Today

Experts/Multiplicative Weights.
Experts/Multiplicative Weights.
Experts/Zero-Sum Games Equilibrium.
Today

Experts/Multiplicative Weights.
Experts/Zero-Sum Games Equilibrium.
Boosting and Experts.
Experts/Multiplicative Weights.
Experts/Zero-Sum Games Equilibrium.
Boosting and Experts.
Routing and Experts.
The multiplicative weights framework.
Experts framework.

$n$ experts.
Experts framework.

$n$ experts.

Every day, each offers a prediction.
Experts framework.

$n$ experts.
Every day, each offers a prediction.
“Rain” or “Shine.”
Experts framework.

\( n \) experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>\cdots</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td></td>
<td></td>
<td>\cdots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td></td>
<td>\cdots</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
</tbody>
</table>

Whose advice do you follow?

“The one who is correct most often.”

Sort of.

How well do you do?
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>...</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Rained!
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>...</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Rained! Shined!
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>…</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td>…</td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>Shine</td>
<td>…</td>
</tr>
</tbody>
</table>

Rained! Shined! Shined!
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>...</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>Shined</td>
</tr>
</tbody>
</table>

Rained! Shined! Shined! ...
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>...</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>Shine</td>
<td>...</td>
</tr>
</tbody>
</table>

Rained! Shined! Shined! ...

Whose advice do you follow?
Experts framework.

$n$ experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>…</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>…</td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>

Rained! Shined! Shined! …

Whose advice do you follow?

“The one who is correct most often.”
Experts framework.

\( n \) experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>⋯</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>⋯</td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>⋯</td>
<td></td>
</tr>
<tr>
<td></td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td></td>
<td>Shine</td>
</tr>
<tr>
<td></td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td></td>
<td>⋯</td>
</tr>
</tbody>
</table>

Rained! Shined! Shined! ⋯

Whose advice do you follow?

“The one who is correct most often.”

Sort of.
Experts framework.

\( n \) experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>…</th>
<th>Day T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>Shine</td>
<td>Rain</td>
<td>Shine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expert 2</td>
<td>Shine</td>
<td>Shine</td>
<td>Shine</td>
<td>…</td>
<td></td>
</tr>
<tr>
<td>Expert 3</td>
<td>Rain</td>
<td>Rain</td>
<td>Rain</td>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>

|        |       |       |       | Shine | …    |

Rained! Shined! Shined! …

Whose advice do you follow?

“The one who is correct most often.”

Sort of.

How well do you do?
Infallible expert.

One of the experts is infallible!
Infallible expert.

One of the experts is infallible!

Your strategy?
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never!
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make?
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? Mistake Bound.
Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? Mistake Bound.

(A) 1
(B) 2
(C) \( \log n \)
(D) \( n - 1 \)

Adversary designs setup to watch who you choose, and make that expert make a mistake.
Infallible expert.

One of the experts is infallible!

Your strategy?
Choose any expert that has not made a mistake!

How long to find perfect expert?
Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? **Mistake Bound.**

(A) 1
(B) 2
(C) $\log n$
(D) $n - 1$

Adversary designs setup to watch who you choose, and make that expert make a mistake.

$n - 1!$
Concept Alert.

Note.

You could have done so well... but you didn't! ha.. ha... ha.

Technical Term: Regret.

Analysis of Algorithms: do as well as possible! Minimize Regret $\equiv$ Loss.
Concept Alert.

Note.

Adversary:

"You could have done so well... but you didn't! ha... ha... ha.

Technical Term: Regret.
Analysis of Algorithms: do as well as possible!
Minimize Regret $\equiv$ Loss.
Concept Alert.

Note.

Adversary: makes you want to look bad.
Concept Alert.

Note.

Adversary: makes you want to look bad. "You could have done so well..."
Note.

Adversary:
  makes you want to look bad.
  "You could have done so well...
  but you didn’t!"
Note.

Adversary:
makes you want to look bad.
"You could have done so well...
but you didn’t! ha..
Concept Alert.

Note.

Adversary:
  makes you want to look bad.
  ”You could have done so well...
  but you didn’t! ha..ha...
Note.

Adversary:
    makes you want to look bad.
"You could have done so well...
    but you didn’t! ha..ha... ha.
Concept Alert.

Note.

Adversary:
  makes you want to look bad.
  "You could have done so well...
  but you didn’t! ha..ha... ha.

Technical Term: Regret.
Note.

Adversary:
    makes you want to look bad.
    "You could have done so well...
    but you didn’t! ha..ha... ha.

Technical Term: Regret.

Analysis of Algorithms: do as well as possible!
Note.

Adversary:
    makes you want to look bad.
    "You could have done so well...
    but you didn’t! ha..ha... ha.

Technical Term: Regret.

Analysis of Algorithms: do as well as possible!
    Minimize Regret
Note.

Adversary:
makes you want to look bad.
"You could have done so well...
but you didn’t! ha..ha... ha.

Technical Term: Regret.

Analysis of Algorithms: do as well as possible!
Minimize Regret $\equiv$ Loss.
Back to mistake bound.

Infallible Experts.
Back to mistake bound.

Infallible Experts.

Alg: Choose one of the perfect experts.
Back to mistake bound.

Infallible Experts.

Alg: Choose one of the perfect experts.

**Mistake Bound:** $n - 1$
Infallible Experts.

Alg: Choose one of the perfect experts.

**Mistake Bound:** $n - 1$
- Lower bound: adversary argument.
Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n - 1$
- Lower bound: adversary argument.
- Upper bound:
Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n - 1$
- Lower bound: adversary argument.
- Upper bound: every mistake finds fallible expert.
Back to mistake bound.

Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: \( n - 1 \)
  - Lower bound: adversary argument.
  - Upper bound: every mistake finds fallible expert.

Better Algorithm?
Infallible Experts.

Alg: Choose one of the perfect experts.

**Mistake Bound:** $n - 1$
- Lower bound: adversary argument.
- Upper bound: every mistake finds fallible expert.

Better Algorithm?

Making decision, not trying to find expert!
Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n - 1$

  Lower bound: adversary argument.
   Upper bound: every mistake finds fallible expert.

Better Algorithm?

Making decision, not trying to find expert!

Algorithm: Go with the majority of previously correct experts.
Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n - 1$
- Lower bound: adversary argument.
- Upper bound: every mistake finds fallible expert.

Better Algorithm?

Making decision, not trying to find expert!

Algorithm: Go with the majority of previously correct experts.

What you would do anyway!
Alg 2: find majority of the perfect

How many mistakes could you make?
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) log \( n \)
(D) \( n - 1 \)
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) \log n
(D) n - 1

At most \log n!
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) $\log n$
(D) $n - 1$

At most $\log n$!

When alg makes a mistake, $|\text{“perfect” experts}|$ drops by a factor of two.
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1

(B) 2

(C) $\log n$

(D) $n - 1$

At most $\log n!$

When alg makes a mistake,
   $|\text{“perfect” experts}|$ drops by a factor of two.

Initially $n$ perfect experts
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) \( \log n \)
(D) \( n - 1 \)

At most \( \log n \)!

When alg makes a mistake,

\[ |\text{"perfect" experts}| \text{ drops by a factor of two.} \]

Initially \( n \) perfect experts

mistake \( \rightarrow \) \( \leq n/2 \) perfect experts
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) $\log n$
(D) $n - 1$

At most $\log n$!

When alg makes a mistake,

$|\text{"perfect" experts}|$ drops by a factor of two.

Initially $n$ perfect experts

mistake $\rightarrow \leq n/2$ perfect experts
mistake $\rightarrow \leq n/4$ perfect experts
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1

(B) 2

(C) $\log n$

(D) $n - 1$

At most $\log n!$

When alg makes a mistake,

$|\text{“perfect” experts}|$ drops by a factor of two.

Initially $n$ perfect experts

mistake $\rightarrow \leq \frac{n}{2}$ perfect experts

mistake $\rightarrow \leq \frac{n}{4}$ perfect experts

\vdots
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) $\log n$
(D) $n - 1$

At most $\log n$!

When alg makes a mistake, $|\text{“perfect” experts}|$ drops by a factor of two.

Initially $n$ perfect experts
  mistake $\rightarrow \leq n/2$ perfect experts
  mistake $\rightarrow \leq n/4$ perfect experts
  :
  mistake $\rightarrow \leq 1$ perfect expert
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) $\log n$
(D) $n - 1$

At most $\log n$!

When alg makes a mistake, \mid “perfect” experts\mid drops by a factor of two.

Initially $n$ perfect experts
\begin{align*}
\text{mistake} & \rightarrow \leq \frac{n}{2} \text{ perfect experts} \\
\text{mistake} & \rightarrow \leq \frac{n}{4} \text{ perfect experts} \\
\vdots \\
\text{mistake} & \rightarrow \leq 1 \text{ perfect expert}
\end{align*}
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) \(\log n\)
(D) \(n - 1\)

At most \(\log n!\)

When alg makes a mistake,
|“perfect” experts| drops by a factor of two.

