Today.

Quick Review:

Today.

Quick Review:

experts framework/multiplicative weights algorithm

Today.

Quick Review:

experts framework/multiplicative weights algorithm

Finish:

Today.

Quick Review:
    experts framework/multiplicative weights algorithm
Finish:
    randomized multiplicative weights algorithm for experts framework.

Today.

Quick Review:
experts framework/multiplicative weights algorithm
Finish:
randomized multiplicative weights algorithm for experts framework.

Equilibrium for two person games:
using experts framework/MW algorithm.

Notes.

Got to definition of Approximate Equilibrium for zero sum games.

The multiplicative weights framework.

Expert's framework.

n experts.

Expert's framework.

n experts.

Every day, each offers a prediction.

Expert's framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

Expert's framework.

n experts.

Every day, each offers a prediction.

“Rain” or “Shine.”

Whose advise do you follow?

Expert's framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

Whose advise do you follow?

"The one who is correct most often."

Expert's framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

Whose advise do you follow?

"The one who is correct most often."

Sort of.

Expert's framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

Whose advise do you follow?

"The one who is correct most often."

Sort of.

How well do you do?

Infallible expert.

One of the expert's is infallible!

Infallible expert.

One of the expert's is infallible!

Your strategy?

Infallible expert.

One of the expert's is infallible!

Your strategy?

Choose any expert that has not made a mistake!

Infallible expert.

One of the expert's is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Infallible expert.

One of the expert's 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 expert's 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 expert's 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 expert's 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 expert's 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 expert's 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 expert's 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 expert's 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.

Concept Alert.

Note.

Adversary:

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”...

Concept Alert.

Note.

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

Concept Alert.

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!

Concept Alert.

Note.

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

Analysis of Algorithms: do as well as possible!

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: $n-1$

Back to mistake bound.

Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n-1$

Lower bound: adversary argument.

Back to mistake bound.

Infallible Experts.

Alg: Choose one of the perfect experts.

Mistake Bound: $n-1$

Lower bound: adversary argument.

Upper bound:

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.

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?

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?

Making decision, not trying to find 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?

Making decision, not trying to find expert!

Algorithm: Go with the majority of previously correct experts.

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?

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,
$|$“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,
$|$"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,
$|$"perfect" experts$|$ 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,
$|$"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,
|“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

$\vdots$

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 $\rightarrow$ ≤ 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 $\rightarrow$ ≤ 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 $\rightarrow$ ≤ 1 perfect expert

≥ 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 $\rightarrow$ ≤ 1 perfect expert

≥ 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

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.

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.

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.

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.

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. $w_i \rightarrow w_i/2$ 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? -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. $w_i \to w_i/2$ 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 $\frac{1}{2}$!

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

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 $\frac{3}{4}$?

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.

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.

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.

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.

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

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

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.

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 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.

$-m \leq -M \log(4/3) + logn.$

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.

$$-m \leq -M \log(4/3) + logn.$$

Solve for M.

$$M \leq (m + logn)/\log(4/3)$$

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.

$$-m \leq -M \log(4/3) + logn.$$

Solve for M.

$$M \leq (m + logn)/\log(4/3) \leq 2.4(m + \log n)$$

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.

$-m \leq -M \log(4/3) + logn.$

Solve for M.

$M \leq (m + logn)/\log(4/3) \leq 2.4(m + \log n)$

Multiple by $1 - \varepsilon$ for incorrect experts...

Analysis: continued.

$$\frac{1}{2^m} \le \sum_i w_i \le \left(\frac{3}{4}\right)^M n.$$

m - best expert mistakes M algorithm mistakes.

$$\frac{1}{2^m} \le \left(\frac{3}{4}\right)^M n.$$

Take log of both sides.

$$-m \le -M\log(4/3) + logn.$$

Solve for M.

$M \le (m + logn)/\log(4/3) \le 2.4(m + \log n)$

Multiple by $1-\varepsilon$ for incorrect experts...

$$(1-\varepsilon)^m \le \left(1 - \frac{\varepsilon}{2}\right)^M n.$$

Analysis: continued.

