Today: The multiplicative weights algorithm.

KFOX

KCBS

NOAA

KTVU

KRON

Today: The multiplicative weights algorithm.

KFOX

KCBS

NOAA

KTVU

KRON

Today: The multiplicative weights algorithm.

KFOX

KCBS

NOAA

KTVU

KRON S

Today: The multiplicative weights algorithm.

KFOX

KCBS

NOAA H

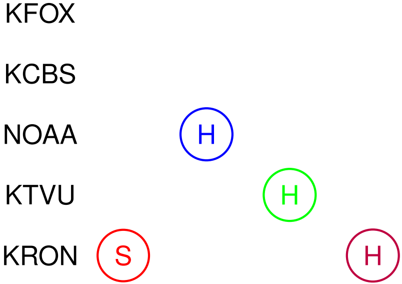
KTVU

KRON S

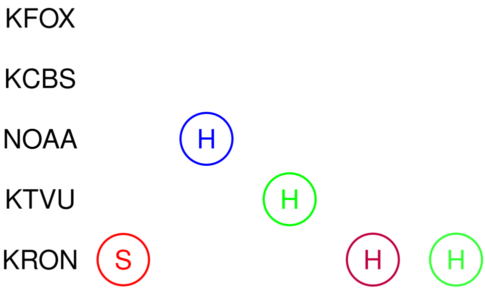
Today: The multiplicative weights algorithm.

KFOX			
KCBS			
NOAA		H	
KTVU			H
KRON	S		

Today: The multiplicative weights algorithm.



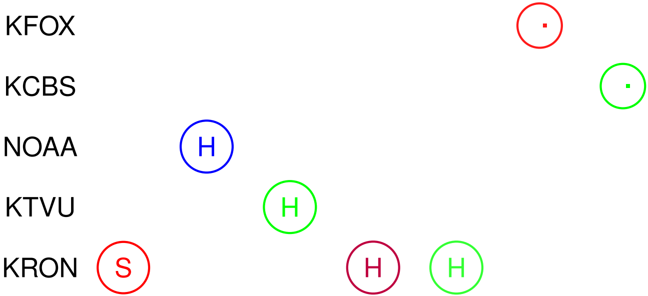
Today: The multiplicative weights algorithm.



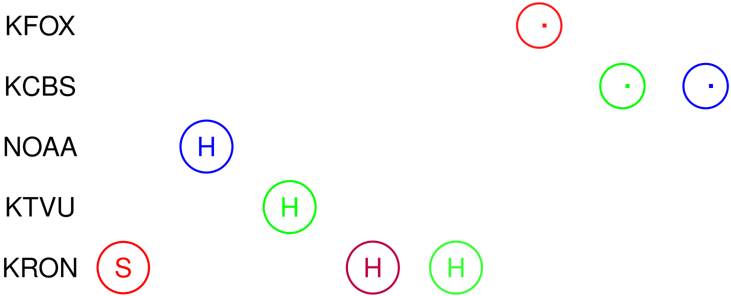
Today: The multiplicative weights algorithm.

KFOX						·
KCBS						
NOAA		H				
KTVU			H			
KRON	S			H	H	

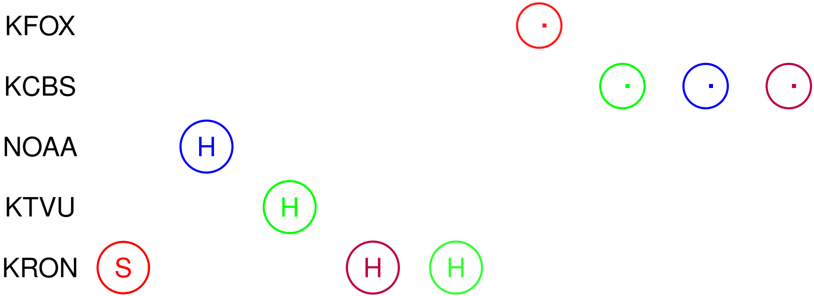
Today: The multiplicative weights algorithm.