Initially \(n\) perfect experts
mistake \(\rightarrow\) \(\leq \frac{n}{2}\) perfect experts
mistake \(\rightarrow\) \(\leq \frac{n}{4}\) perfect experts

...\\

mistake \(\rightarrow\) \(\leq 1\) perfect expert

\(\geq 1\) perfect expert
Alg 2: find majority of the perfect

How many mistakes could you make?

(A) 1
(B) 2
(C) \( \log n \)
(D) \( n - 1 \)

At most \( \log n \)!

When alg makes a mistake,

| “perfect” experts | drops by a factor of two. |

Initially \( n \) perfect experts

mistake \( \rightarrow \) \( \leq n/2 \) perfect experts
mistake \( \rightarrow \) \( \leq n/4 \) perfect experts

\vdots

mistake \( \rightarrow \) \( \leq 1 \) perfect expert

\( \geq 1 \) perfect expert \( \rightarrow \) at most \( \log n \) mistakes!
Imperfect Experts

Goal?
Imperfect Experts

Goal?
Do as well as the best expert!
Imperfect Experts

Goal?
Do as well as the best expert!
Algorithm.
Imperfect Experts

Goal?
Do as well as the best expert!
Algorithm. Suggestions?
Imperfect Experts

Goal?
Do as well as the best expert!
Algorithm. Suggestions?
Go with majority?
Imperfect Experts

Goal?
Do as well as the best expert!

Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Imperfect Experts

Goal?
Do as well as the best expert!
Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Best expert is penalized the least.
Imperfect Experts

Goal?
Do as well as the best expert!

Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Best expert is penalized the least.

1. Initially: $w_i = 1$. 
Imperfect Experts

Goal?
Do as well as the best expert!

Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Best expert is penalized the least.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
Imperfect Experts

Goal?
Do as well as the best expert!
Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Best expert is penalized the least.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Imperfect Experts

Goal?
Do as well as the best expert!

Algorithm. Suggestions?
Go with majority?
Penalize inaccurate experts?
Best expert is penalized the least.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i / 2$ if wrong.

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$.

Initially $n$. For best expert, $b$, $w_b \geq 1 / 2 m$.

Each mistake: total weight of incorrect experts reduced by $-1 / 2$ factor of $1 / 2$?
Each incorrect expert weight multiplied by $1 / 2$!
Total weight decreases by factor of $1 / 2$? factor of $3 / 4$?

Mistake $\rightarrow \geq$ half weight with incorrect experts (\geq $1 / 2$ total).
Mistake $\rightarrow$ potential function decreased by $3 / 4$.

We have $1 / 2 m \leq \sum_i w_i \leq (3 / 4)^n M$.

where $M$ is number of algorithm mistakes.
Analysis: weighted majority

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.

Potential function: $\sum_i w_i$.

Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2} m$.

Each mistake: total weight of incorrect experts reduced by $-\frac{1}{2} w_i$.

Total weight decreases by factor of $\frac{3}{4}$.

Mistake $\rightarrow \geq \frac{1}{2} w_i$ (total).

Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$.

We have $\frac{1}{2} m \leq \sum_i w_i \leq (\frac{3}{4}) M n$.

where $M$ is number of algorithm mistakes.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function:

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.

Mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq \frac{1}{2}$ total).

Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$.

We have $\frac{1}{2}m \leq \sum_i w_i \leq \left(\frac{3}{4}\right) M n$, where $M$ is number of algorithm mistakes.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.
Potential function: $\sum_i w_i$.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i / 2$ if wrong.

For best expert, $b$, $w_b \geq \frac{1}{2}m$.
Each mistake: total weight of incorrect experts reduced by $-\frac{1}{2}\frac{1}{2}$ factor of $\frac{1}{2}$ each incorrect expert weight multiplied by $\frac{1}{2}$ total weight decreases by factor of $\frac{3}{4}$ mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq \frac{1}{2}$ total).
Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$. We have $\frac{1}{2}m \leq \sum_i w_i \leq (\frac{3}{4})^m M n$. Where $M$ is number of algorithm mistakes.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.
Potential function: $\sum_i w_i$. Initially $n$.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.

For best expert, $b$, $w_{b} \geq \frac{1}{2}m$.
Each mistake: total weight of incorrect experts reduced by $\frac{-1}{2}$ factor of $\frac{1}{2}$?
Each incorrect expert weight multiplied by $\frac{1}{2}$! total weight decreases by $\frac{3}{4}$?
Mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq \frac{1}{2}$ total).
Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$.
We have $\frac{1}{2}m \leq \sum_i w_i \leq (\frac{3}{4})^M n$.

where $M$ is number of algorithm mistakes.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

1. Initially: $w_i = 1$.

2. Predict with weighted majority of experts.

3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:
- total weight of incorrect experts reduced by

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow \frac{w_i}{2}$ if wrong.
Analysis: weighted majority

**Goal:** Best expert makes $m$ mistakes.

**Potential function:** $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:
- total weight of incorrect experts reduced by $-1$?

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes \( m \) mistakes.

Potential function: \( \sum_i w_i \). Initially \( n \).

For best expert, \( b \), \( w_b \geq \frac{1}{2^m} \).

Each mistake:
- total weight of incorrect experts reduced by \(-1\) or \(-2\)?

1. Initially: \( w_i = 1 \).
2. Predict with weighted majority of experts.
3. \( w_i \rightarrow w_i/2 \) if wrong.
Analysis: weighted majority

Goal: Best expert makes \( m \) mistakes.

Potential function: \( \sum_i w_i \). Initially \( n \).

For best expert, \( b \), \( w_b \geq \frac{1}{2^m} \).

Each mistake:
- total weight of incorrect experts reduced by \(-1?\) \(-2?\) factor of \( \frac{1}{2} \)?

1. Initially: \( w_i = 1 \).
2. Predict with weighted majority of experts.
3. \( w_i \rightarrow w_i/2 \) if wrong.
Analysis: weighted majority

Goal: Best expert makes \( m \) mistakes.

Potential function: \( \sum_i w_i \). Initially \( n \).

For best expert, \( b \), \( w_b \geq \frac{1}{2^m} \).

Each mistake:
- total weight of incorrect experts reduced by \(-2\) factor of \( \frac{1}{2} \)?
- each incorrect expert weight multiplied by \( \frac{1}{2} \)

1. Initially: \( w_i = 1 \).
2. Predict with weighted majority of experts.
3. \( w_i \rightarrow w_i/2 \) if wrong.
Analysis: weighted majority

Goal: Best expert makes \( m \) mistakes.

Potential function: \( \sum_i w_i \). Initially \( n \).

For best expert, \( b \), \( w_b \geq \frac{1}{2^m} \).

Each mistake:
- total weight of incorrect experts reduced by \( -1 \) if wrong.
- factor of \( \frac{1}{2} \)!
- each incorrect expert weight multiplied by \( \frac{1}{2} \)
- total weight decreases by \( \frac{1}{2} \)!

1. Initially: \( w_i = 1 \).
2. Predict with weighted majority of experts.
3. \( w_i \rightarrow \frac{w_i}{2} \) if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.
Potential function: $\sum_i w_i$. Initially $n$.
For best expert, $b$, $w_b \geq \frac{1}{2^m}$.
Each mistake:
- total weight of incorrect experts reduced by $\frac{1}{2}$?
  -2? factor of $\frac{1}{2}$?
- each incorrect expert weight multiplied by $\frac{1}{2}$!
- total weight decreases by factor of $\frac{1}{2}$? factor of $\frac{3}{4}$?

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:
- total weight of incorrect experts reduced by $-1/2$?
- $-2/2$? factor of $1/2$?
- each incorrect expert weight multiplied by $1/2$!
- total weight decreases by factor of $1/2$? factor of $3/4$?
- mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq 1/2$ total.

1. Initially: $w_i = 1$.

2. Predict with weighted majority of experts.

3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:
- total weight of incorrect experts reduced by $-1$? $-2$? factor of $\frac{1}{2}$? each incorrect expert weight multiplied by $\frac{1}{2}$!
- total weight decreases by factor of $\frac{1}{2}$? factor of $\frac{3}{4}$?

Mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq \frac{1}{2}$ total).

Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$.

1. Initially: $w_i = 1$.
2. Predict with weighted majority of experts.
3. $w_i \rightarrow w_i/2$ if wrong.
Analysis: weighted majority

Goal: Best expert makes $m$ mistakes.

Potential function: $\sum_i w_i$. Initially $n$.

For best expert, $b$, $w_b \geq \frac{1}{2^m}$.

Each mistake:
- total weight of incorrect experts reduced by $-1$? $-2$? factor of $\frac{1}{2}$?
- each incorrect expert weight multiplied by $\frac{1}{2}$!
- total weight decreases by factor of $\frac{1}{2}$? factor of $\frac{3}{4}$?
- mistake $\rightarrow \geq$ half weight with incorrect experts ($\geq \frac{1}{2}$ total).

Mistake $\rightarrow$ potential function decreased by $\frac{3}{4}$.