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

m - best expert mistakes M algorithm mistakes.

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

Take log of both sides.

$$-m \leq -M \log(4/3) + logn.$$

Solve for M.

$M \leq (m + logn)/\log(4/3) \leq 2.4(m + \log n)$

Multiple by $1 - \varepsilon$ for incorrect experts...

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

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.

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

Multiple 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.

$$\frac{1}{2^m} \le \sum_i w_i \le \left(\frac{3}{4}\right)^M n.$$

m - best expert mistakes M algorithm mistakes.

$$\frac{1}{2^m} \le \left(\frac{3}{4}\right)^M n.$$

Take log of both sides.

$$-m \le -M \log(4/3) + logn.$$

Solve for M.

$$M \le (m + logn)/\log(4/3) \le 2.4(m + \log n)$$

Multiple by $1-\varepsilon$ for incorrect experts...

$$(1-\varepsilon)^m \le \left(1 - \frac{\varepsilon}{2}\right)^M n.$$

Massage...

$$M \le 2(1+\varepsilon)m + \frac{2 \ln n}{\varepsilon}$$

Approaches a factor of two of best expert performance!

Best Analysis?

Two experts: A,B

Best Analysis?

Two experts: A,B

Bad example?

Best Analysis?

Two experts: A,B

Bad example?

Which is worse?

(A) A right on even, B right on odd.

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

Best Analysis?

Two experts: A,B

Bad example?

Which is worse?

(A) A right on even, B right on odd.

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

Best expert peformance: $T/2$ mistakes.

Best Analysis?

Two experts: A,B

Bad example?

Which is worse?

(A) A right on even, B right on odd.

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

Best expert peformance: $T/2$ mistakes.

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

Best Analysis?

Two experts: A,B

Bad example?

Which is worse?

(A) A right on even, B right on odd.

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

Best expert peformance: $T/2$ 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 $\propto w_i$

Randomization!!!!

Better approach?

Use?

Randomization!

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

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

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.

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 probabilty.

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 probabilty.

Make a mistake around $1/2$ of the time.

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 probabilty.

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 probabilty.

Make a mistake around $1/2$ of the time.

Best expert makes $T/2$ mistakes.

Rougly

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 probabilty.

Make a mistake around $1/2$ of the time.

Best expert makes $T/2$ mistakes.

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

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

Losses in $[0, 1]$.

Randomized algorithm

Losses in $[0,1]$.

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

Randomized algorithm

Losses in $[0,1]$.

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

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

Randomized algorithm

Losses in $[0,1]$.

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 $\frac{w_i}{W}$, $W = \sum_i w_i$.

Randomized algorithm

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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.

Randomized algorithm

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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. $\rightarrow W(T) \geq w_b \geq (1-\varepsilon)^{L^*}$.

Randomized algorithm

Losses in $[0,1]$.

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.

Randomized algorithm

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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

Losses in $[0,1]$.

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:

Randomized algorithm

Losses in $[0,1]$.

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:

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

Randomized algorithm

Losses in $[0,1]$.

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:

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

Randomized algorithm

Losses in $[0,1]$.

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:

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

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

Randomized algorithm

Losses in $[0,1]$.

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:

$$
\begin{aligned}
W(t+1) \leq \sum_i (1-\varepsilon \ell_i^t) w_i &= \sum_i w_i - \varepsilon \sum_i w_i \ell_i^t \\
&= \sum_i w_i \left(1 - \varepsilon \frac{\sum_i w_i \ell_i^t}{\sum_i w_i}\right) \\
&= W(t)(1-\varepsilon L_t)
\end{aligned}
$$

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

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

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.

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 $[0,1]$.

Gains.

Why so negative?

Each day, each expert gives gain in $[0,1]$.

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

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.

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:

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 $[0,\rho]$.

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.

Summary: multiplicative weights.

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

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

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!

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):

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,

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

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^*)?

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

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

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.)