Today: The multiplicative weights algorithm.

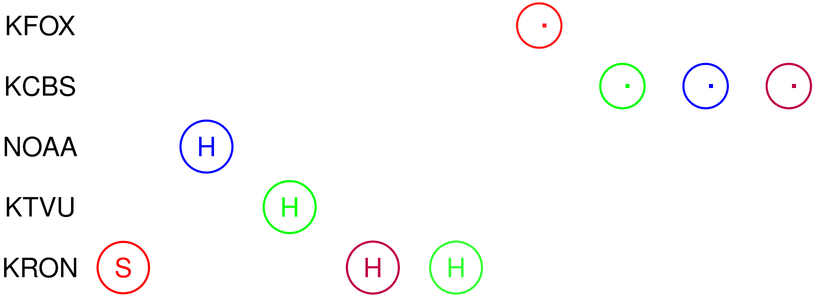


Today: The multiplicative weights algorithm.



If periods indicate good.

Today: The multiplicative weights algorithm.



If periods indicate good.

Which Channel?

Lecture in a Minute

Multiplicative Weights Algorithm.
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: $(1+\varepsilon)L^* + \frac{\log n}{\varepsilon}$

Lecture in a Minute

Multiplicative Weights Algorithm.

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:

Lecture in a Minute

Multiplicative Weights Algorithm.

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

Lecture in a Minute

Multiplicative Weights Algorithm.

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

Lecture in a Minute

Multiplicative Weights Algorithm.
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! Algorithm appears in many settings!
Applications next!

Learning.

Which stock do you buy?

Learning.

Which stock do you buy?

Which weather station is most accurate?

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Today:

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Today: Do what seems to work!

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Today: Do what seems to work!

Softly.

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Today: Do what seems to work!

Softly.
Avoid a little what doesn't work.

Learning.

Which stock do you buy?

Which weather station is most accurate?

Which road do you take?

How should I behave?

Today: Do what seems to work!

Softly.

Avoid a little what doesn't work.

Do something a little more that does.

Experts framework.

n experts.

Experts framework.

n experts.

Every day, each offers a prediction.

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1				$\cdots$	
Expert 2				$\cdots$	
Expert 3				$\cdots$	
$\vdots$				$\cdots$	

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine			$\cdots$	
Expert 2	Shine			$\cdots$	
Expert 3	Rain			$\cdots$	
$\vdots$	$\vdots$			$\cdots$	

Rained!

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain		$\cdots$	
Expert 2	Shine	Shine		$\cdots$	
Expert 3	Rain	Rain		$\cdots$	
$\vdots$	$\vdots$	$\vdots$		$\cdots$	

Rained! Shined!

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	

Rained! Shined! Shined!

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	
	Rained!	Shined!	Shined!	$\cdots$	

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	

Rained! Shined! Shined! $\cdots$

Whose advice do you follow?

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	

Rained! Shined! Shined! $\cdots$

Whose advice do you follow?

"The one who is correct most often."

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	

Rained! Shined! Shined! $\cdots$

Whose advice do you follow?

"The one who is correct most often."

Sort of.

Experts framework.

n experts.

Every day, each offers a prediction.

"Rain" or "Shine."

	Day 1	Day 2	Day 3	$\cdots$	Day T
Expert 1	Shine	Rain	Shine	$\cdots$	
Expert 2	Shine	Shine	Shine	$\cdots$	
Expert 3	Rain	Rain	Rain	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	Shine	$\cdots$	

Rained! Shined! Shined! $\cdots$

Whose advice do you follow?

"The one who is correct most often."

Sort of.

How well do you do?

Infallible expert.

One of the experts is infallible!

Infallible expert.

One of the experts is infallible!

Your strategy?

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never!

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make?

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? Mistake Bound.

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? Mistake Bound.

(A) 1

(B) 2

(C) $\log n$

(D) $n-1$

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

Infallible expert.