We have
\[
\frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n.
\]
where $M$ is number of algorithm mistakes.
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]
Analysis: continued.

\[
\frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n.
\]

\(m\) - best expert mistakes
\[
\frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n.
\]

\(m\) - best expert mistakes  \(M\) algorithm mistakes.
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n. \]

\( m \) - best expert mistakes \( M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left(\frac{3}{4}\right)^M n. \]
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n. \]

\( m \) - best expert mistakes \quad \( M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left(\frac{3}{4}\right)^M n. \]

Take log of both sides.
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n. \]

\( m \) - best expert mistakes  \( M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left(\frac{3}{4}\right)^M n. \]

Take log of both sides.

\[ -m \leq -M \log(4/3) + \log n. \]
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]

\( m \) - best expert mistakes \( \ M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left( \frac{3}{4} \right)^M n. \]

Take log of both sides.

\[ -m \leq -M \log(4/3) + \log n. \]

Solve for \( M \).

\[ M \leq \frac{(m + \log n)}{\log(4/3)} \]
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]

- \( m \) - best expert mistakes
- \( M \) - algorithm mistakes.

\[ \frac{1}{2^m} \leq \left( \frac{3}{4} \right)^M n. \]

Take log of both sides.

\[-m \leq -M \log(4/3) + \log n.\]

Solve for \( M \).

\[ M \leq (m + \log n) / \log(4/3) \leq 2.4(m + \log n) \]
Analysis: continued.

\[
\frac{1}{2^m} \leq \sum_i w_i \leq \left(\frac{3}{4}\right)^M n.
\]

\(m\) - best expert mistakes \(M\) algorithm mistakes.

\[
\frac{1}{2^m} \leq \left(\frac{3}{4}\right)^M n.
\]

Take log of both sides.

\[-m \leq -M \log(4/3) + \log n.\]

Solve for \(M\).

\[M \leq (m + \log n) / \log(4/3) \leq 2.4(m + \log n)\]

Multiply by \(1 - \varepsilon\) for incorrect experts...
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]

\( m \) - best expert mistakes  \( M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left( \frac{3}{4} \right)^M n. \]

Take log of both sides.

\[ -m \leq -M \log(\frac{4}{3}) + \log n. \]

Solve for \( M \).

\[ M \leq \frac{(m + \log n)}{\log(4/3)} \leq 2.4(m + \log n) \]

Multiply by \( 1 - \varepsilon \) for incorrect experts...

\[ (1 - \varepsilon)^m \leq \left(1 - \frac{\varepsilon}{2}\right)^M n. \]
Analysis: continued.

\[
\frac{1}{2^m} \leq \sum_i w_i \leq (\frac{3}{4})^M n.
\]

\(m\) - best expert mistakes \(M\) algorithm mistakes.

\[
\frac{1}{2^m} \leq (\frac{3}{4})^M n.
\]

Take log of both sides.

\[-m \leq -M \log(4/3) + \log n.\]

Solve for \(M\).

\[M \leq (m + \log n) / \log(4/3) \leq 2.4(m + \log n)\]

Multiply by \(1 - \varepsilon\) for incorrect experts...

\[(1 - \varepsilon)^m \leq (1 - \frac{\varepsilon}{2})^M n.\]

Massage...
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]

- \( m \) - best expert mistakes
- \( M \) - algorithm mistakes.

\[ \frac{1}{2^m} \leq \left( \frac{3}{4} \right)^M n. \]

Take log of both sides.

\[ -m \leq -M \log(4/3) + \log n. \]

Solve for \( M \).

\[ M \leq \frac{m + \log n}{\log(4/3)} \leq 2.4(m + \log n) \]

Multiply by \( 1 - \varepsilon \) for incorrect experts...

\[ (1 - \varepsilon)^m \leq \left( 1 - \frac{\varepsilon}{2} \right)^M n. \]

Massage...

\[ M \leq 2(1 + \varepsilon)m + \frac{2\ln n}{\varepsilon} \]
Analysis: continued.

\[ \frac{1}{2^m} \leq \sum_i w_i \leq \left( \frac{3}{4} \right)^M n. \]

\( m \) - best expert mistakes \quad \( M \) algorithm mistakes.

\[ \frac{1}{2^m} \leq \left( \frac{3}{4} \right)^M n. \]

Take log of both sides.

\[-m \leq -M \log(4/3) + \log n.\]

Solve for \( M \).

\[ M \leq (m + \log n) / \log(4/3) \leq 2.4(m + \log n) \]

Multiply by \( 1 - \varepsilon \) for incorrect experts...

\[ (1 - \varepsilon)^m \leq \left( 1 - \frac{\varepsilon}{2} \right)^M n. \]

Massage...

\[ M \leq 2(1 + \varepsilon)m + \frac{2\ln n}{\varepsilon} \]

Approaches a factor of two of best expert performance!
Best Analysis?

Consider two experts: A, B

Bad example?
Which is worse?

(A) A correct even days, B correct odd days
(B) A correct first half of days, B correct second

Best expert performance: \(\frac{T}{2}\) mistakes.

Pattern (A): \(T - 1\) mistakes.

Factor of (almost) two worse!
Best Analysis?

Consider two experts: A, B
Best Analysis?

Consider two experts: A, B

Bad example?
Best Analysis?

Consider two experts: A, B

Bad example?

Which is worse?

(A) A correct even days, B correct odd days

(B) A correct first half of days, B correct second
Best Analysis?

Consider two experts: A,B
Bad example?
Which is worse?
(A) A correct even days, B correct odd days
(B) A correct first half of days, B correct second

Best expert performance: \( T/2 \) mistakes.
Consider two experts: A, B

Bad example?

Which is worse?

(A) A correct even days, B correct odd days

(B) A correct first half of days, B correct second

Best expert performance: $T/2$ mistakes.

Pattern (A): $T - 1$ mistakes.
Best Analysis?

Consider two experts: A,B

Bad example?

Which is worse?

(A) A correct even days, B correct odd days
(B) A correct first half of days, B correct second

Best expert performance: \(T/2\) mistakes.

Pattern (A): \(T - 1\) mistakes.

Factor of (almost) two worse!
Randomization

Better approach?
Randomization

Better approach?
Use?
Randomization!!!!

Better approach?
Use?
   Randomization!
Randomization!!!!

Better approach?
Use?
    Randomization!
That is, choose expert $i$ with prob $\propto w_i$.
Better approach?
Use?
  Randomization!
That is, choose expert $i$ with prob $\propto w_i$
Bad example: A,B,A,B,A...
Better approach? Use? Randomization!
That is, choose expert $i$ with prob $\propto w_i$
Bad example: A,B,A,B,A...
After a bit, A and B make nearly the same number of mistakes.
Better approach?
Use?

Randomization!
That is, choose expert $i$ with prob $\propto w_i$
Bad example: A,B,A,B,A...
After a bit, A and B make nearly the same number of mistakes.
Choose each with approximately the same probability.
Better approach?
Use?

Randomization!

That is, choose expert $i$ with prob $\propto w_i$

Bad example: A,B,A,B,A...

After a bit, A and B make nearly the same number of mistakes.
Choose each with approximately the same probability.
Make a mistake around 1/2 of the time.
Better approach?
Use?
  Randomization!
That is, choose expert \( i \) with prob \( \propto w_i \)
Bad example: A,B,A,B,A...
After a bit, A and B make nearly the same number of mistakes.
Choose each with approximately the same probability.
Make a mistake around \( 1/2 \) of the time.
Best expert makes \( T/2 \) mistakes.
Randomization!!!!

Better approach?
Use?

Randomization!
That is, choose expert $i$ with prob $\propto w_i$
Bad example: A,B,A,B,A...
After a bit, A and B make nearly the same number of mistakes.
Choose each with approximately the same probability.
Make a mistake around \(1/2\) of the time.
Best expert makes \(T/2\) mistakes.
Roughly
Better approach?
Use?

Randomization!
That is, choose expert \(i\) with prob \(\propto w_i\)
Bad example: A,B,A,B,A...
After a bit, A and B make nearly the same number of mistakes.
Choose each with approximately the same probability.
Make a mistake around \(1/2\) of the time.
Best expert makes \(T/2\) mistakes.
Roughly optimal!
Randomized analysis.

Some formulas:
Randomized analysis.

Some formulas:
For $\varepsilon \leq 1, x \in [0, 1]$,
Randomized analysis.

Some formulas:

For $\varepsilon \leq 1, x \in [0, 1],$

\[(1 + \varepsilon)^x \leq (1 + \varepsilon x)\]

\[(1 - \varepsilon)^x \leq (1 - \varepsilon x)\]
Randomized analysis.

Some formulas:

For $\varepsilon \leq 1, x \in [0, 1]$,

$(1 + \varepsilon)^x \leq (1 + \varepsilon x)$
$(1 - \varepsilon)^x \leq (1 - \varepsilon x)$

For $\varepsilon \in [0, \frac{1}{2}]$, 

$-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon$
Randomized analysis.

Some formulas:

For $\varepsilon \leq 1, x \in [0, 1],$

$$ (1 + \varepsilon)^x \leq (1 + \varepsilon x) $$
$$ (1 - \varepsilon)^x \leq (1 - \varepsilon x) $$

For $\varepsilon \in [0, \frac{1}{2}],$

$$ -\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon $$
$$ \varepsilon - \varepsilon^2 \leq \ln(1 + \varepsilon) \leq \varepsilon $$
Randomized analysis.