[1] $A^{(i)}$ is ith 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:
$\min_i A^{(i)} \cdot y = (x^*)^t A y^*$. [1]

[1] $A^{(i)}$ is ith 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:
$\min_i A^{(i)} \cdot y = (x^*)^t A y^*$. [1]

No column is better:
$\max_j (A^t)^{(j)} \cdot x = (x^*)^t A y^*$.

[1] $A^{(i)}$ is ith row.

Best Response

Column goes first:

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..

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

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

Best Response

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

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

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..

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

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..

$$R = \max_{y} \min_{x} (x^t A y).$$

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.

$$C = \min_{x} \max_{y} (x^t A y).$$

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.

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

Agin: 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..

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

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.

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

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

Example: Roshambo. Value of C?

Duality.

$$R = \max_{y} \min_{x} (x^t A y).$$

Duality.

$$R = \max_{y} \min_{x} (x^t A y).$$
$$C = \min_{x} \max_{y} (x^t A y).$$

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. $\square$

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. $\square$

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 $\geq v$

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.

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

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 $\leq v$

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. $\square$

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.

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

Weak Duality: $R \leq C$.
Proof: Better to go second. $\square$

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

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.

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

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 $\implies R = C$!

Strong Duality: There is an equilibrium point!

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 $\implies R = C$!

Strong Duality: There is an equilibrium point! and $R = C$!

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 $\implies R = C$!

Strong Duality: There is an equilibrium point! and $R = C$!

Doesn't matter who plays first!

Proof of Equilibrium.

Later.

Proof of Equilibrium.

Later. Still later...

Proof of Equilibrium.

Later. Still later...

Aproximate equilibrium ...

Proof of Equilibrium.

Later. Still later...

Aproximate equilibrium ...

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

Proof of Equilibrium.

Later. Still later...

Aproximate equilibrium ...

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

$R(y) = \min_x x^t Ay$

Proof of Equilibrium.

Later. Still later...

Aproximate equilibrium ...

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

$R(y) = \min_x x^t Ay$

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

Proof of Equilibrium.

Later. Still later...

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)

Proof of Equilibrium.

Later. Still later...

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)

Proof of Equilibrium.

Later. Still later...

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)

$R(y) = C(x)$

Proof of Equilibrium.

Later. Still later...

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)

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

Proof of Equilibrium.

Later. Still later...

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

Proof of Equilibrium.

Later. Still later...

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 $R(y) \leq C(x)$

Proof of Equilibrium.

Later. Still later...

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 $R(y) \leq C(x)$
$\rightarrow$ "Response y to x is within ε of best response"

Proof of Equilibrium.

Later. Still later...

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 $R(y) \leq C(x)$
$\rightarrow$ "Response y to x is within ε of best response"
$\rightarrow$ "Response x to y is within ε of best response"

Proof of Equilibrium.

Later. Still later...

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 $R(y) \leq C(x)$
$\rightarrow$ "Response y to x is within ε of best response"
$\rightarrow$ "Response x to y is within ε 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

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

$C(x^*) \quad - \quad R(y^*) \leq \varepsilon$

Games and experts

Again: find $(x*, y*)$, such that

$(\max_y x^* A y) - (\min_x x^* A y^*) \leq \varepsilon$

$C(x^*) \quad - \quad R(y^*) \leq \varepsilon$

Experts Framework: n Experts, T days, L^* -total loss.

Games and experts

Again: find $(x*, y*)$, such that

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

$C(x^*) \quad - \quad R(y^*) \leq \varepsilon$

Experts Framework: n Experts, T days, L^* -total loss.

Multiplicative Weights Method yields loss L where

Games and experts

Again: find $(x*, y*)$, such that

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

$C(x^*) \quad - \quad R(y^*) \leq \varepsilon$

Experts Framework: n Experts, T days, L^* -total loss.

Multiplicative Weights Method yields loss L where

$L \leq (1+\varepsilon)L^* + \frac{\log n}{\varepsilon}$

Games and Experts.

Games and Experts.

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

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.

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.
Use multiplicative weights, produce row distribution.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t on day t.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t on day t.

2) Each day, adversary plays best column response to x_t.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t on day t.