One of the experts is infallible!

Your strategy?

Choose any expert that has not made a mistake!

How long to find perfect expert?

Maybe..never! Never see a mistake.

Better model?

How many mistakes could you make? Mistake Bound.

(A) 1

(B) 2

(C) $\log n$

(D) $n-1$

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

$n-1$!

Regret!

Note.

Regret!

Note.

Adversary:

Regret!

Note.

Adversary:
makes you want to look bad.

Regret!

Note.

Adversary:

makes you want to look bad.

”You could have done so well”...

Regret!

Note.

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

Regret!

Note.

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

Regret!

Note.

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

Regret!

Note.

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

Regret!

Note.

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

Regret!

Note.

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

Analysis of Algorithms: do as well as possible!

Regret!

Note.

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

Analysis of Algorithms: do as well as possible!

Minimize Regret:

Regret!

Note.

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

Analysis of Algorithms: do as well as possible!

Minimize Regret:
Regret $\equiv$ Difference between Loss/Gain compared to the best.

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. $w_i \to w_i/2$ if wrong.

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

Potential function:

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

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

Goal: Best expert makes m mistakes.

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

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

-1? -2?

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

-1? -2? factor of $\frac{1}{2}$?

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

-1? -2? factor of $\frac{1}{2}$?

each incorrect expert weight multiplied by $\frac{1}{2}$!

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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 $\frac{3}{4}$?

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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 $\to \geq$ half weight with incorrect experts.

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

-1? -2? factor of $\frac{1}{2}$?

each incorrect expert weight multiplied by $\frac{1}{2}$!

total weight decreases by

factor of $\frac{1}{2}$? factor of $\frac{3}{4}$?

mistake $\rightarrow \geq$ half weight with incorrect experts.

($\geq \frac{1}{2}$ total.) $\times 1/2$

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

Each mistake:

total weight of incorrect experts reduced by

-1? -2? factor of $\frac{1}{2}$?

each incorrect expert weight multiplied by $\frac{1}{2}$!

total weight decreases by

factor of $\frac{1}{2}$? factor of $\frac{3}{4}$?

mistake $\rightarrow \geq$ half weight with incorrect experts.

($\geq \frac{1}{2}$ total.) $\times 1/2$

13 So reduction by $\geq 1/4$.

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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 $\to \geq$ half weight with incorrect experts.

($\geq \frac{1}{2}$ total.) $\times 1/2$

13 So reduction by $\geq 1/4$.

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

Analysis: weighted majority

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

Goal: Best expert makes m mistakes.

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

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

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 $\to \geq$ half weight with incorrect experts.
($\geq \frac{1}{2}$ total.) $\times 1/2$
13 So reduction by $\geq 1/4$.

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

We have

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

where M is number of algorithm mistakes.

Analysis: continued.

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

Analysis: continued.

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

m - best expert mistakes

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.

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

Analysis: continued.

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

m - best expert mistakes M - algorithm mistakes.

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

Take log of both sides.

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

Solve for M.

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

Analysis: continued.

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

m - best expert mistakes M - algorithm mistakes.

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

Take log of both sides.

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

Solve for M.

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

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

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

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \leq \ln((\left(1-\frac{\varepsilon}{2}\right)^M n) \implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2)+\ln n$$

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\begin{aligned}\ln(1-\varepsilon)^m \leq \ln(\left(\left(1-\tfrac{\varepsilon}{2}\right)^M n\right) &\implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2) + \ln n \\ &\implies (-\varepsilon-\varepsilon^2)m \leq M(-\varepsilon/2) + \ln n\end{aligned}$$