Some formulas:

For \( \varepsilon \leq 1, x \in [0, 1] \),

\[
(1 + \varepsilon)^x \leq (1 + \varepsilon x)
\]

\[
(1 - \varepsilon)^x \leq (1 - \varepsilon x)
\]

For \( \varepsilon \in [0, \frac{1}{2}] \),

\[
-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon
\]

\[
\varepsilon - \varepsilon^2 \leq \ln(1 + \varepsilon) \leq \varepsilon
\]

Proof Idea: \( \ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots \)
Randomized algorithm
Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$. 

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $w_i/W$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \epsilon) \ell_i^t W(t)$

Best expert, $b$, loses $L^*$. 

$W(0) = n$ 

$W(t) \geq w_b \geq (1 - \epsilon)L^*$. 

$L_t = \sum_i w_i \ell_i^t$ expected loss of alg. in time $t$. 

Claim: 

$W(t+1) \leq W(t)(1 - \epsilon L_t)$. 

Loss $\rightarrow$ weight loss. 

Proof: 

$W(t+1) = \sum_i (1 - \epsilon) \ell_i^t w_i \leq \sum_i (1 - \epsilon) \ell_i^t w_i = \sum_i w_i - \epsilon \sum_i \ell_i^t w_i = \sum_i w_i(1 - \epsilon \sum_i \ell_i^t) = W(t)(1 - \epsilon L_t)$.
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$. 
Randomized algorithm

Expert $i$ loses $\ell^t_i \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$. 
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon)^{\ell_i^t}$
Randomized algorithm

Expert $i$ loses $\ell^t_i \in [0,1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)^{\ell^t_i}$

$W(t)$ sum of $w_i$ at time $t$. 
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)^\ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) =$
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon) \ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0,1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon)\ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total.
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)^{\ell_i^t}$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\implies W(T) \geq w_b \geq (1 - \varepsilon)^{L^*}$. 

Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \epsilon)^{\ell_i^t}$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \epsilon)^{L^*}$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$. 
**Randomized algorithm**

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon) \ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon) L^*$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t+1) \leq W(t)(1 - \varepsilon L_t)$
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)^{\ell_i^t}$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon) L^*$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon)\ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)L^*$.

$L_t = \sum_i \frac{w_i\ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ \textbf{Loss $\rightarrow$ weight loss.}

Proof:
Randomized algorithm

Expert $i$ loses $\ell_t^i \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon)^{\ell_t^i}$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)^{L^*}$.

$L_t = \sum_i \frac{w_i \ell_t^i}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:
$W(t + 1) = \sum_i (1 - \varepsilon)^{\ell_t^i} w_i$
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon) \ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)^{L^*}$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

$W(t + 1) = \sum_i (1 - \varepsilon)^{\ell_i^t} w_i \leq \sum_i (1 - \varepsilon \ell_i^t) w_i$
Randomized algorithm

Expert $i$ loses $\ell^t_i \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)^{\ell^t_i}$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)^{L^*}$.

$L_t = \sum_i \frac{w_i \ell^t_i}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

$W(t + 1) = \sum_i (1 - \varepsilon)^{\ell^t_i}w_i \leq \sum_i (1 - \varepsilon \ell^t_i)w_i = \sum_i w_i - \varepsilon \sum_i w_i \ell^t_i$
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i (1 - \varepsilon)\ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)L^*$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t+1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

\[
W(t+1) = \sum_i (1 - \varepsilon)\ell_i^t w_i \leq \sum_i (1 - \varepsilon \ell_i^t) w_i = \sum_i w_i - \varepsilon \sum_i w_i \ell_i^t = \sum_i w_i \left( 1 - \varepsilon \frac{\sum_i w_i \ell_i^t}{\sum_i w_i} \right)
\]
Randomized algorithm

Expert $i$ loses $\ell_i^t \in [0, 1]$ in round $t$.

1. Initially $w_i = 1$ for expert $i$.
2. Choose expert $i$ with prob $\frac{w_i}{W}$, $W = \sum_i w_i$.
3. $w_i \leftarrow w_i(1 - \varepsilon)\ell_i^t$

$W(t)$ sum of $w_i$ at time $t$. $W(0) = n$

Best expert, $b$, loses $L^*$ total. $\rightarrow W(T) \geq w_b \geq (1 - \varepsilon)L^*$.

$L_t = \sum_i \frac{w_i \ell_i^t}{W}$ expected loss of alg. in time $t$.

Claim: $W(t + 1) \leq W(t)(1 - \varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

$W(t + 1) = \sum_i (1 - \varepsilon)\ell_i^t w_i \leq \sum_i (1 - \varepsilon \ell_i^t)w_i = \sum_i w_i - \varepsilon \sum_i w_i \ell_i^t$

$= \sum_i w_i \left(1 - \varepsilon \frac{\sum_i w_i \ell_i^t}{\sum_i w_i}\right)$
$= W(t)(1 - \varepsilon L_t)$
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \, \prod_t (1 - \varepsilon L_t)\]
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs

\[(L^*) \ln (1 - \varepsilon) \leq \ln n + \sum \ln (1 - \varepsilon L_t)\]
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs

\[(L^*) \ln (1 - \varepsilon) \leq \ln n + \sum \ln (1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln (1 - \varepsilon) \leq -\varepsilon\)
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs

\[(L^*) \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon\)

\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs
\[(L^*) \ln (1 - \varepsilon) \leq \ln n + \sum \ln (1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln (1 - \varepsilon) \leq -\varepsilon\)
\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs

\[(L^*) \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon\)

\[-(L^*) (\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And

\[\sum_t L_t \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon} .\]
Analysis

$$(1 - \epsilon)^{L^*} \leq W(T) \leq n \prod_t(1 - \epsilon L_t)$$

Take logs

$$(L^*)\ln(1 - \epsilon) \leq \ln n + \sum \ln(1 - \epsilon L_t)$$

Use $-\epsilon - \epsilon^2 \leq \ln(1 - \epsilon) \leq -\epsilon$

$$-(L^*)(\epsilon + \epsilon^2) \leq \ln n - \epsilon \sum L_t$$

And

$$\sum_t L_t \leq (1 + \epsilon)L^* + \frac{\ln n}{\epsilon}.$$  

$\sum_t L_t$ is total expected loss of algorithm.
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs
\[(L^*) \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \( -\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon \)
\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And
\[\sum_t L_t \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}\]

\(\sum_t L_t\) is total expected loss of algorithm.
Within \((1 + \varepsilon)\)
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs
\[(L^*) \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \[-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon\]
\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And
\[\sum_t L_t \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}\]

\[\sum_t L_t\] is total expected loss of algorithm.
Within \((1 + \varepsilon)\) ish
Analysis

\[(1 - \varepsilon)^L \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs

\[L^* \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon\)

\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And

\[\sum t L_t \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}\]

\[\sum_t L_t\] is total expected loss of algorithm.

Within \((1 + \varepsilon)\) ish of the best expert!
Analysis

\[(1 - \varepsilon)^{L^*} \leq W(T) \leq n \prod_t (1 - \varepsilon L_t)\]

Take logs
\[(L^*) \ln(1 - \varepsilon) \leq \ln n + \sum \ln(1 - \varepsilon L_t)\]

Use \(-\varepsilon - \varepsilon^2 \leq \ln(1 - \varepsilon) \leq -\varepsilon\)

\[-(L^*)(\varepsilon + \varepsilon^2) \leq \ln n - \varepsilon \sum L_t\]

And
\[\sum_t L_t \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}\]

\[\sum_t L_t\] is total expected loss of algorithm.

Within \((1 + \varepsilon)\) ish of the best expert!

No factor of 2 loss!
Gains.

Why so negative?
Gains.

Why so negative?
Each day, each expert gives gain in $[0, 1]$. 

Multiplicative weights with $(1 + \varepsilon)g_t$.

$G \geq (1 - \varepsilon)G^* - \log n \varepsilon$ where $G^*$ is payoff of best expert.

Scaling: Not $[0, 1]$, say $[0, \rho]$.

$L \leq (1 + \varepsilon)L^* + \rho \log n \varepsilon$
Gains.

Why so negative?
Each day, each expert gives gain in $[0, 1]$.
Multiplicative weights with $(1 + \varepsilon)g_i^t$. 

$G \geq (1 - \varepsilon)G^* - \log n$ where $G^*$ is payoff of best expert.

Scaling: Not $[0, 1]$, say $[0, \rho]$.

$L \leq (1 + \varepsilon)L^* + \rho \log n$.
Why so negative?
Each day, each expert gives gain in $[0, 1]$.
Multiplicative weights with $(1 + \varepsilon)^{g_i^t}$.

$$G \geq (1 - \varepsilon)G^* - \frac{\log n}{\varepsilon}$$

where $G^*$ is payoff of best expert.
Gains.

Why so negative?
Each day, each expert gives gain in $[0, 1]$.
Multiplicative weights with $(1 + \varepsilon)^{g_i}$.

$$G \geq (1 - \varepsilon)G^* - \frac{\log n}{\varepsilon}$$

where $G^*$ is payoff of best expert.

Scaling:
Gains.