2) Each day, adversary plays best column response to x_t.
Choose column of A that maximizes row's expected loss.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

Let $y^* = \frac{1}{T}\sum_t y_t$

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.
column best response is at least what it is against x_t.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.
column best response is at least what it is against x_t. Total loss, L is at least column payoff.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.
column best response is at least what it is against x_t. Total loss, L is at least column payoff. Best row payoff, L^* is roughly less than L due to MW anlysis.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.
column best response is at least what it is against x_t. Total loss, L is at least column payoff. Best row payoff, L^* is roughly less than L due to MW anlysis.
Combine bounds.

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.
Use multiplicative weights, produce row distribution.
Let x_t be distribution (row strategy) x_t 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.

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

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

Proof Idea:
x_t minimizes the best column response is chosen. Clearly good for row.
column best response is at least what it is against x_t. Total loss, L is at least column payoff. Best row payoff, L^* is roughly less than L due to MW anlysis.
Combine bounds. Done!

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 $y^* = \frac{1}{T} \sum_t y_t$

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

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 2ε-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 2ε-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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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!

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 2ε-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 $\geq TC(x^*)$.

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!

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 2ε-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 $\geq TC(x^*)$.

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^*.
$\rightarrow$ $L^* \leq TR(y^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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

best row against $\sum_t A y_t$ and $Ty^* = \sum_t y_t$
$\rightarrow$ best row against TAy^*.
$\rightarrow L^* \leq TR(y^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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

best row against $\sum_t A y_t$ and $Ty^* = \sum_t y_t$
$\rightarrow$ best row against TAy^*.
$\rightarrow$ $L^* \leq TR(y^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

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

Column payoff: $C(x^*) = \max_y x^* Ay$.
Loss on day t, $x_t A y_t \geq C(x^*)$ by the choice of x .
Thus, algorithm loss, L, is $\geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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 TAy^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon}$$

Approximate Equilibrium!

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 2ε-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 $\geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$

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 2ε-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 $\geq TC(x^*)$.

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

best row against $\sum_t A y_t$ and $Ty^* = \sum_t y_t$
$\rightarrow$ best row against TAy^*.
$\rightarrow L^* \leq TR(y^*)$.

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!

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 2ε-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 $\geq TC(x^*)$.

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

best row against $\sum_t A y_t$ and $Ty^* = \sum_t y_t$
$\rightarrow$ best row against TAy^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$
$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$

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 $x^* = \text{argmin}_{x_t} x_t A y_t$.

Claim: $(x^*, y)^*$ are 2ε-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 $\geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$
$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}$.

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$
$\rightarrow C(x^*) - R(y^*) \leq 2\varepsilon$.

Approximate Equilibrium: notes!

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

Approximate Equilibrium: notes!

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: notes!

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: notes!

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 2ε-optimal for matrix A.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* A y$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.
Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq TC(x^*)$.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
  Let y_r be best response to $C(x^*)$.
  Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.
  Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
  $L \geq TC(x^*)$.

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

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq TC(x^*)$.

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

best row against $\sum_t Ay_t$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Day t,y_t best response to $x_t \to x_t Ay_t \geq x_t Ay_r$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq TC(x^*)$.

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: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.
Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
$L \geq TC(x^*)$.

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: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \to x_t Ay_t \geq x_t Ay_r$.
Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
$L \geq TC(x^*)$.

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

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$
$\to$ best row against TAy^*.
$\to L^* \leq TR(y^*)$.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.
Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$
$L \geq TC(x^*)$.

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^*.
$\rightarrow$ $L^* \leq TR(y^*)$.

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \to x_t A y_t \geq x_t A y_r$.
Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$
$L \geq TC(x^*)$.

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

best row against $\sum_t Ay_t$ and $Ty^* = \sum_t y_t$
$\to$ best row against TAy^*.
$\to L^* \leq TR(y^*)$.

Multiplicative Weights:

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.
Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
$L \geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.
Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$
$L \geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon}$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* A y$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.
Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$
$L \geq TC(x^*)$.

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^*.
$\rightarrow$ $L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

Column payoff: $C(x^*) = \max_y x^* Ay$.
Let y_r be best response to $C(x^*)$.
Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.
Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$
$L \geq TC(x^*)$.