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \le \ln(\left(\left(1-\tfrac{\varepsilon}{2}\right)^M n\right) \implies m\ln(1-\varepsilon) \le M\ln(1-\varepsilon/2) + \ln n$$

$$\implies (-\varepsilon-\varepsilon^2)m \le M(-\varepsilon/2) + \ln n$$

$$\implies -\varepsilon(1+\varepsilon)m \le M(-\varepsilon/2) + \ln n$$

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \leq \ln(\left(\left(1-\frac{\varepsilon}{2}\right)^M n\right) \implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2) + \ln n$$
$$\implies (-\varepsilon - \varepsilon^2)m \leq M(-\varepsilon/2) + \ln n$$
$$\implies -\varepsilon(1+\varepsilon)m \leq M(-\varepsilon/2) + \ln n \implies 2(1+\varepsilon)m \geq M - \ln n$$

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \leq \ln(\left(\left(1-\frac{\varepsilon}{2}\right)^M n\right) \implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2) + \ln n$$
$$\implies (-\varepsilon-\varepsilon^2)m \leq M(-\varepsilon/2) + \ln n$$
$$\implies -\varepsilon(1+\varepsilon)m \leq M(-\varepsilon/2) + \ln n \implies 2(1+\varepsilon)m \geq M - \ln n$$

Penultimate step from: $-\varepsilon-\varepsilon^2 \leq \ln(1-\varepsilon) \leq -\varepsilon$, $\varepsilon \leq 1/2$

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \leq \ln(\left(\left(1-\tfrac{\varepsilon}{2}\right)^M n\right) \implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2)+\ln n$$
$$\implies (-\varepsilon-\varepsilon^2)m \leq M(-\varepsilon/2)+\ln n$$
$$\implies -\varepsilon(1+\varepsilon)m \leq M(-\varepsilon/2)+\ln n \implies 2(1+\varepsilon)m \geq M-\ln n$$

Penultimate step from: $-\varepsilon-\varepsilon^2 \leq \ln(1-\varepsilon) \leq -\varepsilon$, $\varepsilon \leq 1/2$

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

Analysis: $(1-\varepsilon)$ penalty

Algorithm: Multiply by $1-\varepsilon$ for incorrect experts...

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

Massage...

$$\ln(1-\varepsilon)^m \leq \ln(\left(\left(1-\tfrac{\varepsilon}{2}\right)^M n\right) \implies m\ln(1-\varepsilon) \leq M\ln(1-\varepsilon/2)+\ln n$$
$$\implies (-\varepsilon-\varepsilon^2)m \leq M(-\varepsilon/2)+\ln n$$
$$\implies -\varepsilon(1+\varepsilon)m \leq M(-\varepsilon/2)+\ln n \implies 2(1+\varepsilon)m \geq M-\ln n$$

Penultimate step from: $-\varepsilon-\varepsilon^2 \leq \ln(1-\varepsilon) \leq -\varepsilon$, $\varepsilon \leq 1/2$

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

Approaches a factor of two of best expert performance as $m \to \infty$.

Best Analysis?

Best Analysis?

Consider two experts: A,B

Best Analysis?

Consider two experts: A,B

Adversary: A correct even days, B correct odd days

Best Analysis?

Consider two experts: A,B

Adversary: A correct even days, B correct odd days

Best expert peformance: $T/2$ mistakes.

Best Analysis?

Consider two experts: A,B

Adversary: A correct even days, B correct odd days

Best expert peformance: $T/2$ mistakes.

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

Best Analysis?

Consider two experts: A,B

Adversary: A correct even days, B correct odd days

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.

Roughly

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.

Roughly optimal!

Randomized analysis.

Some formulas:

Randomized analysis.

Some formulas:

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

Randomized analysis.

Some formulas:

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

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

Randomized analysis.

Some formulas:

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

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

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

Randomized analysis.

Some formulas:

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

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

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

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

Randomized analysis.

Some formulas:

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

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

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

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

Proof Idea: $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots$

Randomized algorithm

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

Randomized algorithm

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

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

Randomized algorithm

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

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

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

Randomized algorithm

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

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

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

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

Randomized algorithm

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

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

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

Randomized algorithm

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

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

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

Best expert, b, loses L^* total.