Why so negative?
Each day, each expert gives gain in $[0, 1]$.
Multiplicative weights with $(1 + \varepsilon)^g_i$.

$$G \geq (1 - \varepsilon)G^* - \frac{\log n}{\varepsilon}$$

where $G^*$ is payoff of best expert.

Scaling:
Not $[0, 1]$, say $[0, \rho]$. 
Gains.

Why so negative?
Each day, each expert gives gain in \([0, 1]\).

Multiplicative weights with \((1 + \varepsilon)^{g_i^t}\).

\[
G \geq (1 - \varepsilon)G^* - \frac{\log n}{\varepsilon}
\]

where \(G^*\) is payoff of best expert.

Scaling:
Not \([0, 1]\), say \([0, \rho]\).

\[
L \leq (1 + \varepsilon)L^* + \frac{\rho \log n}{\varepsilon}
\]
Summary: multiplicative weights.
Summary: multiplicative weights.

Framework: $n$ experts, each loses different amount every day.
Framework: $n$ experts, each loses different amount every day.
Perfect Expert: $\log n$ mistakes.
Summary: multiplicative weights.

Framework: \( n \) experts, each loses different amount every day.

Perfect Expert: \( \log n \) mistakes.

Imperfect Expert: best makes \( m \) mistakes.
Summary: multiplicative weights.

Framework: $n$ experts, each loses different amount every day.
Perfect Expert: $\log n$ mistakes.
Imperfect Expert: best makes $m$ mistakes.
Deterministic Strategy: $2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}$
Summary: multiplicative weights.

Framework: \(n\) experts, each loses different amount every day.
Perfect Expert: \(\log n\) mistakes.
Imperfect Expert: best makes \(m\) mistakes.
Deterministic Strategy: \(2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}\)
Real numbered losses: Best loses \(L^*\) total.
Summary: multiplicative weights.

Framework: \( n \) experts, each loses different amount every day.
Perfect Expert: \( \log n \) mistakes.
Imperfect Expert: best makes \( m \) mistakes.
Deterministic Strategy: \( 2(1 + \varepsilon)m + \frac{\log n}{\varepsilon} \)
Real numbered losses: Best loses \( L^* \) total.
Randomized Strategy: \( (1 + \varepsilon)L^* + \frac{\log n}{\varepsilon} \)
Summary: multiplicative weights.

Framework: $n$ experts, each loses different amount every day.
Perfect Expert: $\log n$ mistakes.
Imperfect Expert: best makes $m$ mistakes.
Deterministic Strategy: $2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}$
Real numbered losses: Best loses $L^*$ total.
Randomized Strategy: $(1 + \varepsilon)L^* + \frac{\log n}{\varepsilon}$
Strategy:
Summary: multiplicative weights.

Framework: $n$ experts, each loses different amount every day.
Perfect Expert: $\log n$ mistakes.
Imperfect Expert: best makes $m$ mistakes.

Deterministic Strategy: $2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}$

Real numbered losses: Best loses $L^*$ total.

Randomized Strategy: $(1 + \varepsilon)L^* + \frac{\log n}{\varepsilon}$

Strategy:
  - Choose proportional to weights
Summary: multiplicative weights.

Framework: \( n \) experts, each loses different amount every day.
Perfect Expert: \( \log n \) mistakes.
Imperfect Expert: best makes \( m \) mistakes.

Deterministic Strategy: \( 2(1 + \varepsilon)m + \frac{\log n}{\varepsilon} \)

Real numbered losses: Best loses \( L^* \) total.

Randomized Strategy: \( (1 + \varepsilon)L^* + \frac{\log n}{\varepsilon} \)

Strategy:
Choose proportional to weights
multiply weight by \( (1 - \varepsilon)^{\text{loss}} \).
Framework: $n$ experts, each loses different amount every day.

Perfect Expert: $\log n$ mistakes.

Imperfect Expert: best makes $m$ mistakes.

Deterministic Strategy: $2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}$

Real numbered losses: Best loses $L^*$ total.

Randomized Strategy: $(1 + \varepsilon)L^* + \frac{\log n}{\varepsilon}$

Strategy:

Choose proportional to weights

multiply weight by $(1 - \varepsilon)^{\text{loss}}$.

Multiplicative weights framework!
Summary: multiplicative weights.

Framework: $n$ experts, each loses different amount every day.
Perfect Expert: $\log n$ mistakes.
Imperfect Expert: best makes $m$ mistakes.

Deterministic Strategy: $2(1 + \varepsilon)m + \frac{\log n}{\varepsilon}$

Real numbered losses: Best loses $L^*$ total.

Randomized Strategy: $(1 + \varepsilon)L^* + \frac{\log n}{\varepsilon}$

Strategy:
Choose proportional to weights
multiply weight by $(1 - \varepsilon)^{\text{loss}}$.

Multiplicative weights framework!
Applications next!
Two person zero sum games.

$m \times n$ payoff matrix $A$. 
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.

Payoff for strategy pair $(x, y)$:

\[
\text{payoff} = x^T A y
\]

Recall row minimizes, column maximizes.

Equilibrium pair:

\[
(x^*, y^*) = \left( x^* \right)^T A \left( y^* \right) = \min x \left( x^* \right)^T A y
\]
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.

Payoff for strategy pair $(x, y)$:

$$p(x, y) = x^t Ay$$

That is,

$$\sum_i x_i \left( \sum_j a_{i,j} y_j \right) = \sum_j \left( \sum_i x_i a_{i,j} \right) y_j.$$
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.

Payoff for strategy pair $(x, y)$:

$$p(x, y) = x^t Ay$$

That is,

$$\sum_i x_i \left( \sum_j a_{i,j} y_j \right) = \sum_j \left( \sum_i x_i a_{i,j} \right) y_j.$$

Recall row minimizes, column maximizes.
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.

Payoff for strategy pair $(x, y)$:

$$ p(x, y) = x^t Ay $$

That is,

$$ \sum_i x_i \left( \sum_j a_{i,j} y_j \right) = \sum_j \left( \sum_i x_i a_{i,j} \right) y_j. $$

Recall row minimizes, column maximizes.

Equilibrium pair: $(x^*, y^*)$?
Two person zero sum games.

$m \times n$ payoff matrix $A$.

Row mixed strategy: $x = (x_1, \ldots, x_m)$.
Column mixed strategy: $y = (y_1, \ldots, y_n)$.

Payoff for strategy pair $(x, y)$:

$$p(x, y) = x^t A y$$

That is,

$$\sum_i x_i \left( \sum_j a_{i,j} y_j \right) = \sum_j \left( \sum_i x_i a_{i,j} \right) y_j.$$ 

Recall row minimizes, column maximizes.

Equilibrium pair: $(x^*, y^*)$?

$$(x^*)^tAy^* = \max_y (x^*)^tAy = \min_x x^tAy^*.$$ 

(No better column strategy, no better row strategy.)
Equilibrium.

Equilibrium pair: \((x^*, y^*)\)?

\[
p(x, y) = (x^*)^t Ay^* = \max_y (x^*)^t Ay = \min_x x^t Ay^*.
\]

(No better column strategy, no better row strategy.)

\(^1A^{(i)}\) is \(i\)th row.
Equilibrium.

Equilibrium pair: \((x^*, y^*)\)?

\[
p(x, y) = (x^*)^t A y^* = \max_y (x^*)^t A y = \min_x x^t A y^*.
\]

(No better column strategy, no better row strategy.)

No row is better:

\[
\min_i A^{(i)} \cdot y = (x^*)^t A y^*. \quad ^1
\]
Equilibrium.

Equilibrium pair: \((x^*, y^*)\)?

\[ p(x, y) = (x^*)^t Ay^* = \max_y (x^*)^t Ay = \min_x x^t Ay^*. \]

(No better column strategy, no better row strategy.)

No row is better:

\[ \min_i A^{(i)} \cdot y = (x^*)^t Ay^*. \]

No column is better:

\[ \max_j (A^t)^{(j)} \cdot x = (x^*)^t Ay^*. \]

\(^1 A^{(i)} \) is \( i \)th row.
Best Response

Column goes first:

\[ R = \max_{x} \min_{y} (x^T Ay) \]

Note: \( x \) can be \((0, 0, \ldots, 1, \ldots, 0)\).

Example: Roshambo.

Value of \( R \)?

Row goes first:

\[ C = \min_{x} \max_{y} (x^T Ay) \]

Again: \( y \) of form \((0, 0, \ldots, 1, \ldots, 0)\).

Example: Roshambo.

Value of \( C \)?
Best Response

**Column goes first:**
Find $y$, where best row is not too low.

\[ R = \max_y \min_x (x^t A y). \]
Best Response

**Column goes first:**
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^tAy).$$

Note: $x$ can be $(0,0,...,1,...0)$.

**Row goes first:**
Find $x$, where best column is not high.

$$C = \min_x \max_y (x^tAy).$$

Again: $y$ of form $(0,0,...,1,...0)$. 

Example: Roshambo.

Value of $R$?

Example: Roshambo.

Value of $C$?
Best Response

**Column goes first:**
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^t A y).$$

Note: $x$ can be $(0, 0, \ldots, 1, \ldots 0)$.