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^*.
$\rightarrow L^* \leq TR(y^*)$.

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: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Day t,y_t best response to $x_t \rightarrow x_t A y_t \geq x_t A y_r$.

Algorithm loss: $\sum_t x_t A y_t \geq \sum_t x_t A y_r$

$L \geq TC(x^*)$.

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^*.

$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$

$$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}.$$

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1$

Approximate Equilibrium: notes!

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 2ε-optimal for matrix A.

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

Let y_r be best response to $C(x^*)$.

Day t,y_t best response to $x_t \rightarrow x_t Ay_t \geq x_t Ay_r$.

Algorithm loss: $\sum_t x_t Ay_t \geq \sum_t x_t Ay_r$

$L \geq TC(x^*)$.

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^*.

$\rightarrow L^* \leq TR(y^*)$.

Multiplicative Weights: $L \leq (1+\varepsilon)L^* + \frac{\ln n}{\varepsilon}$

$$TC(x^*) \leq (1+\varepsilon)TR(y^*) + \frac{\ln n}{\varepsilon} \rightarrow C(x^*) \leq (1+\varepsilon)R(y^*) + \frac{\ln n}{\varepsilon T}$$
$$\rightarrow C(x^*) - R(y^*) \leq \varepsilon R(y^*) + \frac{\ln n}{\varepsilon T}.$$

$T = \frac{\ln n}{\varepsilon^2}$, $R(y^*) \leq 1 \rightarrow C(x^*) - R(y^*) \leq 2\varepsilon$.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist?

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here?

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Complexity?

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Complexity?

$T = \frac{\ln n}{\varepsilon^2}$

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Complexity?

$T = \frac{\ln n}{\varepsilon^2} \rightarrow O(nm\frac{\log n}{\varepsilon^2})$.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Complexity?
$T = \frac{\ln n}{\varepsilon^2} \to O(nm\frac{\log n}{\varepsilon^2})$. Basically linear!

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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: $O(n^3 m)$

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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 solution solves.)

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Later: will use geometry, linear programming.

Complexity?
$T = \frac{\ln n}{\varepsilon^2} \to O(nm\frac{\log n}{\varepsilon^2})$. Basically linear!

Versus Linear Programming: $O(n^3 m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear solution solves.)
Still much slower

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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 solution solves.)
Still much slower ... and more complicated.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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 solution solves.)
Still much slower ... and more complicated.

Dynamics: best response, update weight, best response.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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 solution solves.)
Still much slower ... and more complicated.

Dynamics: best response, update weight, best response.

Also works with both using multiplicative weights.

Comments

For any ε, there exists an ε-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

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 solution solves.)
Still much slower ... and more complicated.

Dynamics: best response, update weight, best response.

Also works with both using multiplicative weights.

"In practice."

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \dots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.
Toll: charge most loaded edge.

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.
Toll: charge most loaded edge.

Defense: Toll: maximize shortest path under tolls.

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.
Toll: charge most loaded edge.

Defense: Toll: maximize shortest path under tolls.
Route: minimize max loaded on any edge.

Toll/Congestion

Given: $G = (V, E)$.
Given $(s_1, t_1) \ldots (s_k, t_k)$.
Row: choose routing of all paths.
Column: choose edge.
Row pays if column chooses edge on any path.

Matrix:
row for each routing: r
column for each edge: e

$A[r, e]$ is congestion on edge e by routing r

Offense: (Best Response.)
Router: route along shortest paths.
Toll: charge most loaded edge.

Defense: Toll: maximize shortest path under tolls.
Route: minimize max loaded on any edge.

Two person game.

Row is router.

Two person game.

Row is router.

An exponential number of rows.

Two person game.

Row is router.

An exponential number of rows.

Two person game with experts won't be so easy to implement.

Two person game.

Row is router.

An exponential number of rows.

Two person game with experts won’t be so easy to implement.

Version with row and column flipped may work.

Two person game.

Row is router.

An exponential number of rows.

Two person game with experts won't be so easy to implement.

Version with row and column flipped may work.

Next Time.