance.

Summary & Extensions

1. Near Optimal Online Auction: (up to constant factors)
   - **E**$[P] \geq (1-\epsilon)\,\mathrm{OPT} - O(h/\epsilon)$.
     (Prior result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$) [Blum et al. 2003]
   - No need to know bid range in advance.

Summary & Extensions

1. Near Optimal Online Auction: (up to constant factors)
   - $\mathbf{E}[P] \geq (1-\epsilon)\,\mathrm{OPT} - O(h/\epsilon)$.
     (Prior result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$) [Blum et al. 2003]
   - No need to know bid range in advance.
2. Posted Price Online "Auctions":

Summary & Extensions

1. Near Optimal Online Auction: (up to constant factors)
   - **E**$[P] \geq (1-\epsilon)\,\mathrm{OPT} - O(h/\epsilon)$.
     (Prior result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$) [Blum et al. 2003]
   - No need to know bid range in advance.
2. Posted Price Online "Auctions":
   - Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$

- **Open:** Constant additive term?

Summary & Extensions

1. Near Optimal Online Auction: (up to constant factors)
   - $\mathbf{E}[P] \geq (1-\epsilon)\,\mathrm{OPT} - O(h/\epsilon)$.
     (Prior result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$) [Blum et al. 2003]
   - No need to know bid range in advance.
2. Posted Price Online "Auctions":
   - Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$
     (Prior result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log h \log\log h)$)
     [Blum et al. 2003]
   - **Open:** Constant additive term?

Overview

1. Standard Expert Learning Algorithm (Weighted Majority).

2. Online (Expert) Learning $\Rightarrow$ Online Auction
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon}\log\log h)$, given $[1, h]$.
   [Blum Kumar Rudra Wu 2003]

3. Kalai's Expert Learning Algorithm & Analysis.

4. Modifications of Kalai's Expert Algorithm for Online Auctions.
   Result: $(1-\epsilon)\,\mathrm{OPT} - O(\frac{h}{\epsilon})$.

$\Longrightarrow$ 5. An Application: Attribute Auctions.

Attribute Auctions

The *Unlimited Supply (Offline) Auction Problem*:

Given:

- n identical items for sale.
- n indistinguishable bidders.

Design: Auction with profit near $\mathrm{OPT} =$ optimal single price sale.

Solution: E.g., [Myersion 1981, Goldberg Hartline Wright 2001]

Attribute Auctions

The *Unlimited Supply (Offline) Auction Problem*:

Given:

- n identical items for sale.
- n indistinguishable bidders.

Design: Auction with profit near $\mathrm{OPT} =$ optimal single price sale.

Solution: E.g., [Myersion 1981, Goldberg Hartline Wright 2001]

What if bidders are distinguishible?

Attribute Auctions

The *Attribute Auction Problem*:

Given:

- n identical items for sale.
- n bidders, bidder i with attribute a_i.

Design: Auction with maximal profit. (use attribute to segment market)

Attribute Auctions

The *Attribute Auction Problem*:

Given:

- n identical items for sale.
- n bidders, bidder i with attribute a_i.

Design: Auction with maximal profit. (use attribute to segment market)

Truthful mechanism design: For bidder i offer price

$$p_i = f\begin{pmatrix} a_1, & \dots, & a_{i-1}, & a_i, & a_{i+1}, & \dots, & a_n \\ b_1, & \dots, & b_{i-1}, & ?, & b_{i+1}, & \dots, & b_n \end{pmatrix}$$

Special Case: $a_i \in \mathbb{R}$

Attribute Auction, AA:

1. Order bids by attribute.
2. Simulate online auction.
3. Reset simulation whenever $\mathrm{OPT} > \gamma h$.

Special Case: $a_i \in \mathbb{R}$

Attribute Auction, AA:

1. Order bids by attribute.
2. Simulate online auction.
3. Reset simulation whenever $\mathrm{OPT} > \gamma h$.

Result: performance comparable to approximately optimal auctions on optimal market segmentations.

Special Case: $a_i \in \mathbb{R}$

Attribute Auction, AA:

1. Order bids by attribute.
2. Simulate online auction.
3. Reset simulation whenever $\mathrm{OPT} > \gamma h$.

Result: performance comparable to approximately optimal auctions on optimal market segmentations.

Open: Multidimensional Attribute Auctions?

Open: Structured Attribute Auctions?