Example: Roshambo.
Best Response

**Column goes first:**
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^t Ay).$$

Note: $x$ can be $(0,0,\ldots,1,\ldots 0)$.

Example: Roshambo. Value of $R$?
**Best Response**

**Column goes first:**
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^t A y).$$

Note: $x$ can be $(0,0,\ldots,1,\ldots 0)$.

Example: Roshambo. Value of $R$?

**Row goes first:**
Find $x$, where best column is not high.
Best Response

**Column goes first:**  
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^t Ay).$$

Note: $x$ can be $(0, 0, \ldots, 1, \ldots 0)$.

Example: Roshambo. Value of $R$?

**Row goes first:**  
Find $x$, where best column is not high.

$$C = \min_x \max_y (x^t Ay).$$
Column goes first:
Find $y$, where best row is not too low.

$$R = \max_y \min_x (x^t Ay).$$

Note: $x$ can be $(0, 0, \ldots, 1, \ldots 0)$.

Example: Roshambo. Value of $R$?

Row goes first:
Find $x$, where best column is not high.

$$C = \min_x \max_y (x^t Ay).$$

Again: $y$ of form $(0, 0, \ldots, 1, \ldots 0)$. 
Best Response

**Column goes first:**
Find $y$, where best row is not too low..

$$R = \max_y \min_x (x^t Ay).$$

Note: $x$ can be $(0, 0, \ldots, 1, \ldots 0)$.
Example: Roshambo. Value of $R$?

**Row goes first:**
Find $x$, where best column is not high.

$$C = \min_x \max_y (x^t Ay).$$

Agin: $y$ of form $(0, 0, \ldots, 1, \ldots 0)$.
Example: Roshambo.
**Best Response**

**Column goes first:**
Find \( y \), where best row is not too low.

\[
R = \max_y \min_x (x^t Ay).
\]

Note: \( x \) can be \((0, 0, \ldots, 1, \ldots 0)\).

Example: Roshambo. Value of \( R \)?

**Row goes first:**
Find \( x \), where best column is not high.

\[
C = \min_x \max_y (x^t Ay).
\]

Again: \( y \) of form \((0, 0, \ldots, 1, \ldots 0)\).

Example: Roshambo. Value of \( C \)?
Duality.

\[ R = \max_y \min_x (x^t A y). \]

Weak Duality: 
\[ R \leq C. \]

Proof: Better to go second.

At equilibrium \((x^*, y^*)\), payoff \(v\):
- Row payoffs (\(Ay^*)\) all \(\geq v\) \(\Rightarrow R \geq v\).
- Column payoffs (\((x^*)^t A\)) all \(\leq v\) \(\Rightarrow v \geq C\).

\(\Rightarrow R \geq C\).

Equilibrium \(\Rightarrow R = C\)!

Strong Duality: There is an equilibrium point! and \(R = C\)!

Doesn't matter who plays first!
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

Weak Duality:
\[ R \leq C. \]

Proof:
Better to go second.

At equilibrium \((x^*, y^*)\), payoff \(v\):
\[ \text{row payoffs (} x^* A y^*) \text{ all } \geq v \Rightarrow R \geq v. \]
\[ \text{column payoffs (} x^* t A \text{) all } \leq v \Rightarrow v \geq C. \]
\[ \Rightarrow R \geq C. \]

Equilibrium \(\Rightarrow R = C!\)

Strong Duality:
There is an equilibrium point!

and \(R = C!\)

Doesn't matter who plays first!
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

**Equilibrium** \( \Rightarrow R = C \!\!\!\!\rangle \)

**Strong Duality:** There is an equilibrium point! And \( R = C \!\!\!\!\rangle \)

Doesn't matter who plays first!
Duality.

\[ R = \max_{y} \min_{x} (x^t Ay). \]
\[ C = \min_{x} \max_{y} (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \( v \): row payoffs \((Ay^*)\) all \( \geq v \)

Doesn't matter who plays first!
Duality.

\[ R = \max_y \min_x (x^t A y). \]

\[ C = \min_x \max_y (x^t A y). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \( v \):
row payoffs \((Ay^*)\) all \(\geq v \implies R \geq v.\)
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):
row payoffs \((Ay^*)\) all \(\geq v \implies R \geq v.\)
column payoffs \(((x^*)^t A)\) all \(\leq v.\)
Duality.

\[
R = \max_y \min_x (x^t Ay).
\]
\[
C = \min_x \max_y (x^t Ay).
\]

**Weak Duality:** \( R \leq C \).

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):

- row payoffs \((Ay^*)\) all \(\geq v\) \(\implies R \geq v\).
- column payoffs \(((x^*)^t A)\) all \(\leq v\) \(\implies v \geq C\).
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):
row payoffs \((Ay^*)\) all \(\geq v \implies R \geq v.\)
column payoffs \(((x^*)^t A)\) all \(\leq v \implies v \leq C.\)
\[ \implies R \geq C \]
Duality.

\[ R = \max_y \min_x (x^t A y). \]
\[ C = \min_x \max_y (x^t A y). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):
- row payoffs \((Ay^*)\) all \( \geq v \) \( \implies \) \( R \geq v. \)
- column payoffs \(((x^*)^t A)\) all \( \leq v \) \( \implies \) \( v \geq C. \)

\[ \implies R \geq C \]

Equilibrium \( \implies R = C! \)
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

Weak Duality: \( R \leq C. \)

Proof: Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\): row payoffs \((Ay^*)\) all \(\geq v \implies R \geq v.\)

column payoffs \(((x^*)^t A)\) all \(\leq v \implies v \geq C.\)

\[ \implies R \geq C \]

Equilibrium \(\implies R = C! \)

Strong Duality: There is an equilibrium point!
Duality.

\[ R = \max_y \min_x (x^t Ay). \]
\[ C = \min_x \max_y (x^t Ay). \]

**Weak Duality:** \( R \leq C. \)

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \(v\):
row payoffs \((Ay^*)\) all \(\geq v \implies R \geq v. \)
column payoffs \(((x^*)^t A)\) all \(\leq v \implies v \geq C. \)
\(\implies R \geq C\)

Equilibrium \(\implies R = C!\)

**Strong Duality:** There is an equilibrium point! and \(R = C!\)
Duality.

\[
R = \max_y \min_x (x^t Ay), \\
C = \min_x \max_y (x^t Ay).
\]

**Weak Duality:** \( R \leq C \).

**Proof:** Better to go second.

At Equilibrium \((x^*, y^*)\), payoff \( v \):
row payoffs \((Ay^*)\) all \( \geq v \) \( \implies \) \( R \geq v \).
column payoffs \(((x^*)^t A)\) all \( \leq v \) \( \implies \) \( v \geq C \).
\( \implies \) \( R \geq C \)

Equilibrium \( \implies \) \( R = C \)!

**Strong Duality:** There is an equilibrium point! and \( R = C \! \)!

Doesn’t matter who plays first!
Proof of Equilibrium.

Later.
Proof of Equilibrium.

Later. Well in just a minute.....
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t A y \]
\[ R(y) = \min_x x^t A y \]
Always: \[ R(y) \leq C(x) \]
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \]
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t A y \]
\[ R(y) = \min_x x^t A y \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]

Approximate Equilibrium: \( C(x) - R(y) \leq \epsilon. \)
Proof of Equilibrium.

Later. Well in just a minute....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]

Approximate Equilibrium: \( C(x) - R(y) \leq \varepsilon. \)

With \( R(y) < C(x) \)
Proof of Equilibrium.

Later. Well in just a minute.....

Aproximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]

Approximate Equilibrium: \( C(x) - R(y) \leq \varepsilon. \)

With \( R(y) < C(x) \)

\( \rightarrow \) “Response \( y \) to \( x \) is within \( \varepsilon \) of best response”
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t Ay \]
\[ R(y) = \min_x x^t Ay \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]

Approximate Equilibrium: \( C(x) - R(y) \leq \varepsilon. \)

With \( R(y) < C(x) \)

\( \rightarrow \) “Response \( y \) to \( x \) is within \( \varepsilon \) of best response”

\( \rightarrow \) “Response \( x \) to \( y \) is within \( \varepsilon \) of best response”
Proof of Equilibrium.

Later. Well in just a minute.....

Approximate equilibrium ...

\[ C(x) = \max_y x^t A y \]
\[ R(y) = \min_x x^t A y \]

Always: \( R(y) \leq C(x) \)

Strategy pair: \((x, y)\)

Equilibrium: \((x, y)\)

\[ R(y) = C(x) \rightarrow C(x) - R(y) = 0. \]

Approximate Equilibrium: \( C(x) - R(y) \leq \varepsilon. \)

With \( R(y) < C(x) \)

\( \rightarrow \) “Response \( y \) to \( x \) is within \( \varepsilon \) of best response”

\( \rightarrow \) “Response \( x \) to \( y \) is within \( \varepsilon \) of best response”
Proof of approximate equilibrium.

How?

(A) Using geometry.
Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
(D) By the skin of my teeth.
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
(D) By the skin of my teeth.

(C)
Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.

(C) Using multiplicative weights.

(D) By the skin of my teeth.

(C) ..and (D).
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
(D) By the skin of my teeth.

(C) ..and (D).
Not hard.
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
(D) By the skin of my teeth.

(C) ..and (D).
Not hard. Even easy.
Proof of approximate equilibrium.

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.
(D) By the skin of my teeth.

(C) ..and (D).
Not hard. Even easy. Still, head scratching happens.
Games and experts

Again: find \((x^*, y^*)\), such that
Games and experts

Again: find \((x^*, y^*)\), such that
\[
(max_y x^* Ay) - (min_x x^* Ay^*) \leq \varepsilon
\]
Games and experts

Again: find \((x^*, y^*)\), such that

\[
\left( \max_y x^*Ay \right) - \left( \min_x x^*Ay^* \right) \leq \varepsilon
\]

\[
C(x^*) - R(y^*) \leq \varepsilon
\]
Games and experts

Again: find \((x^*, y^*)\), such that

\[
\left( \max_y x^* Ay \right) - \left( \min_x x^* Ay^* \right) \leq \epsilon
\]

\[
C(x^*) - R(y^*) \leq \epsilon
\]
Games and experts

Again: find \((x^*, y^*)\), such that

\[
\max_y x^* A y - \min_x x^* A y^* \leq \epsilon
\]

\[
C(x^*) - R(y^*) \leq \epsilon
\]

Experts Framework:

\(n\) Experts,
Games and experts

Again: find \((x^*, y^*)\), such that

\[
(\max_y x^* Ay) - (\min_x x^* Ay) \leq \varepsilon
\]

\[
C(x^*) - R(y^*) \leq \varepsilon
\]

Experts Framework:

\(n\) Experts, \(T\) days,
Games and experts

Again: find \((x^*, y^*)\), such that
\[
(\max_y x^* A y) - (\min_x x^* A y^*) \leq \varepsilon
\]
\[
C(x^*) - R(y^*) \leq \varepsilon
\]

Experts Framework:
\(n\) Experts, \(T\) days, \(L^*\) - total loss of best expert.
Games and experts

Again: find \((x^*, y^*)\), such that

\[
\left( \max_y x^* Ay \right) - \left( \min_x x^* Ay^* \right) \leq \varepsilon
\]

\[
C(x^*) - R(y^*) \leq \varepsilon
\]

Experts Framework:
\(n\) Experts, \(T\) days, \(L^*\) -total loss of best expert.

Multiplicative Weights Method yields loss \(L\) where
Games and experts

Again: find \((x^*, y^*)\), such that
\[
(\max_y x^* A y) - (\min_x x^* A y^*) \leq \epsilon
\]
\[
C(x^*) - R(y^*) \leq \epsilon
\]

Experts Framework:
n Experts, \(T\) days, \(L^*\) -total loss of best expert.

Multiplicative Weights Method yields loss \(L\) where
\[
L \leq (1 + \epsilon)L^* + \frac{\log n}{\epsilon}
\]
Assume:

A has payoffs in $[0, 1]$. For $T = \log n \in \mathbb{R}^{\geq 0}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution.

Let $x_t$ be distribution (row strategy) on day $t$.

2) Each day, adversary plays best column response to $x_t$. Choose column of $A$ that maximizes row's expected loss. Let $y_t$ be indicator vector for this column.
Assume: $A$ has payoffs in $[0, 1]$. 

Games and Experts.
Games and Experts.

Assume: \(A\) has payoffs in \([0, 1]\).

For \(T = \frac{\log n}{\varepsilon^2}\) days:
Games and Experts.

Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts.
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution.
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution. Let $x_t$ be distribution (row strategy) on day $t$. 
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution. Let $x_t$ be distribution (row strategy) on day $t$.

2) Each day, adversary plays best column response to $x_t$. Choose column of $A$ that maximizes row’s expected loss.
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\epsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution. Let $x_t$ be distribution (row strategy) on day $t$.

2) Each day, adversary plays best column response to $x_t$. Choose column of $A$ that maximizes row’s expected loss.
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution. Let $x_t$ be distribution (row strategy) on day $t$.

2) Each day, adversary plays best column response to $x_t$. Choose column of $A$ that maximizes row's expected loss. Let $y_t$ be indicator vector for this column.
Assume: $A$ has payoffs in $[0, 1]$.

For $T = \frac{\log n}{\varepsilon^2}$ days:

1) $m$ pure row strategies are experts. Use multiplicative weights, produce row distribution. Let $x_t$ be distribution (row strategy) on day $t$.

2) Each day, adversary plays best column response to $x_t$. Choose column of $A$ that maximizes row’s expected loss. Let $y_t$ be indicator vector for this column.
Approximate Equilibrium!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.
Let $y^* = \frac{1}{T} \sum_{t} y_t$
Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( y^* = \frac{1}{T} \sum_t y_t \) and \( x^* = \arg\min_{x_t} x_t A y_t \).

Approximate Equilibrium!
Approximate Equilibrium!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( y^* = \frac{1}{T} \sum_t y_t \) and \( x^* = \arg\min_{x_t} x_t A y_t \).

Claim: \((x^*, y^*)\) are \(2\varepsilon\)-optimal for matrix \( A \).
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \text{argmin}_{x_t} x_t A y_t$.

**Claim:** $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$. 
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$. 

Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$. 
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.
Approximate Equilibrium!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( y^* = \frac{1}{T} \sum_t y_t \) and \( x^* = \arg\min_{x_t} x_t A y_t \).

Claim: \((x^*, y^*)\) are \(2\epsilon\)-optimal for matrix \( A\).

Column payoff: \( C(x^*) = \max_y x^* A y \).

Loss on day \( t \), \( x_t A y_t \geq C(x^*) \) by the choice of \( x^* \).

Thus, algorithm loss, \( L \), is \( \geq T \times C(x^*) \).

Best expert: \( L^* \)- best row against all the columns played.

best row against \( \sum_t A y_t \)
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t Ay_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Loss on day $t$, $x_t Ay_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $T \times y^* = \sum_t y_t$
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$

$\rightarrow$ best row against $T \times A y^*$. 
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_tAy_t$.

**Claim:** $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^*Ay$.

Loss on day $t$, $x_tAy_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

\[
\text{best row against } \sum_t Ay_t \text{ and } T \times y^* = \sum_t y_t \\
\rightarrow \text{best row against } T \times Ay^*.
\]

\[
\rightarrow L^* \leq T \times R(y^*).
\]
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t Ay_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Loss on day $t$, $x_t Ay_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $T \times y^* = \sum_t y_t$

$\rightarrow$ best row against $T \times Ay^*$.

$\rightarrow L^* \leq T \times R(y^*)$. 

Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$
- $L^* \leq T \times R(y^*)$.

Multiplicative Weights:
Approximate Equilibrium!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( y^* = \frac{1}{T} \sum_t y_t \) and \( x^* = \text{argmin}_{x_t} x_t Ay_t \).

**Claim:** \((x^*, y^*)\) are \(2\varepsilon\)-optimal for matrix \( A\).

Column payoff: \( C(x^*) = \max_y x^* Ay \).

Loss on day \( t \), \( x_t Ay_t \geq C(x^*) \) by the choice of \( x^* \).

Thus, algorithm loss, \( L \), is \( \geq T \times C(x^*) \).

Best expert: \( L^* \)- best row against all the columns played.

- best row against \( \sum_t Ay_t \) and \( T \times y^* = \sum_t y_t \)
- \( \rightarrow \) best row against \( T \times Ay^* \).
- \( \rightarrow L^* \leq T \times R(y^*) \).

Multiplicative Weights: \( L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon} \)
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

**Claim:** $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$ - best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$
- $\rightarrow$ best row against $T \times A y^*$.
- $\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$T \times C(x^*) \leq (1 + \varepsilon) T \times R(y^*) + \frac{\ln n}{\varepsilon}$
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t Ay_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Loss on day $t$, $x_t Ay_t \geq C(x^*)$ by the choice of $x^*$.
Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$ - best row against all the columns played.

- best row against $\sum_t Ay_t$ and $T \times y^* = \sum_t y_t$
- $\rightarrow$ best row against $T \times Ay^*$.
- $\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$T \times C(x^*) \leq (1 + \varepsilon)T \times R(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$
- best row against $T \times A y^*$.  

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}$

$T \times C(x^*) \leq (1 + \varepsilon) T \times R(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon) R(y^*) + \frac{\ln n}{\varepsilon T}$

$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.  

Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$.

Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$
- $\rightarrow$ best row against $T \times A y^*$.
- $\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$T \times C(x^*) \leq (1 + \varepsilon)T \times R(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$

$$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}.$$

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$
Approximate Equilibrium!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $y^* = \frac{1}{T} \sum_t y_t$ and $x^* = \arg\min_{x_t} x_t A y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Loss on day $t$, $x_t A y_t \geq C(x^*)$ by the choice of $x^*$. Thus, algorithm loss, $L$, is $\geq T \times C(x^*)$.

Best expert: $L^*$ - best row against all the columns played.

- best row against $\sum_t A y_t$ and $T \times y^* = \sum_t y_t$
- $\rightarrow$ best row against $T \times A y^*$.
- $\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}$

- $T \times C(x^*) \leq (1 + \varepsilon) T \times R(y^*) + \frac{\ln n}{\varepsilon}$
- $\rightarrow C(x^*) \leq (1 + \varepsilon) R(y^*) + \frac{\ln n}{\varepsilon T}$
- $\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$

$\rightarrow C(x^*) - R(y^*) \leq 2\varepsilon$. 

Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$. 

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r \Rightarrow L \geq T \times C(x^*)$.

Best expert: $L^*$ - best row against all the columns played. 

Let $y^*$ be best row against $\sum_t A y_t$ and $T y^* = \sum_t y_t$.

$\Rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon) L^* + \ln n$.

$\Rightarrow C(x^*) \leq (1 + \varepsilon) R(y^*) + \ln n \varepsilon T$.

$T = \ln n \varepsilon^2$, $R(y^*) \leq 1$.

$\Rightarrow C(x^*) - R(y^*) \leq 2 \varepsilon$. 
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$. 
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$. 
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

**Claim**: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$. 

Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.
   Let $y_r$ be best response to $C(x^*)$. 


Approximate Equilibrium: slightly different!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( x^* = \frac{1}{T} \sum_t x_t \) and \( y^* = \frac{1}{T} \sum_t y_t \).

Claim: \((x^*, y^*)\) are \( 2\varepsilon \)-optimal for matrix \( A \).

Column payoff: \( C(x^*) = \max_y x^* A y \).

Let \( y_r \) be best response to \( C(x^*) \).

Day \( t \), \( x_t A y_t \geq x_t A y_r - y_t \) is best response to \( x_t \).
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t Ay_t \geq x_t Ay_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq T \times C(x^*)$. 
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t Ay_t \geq x_t Ay_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$ - best row against all the columns played.
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t A y_t$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t Ay_t \geq x_t Ay_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t Ay_t \geq x_t Ay_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$

$\rightarrow$ best row against $TAy^*$. 
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\epsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t A y_t$ and $T y^* = \sum_t y_t$

$\rightarrow$ best row against $T A y^*$.

$\rightarrow L^* \leq T \times R(y^*)$. 
Approximate Equilibrium: slightly different!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( x^* = \frac{1}{T} \sum_t x_t \) and \( y^* = \frac{1}{T} \sum_t y_t \).

Claim: \((x^*, y^*)\) are \(2\varepsilon\)-optimal for matrix \( A \).

Column payoff: \( C(x^*) = \max_y x^*Ay \).

Let \( y_r \) be best response to \( C(x^*) \).

Day \( t \), \( x_tAy_t \geq x_tAy_r - y_t \) is best response to \( x_t \).

Algorithm loss: \( \sum_t x_tAy_t \geq \sum_t x_tAy_r \)

\[ L \geq T \times C(x^*). \]

Best expert: \( L^* \)- best row against all the columns played.

best row against \( \sum_t Ay_t \) and \( Ty^* = \sum_t y_t \)

\[ \rightarrow \text{best row against } TAy^*. \]

\[ \rightarrow L^* \leq T \times R(y^*). \]
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t Ay_t \geq x_t Ay_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$

$\rightarrow$ best row against $TAy^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights:
Approximate Equilibrium: slightly different!

Experts: \( x_t \) is strategy on day \( t \), \( y_t \) is best column against \( x_t \).

Let \( x^* = \frac{1}{T} \sum_t x_t \) and \( y^* = \frac{1}{T} \sum_t y_t \).

Claim: \((x^*, y^*)\) are \(2\varepsilon\)-optimal for matrix \( A\).

Column payoff: \( C(x^*) = \max_y x^* Ay \).

Let \( y_r \) be best response to \( C(x^*) \).

Day \( t \), \( x_t Ay_t \geq x_t Ay_r - y_t \) is best response to \( x_t \).

Algorithm loss: \( \sum_t x_t Ay_t \geq \sum_t x_t Ay_r \)

\( L \geq T \times C(x^*) \).

Best expert: \( L^* \)- best row against all the columns played.

best row against \( \sum_t Ay_t \) and \( Ty^* = \sum_t y_t \)

\( \rightarrow \) best row against \( TAy^* \).

\( \rightarrow L^* \leq T \times R(y^*) \).

Multiplicative Weights: \( L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon} \)
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$.

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t A y_t$ and $T y^* = \sum_t y_t$

$\rightarrow$ best row against $T A y^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$TC(x^*) \leq (1 + \varepsilon) TR(y^*) + \frac{\ln n}{\varepsilon}$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t A y_t$ and $T y^* = \sum_t y_t$

$\rightarrow$ best row against $T A y^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}$

$T C(x^*) \leq (1 + \varepsilon) T R(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon) R(y^*) + \frac{\ln n}{\varepsilon T}$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^*Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_tAy_t \geq x_tAy_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_tAy_t \geq \sum_t x_tAy_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$

$\rightarrow$ best row against $TAy^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$TC(x^*) \leq (1 + \varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$

$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^* A y$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_t A y_t \geq x_t A y_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t A y_t$ and $T y^* = \sum_t y_t$

$\rightarrow$ best row against $T A y^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \varepsilon) L^* + \frac{\ln n}{\varepsilon}$

$T C(x^*) \leq (1 + \varepsilon) T R(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1 + \varepsilon) R(y^*) + \frac{\ln n}{\varepsilon T}$

$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$
Approximate Equilibrium: slightly different!

Experts: $x_t$ is strategy on day $t$, $y_t$ is best column against $x_t$.

Let $x^* = \frac{1}{T} \sum_t x_t$ and $y^* = \frac{1}{T} \sum_t y_t$.

Claim: $(x^*, y^*)$ are $2\epsilon$-optimal for matrix $A$.

Column payoff: $C(x^*) = \max_y x^*Ay$.

Let $y_r$ be best response to $C(x^*)$.

Day $t$, $x_tAy_t \geq x_tAy_r - y_t$ is best response to $x_t$.

Algorithm loss: $\sum_t x_tAy_t \geq \sum_t x_tAy_r$

$L \geq T \times C(x^*)$.

Best expert: $L^*$- best row against all the columns played.

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$

$\rightarrow$ best row against $TAy^*$.

$\rightarrow L^* \leq T \times R(y^*)$.

Multiplicative Weights: $L \leq (1 + \epsilon)L^* + \frac{\ln n}{\epsilon}$

$TC(x^*) \leq (1 + \epsilon)TR(y^*) + \frac{\ln n}{\epsilon} \rightarrow C(x^*) \leq (1 + \epsilon)R(y^*) + \frac{\ln n}{\epsilon T}$

$\rightarrow C(x^*) - R(y^*) \leq \epsilon R(y^*) + \frac{\ln n}{\epsilon T}$.

$T = \frac{\ln n}{\epsilon^2}$, $R(y^*) \leq 1 \rightarrow C(x^*) - R(y^*) \leq 2\epsilon.$
Comments

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.
Comments

For any $\epsilon$, there exists an $\epsilon$-Approximate Equilibrium.

Does an equilibrium exist?
Comments

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.
Does an equilibrium exist? Yes.
Something about math here?
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
  Limit of a sequence on some closed set..hmmm..
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
- Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.
For any $\epsilon$, there exists an $\epsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
Comments

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
\[
T = \frac{\ln n}{\varepsilon^2}
\]
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

  Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
  Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$ Basically linear!
Comments

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

  Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$

Basically linear!

Versus Linear Programming: $O(n^3m)$
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

  Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm \frac{\log n}{\varepsilon^2}).$$

Basically linear!

Versus Linear Programming: $O(n^3 m)$ Basically quadratic.
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
   $T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2})$. Basically linear!

Versus Linear Programming: $O(n^3 m)$ Basically quadratic.
   (Faster linear programming: $O(\sqrt{n + m})$ linear system solves.)
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$
Basically linear!

Versus Linear Programming: $O(n^3m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear system solves.)
Still much slower
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?

$$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$

Basically linear!

Versus Linear Programming: $O(n^3m)$ Basically quadratic.

(Faster linear programming: $O(\sqrt{n+m})$ linear system solves.)

Still much slower ... and more complicated.
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
\[ T = \frac{\ln n}{\varepsilon^2} \to O\left(nm\frac{\log n}{\varepsilon^2}\right). \text{ Basically linear!} \]

Versus Linear Programming: $O(n^3m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear system solves.)
Still much slower ... and more complicated.

For any $\epsilon$, there exists an $\epsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
   Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
   
   $$T = \frac{\ln n}{\epsilon^2} \rightarrow O(nm \log \frac{n}{\epsilon^2}).$$ Basically linear!

Versus Linear Programming: $O(n^3m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear system solves.)
Still much slower ... and more complicated.

Also works with both using multiplicative weights.
Comments

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?
  Limit of a sequence on some closed set..hmmm..

Later: will use geometry, linear programming.

Complexity?
  $$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2}).$$ Basically linear!

Versus Linear Programming: $O(n^3m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear system solves.)
Still much slower ... and more complicated.

Also works with both using multiplicative weights.

“In practice.”
Homework 2 out this week.
Homework 2 out this week.
See you on Thursday.