Randomized algorithm

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

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

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

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

Randomized algorithm

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

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

$W(t)$ 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

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: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

Randomized algorithm

Expert i loses $\ell_i^t \in [0,1]$ 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: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

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

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

Randomized algorithm

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

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

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

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

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

Claim: For $\varepsilon \leq 1/2$, $W(t+1) \leq W(t)(1-\varepsilon L_t)$ Loss $\rightarrow$ weight loss.

Proof:

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

$\square$

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

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

Take logs

$$(L^*)\ln(1-\varepsilon) \le \ln n + \sum \ln(1-\varepsilon L_t)$$

Use $-\varepsilon - \varepsilon^2 \le \ln(1-\varepsilon) \le -\varepsilon$

$$-(L^*)(\varepsilon+\varepsilon^2) \le \ln n - \varepsilon \sum L_t$$

And

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

$\sum_t L_t$ 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 $[0,\rho]$.

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

Other Models

What if you only know the advice of the expert you pick?

Other Models

What if you only know the advice of the expert you pick?

Bandit Problems.

Other Models

What if you only know the advice of the expert you pick?

- Bandit Problems.
  - Model for studying exploration versus exploitation.

Other Models

What if you only know the advice of the expert you pick?

- Bandit Problems.
  - Model for studying exploration versus exploitation.
- Reinforcement Learning.

Other Models

What if you only know the advice of the expert you pick?

- Bandit Problems.
  - Model for studying exploration versus exploitation.
- Reinforcement Learning.
  - Get reward with certain action.

Other Models

What if you only know the advice of the expert you pick?

Bandit Problems.

Model for studying exploration versus exploitation.

Reinforcement Learning.

Get reward with certain action.

Don't know anything about action not taken.

Other Models

What if you only know the advice of the expert you pick?

- Bandit Problems.
  - Model for studying exploration versus exploitation.
- Reinforcement Learning.
  - Get reward with certain action.
  - Don't know anything about action not taken.

Algorithms for Reinforcement Learning.

Other Models

What if you only know the advice of the expert you pick?

- Bandit Problems.
  - Model for studying exploration versus exploitation.
- Reinforcement Learning.
  - Get reward with certain action.
  - Don't know anything about action not taken.

Algorithms for Reinforcement Learning.

Same as the Bandit model in Learning with the Regret Framework.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Linear combination of gradients linear lower bound on $f(x)$.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Linear combination of gradients linear lower bound on $f(x)$.

Expert's says average value of loss is low relative to lower bound,

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Linear combination of gradients linear lower bound on $f(x)$.

Expert's says average value of loss is low relative to lower bound,
$\implies$ average difference of $f(x_t)$ from $f(x^*)$ is low.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Linear combination of gradients linear lower bound on $f(x)$.

Expert's says average value of loss is low relative to lower bound,
$\implies$ average difference of $f(x_t)$ from $f(x^*)$ is low.

Expert's algorithm is a dual version of gradient descent.

Optimization (Online and offline).

Minimize a convex function $f(x)$. $x^* \in R^n$.

x^* minimizes $f(x)$.

Gradients: $\nabla f(\cdot)$.

Loss on experts corresponding to coordinates x_i.

Linear combination of gradients linear lower bound on $f(x)$.

Expert's says average value of loss is low relative to lower bound,
$\implies$ average difference of $f(x_t)$ from $f(x^*)$ is low.

Expert's algorithm is a dual version of gradient descent.

Online: works with just local gradients.

Lecture in a Minute

Multiplicative Weights Algorithm.
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: $(1+\varepsilon)L^* + \frac{\log n}{\varepsilon}$

Lecture in a Minute

Multiplicative Weights Algorithm.
  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:

Lecture in a Minute

Multiplicative Weights Algorithm.

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

Lecture in a Minute

Multiplicative Weights Algorithm.

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

Lecture in a Minute

Multiplicative Weights Algorithm.

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! Algorithm appears in many settings!

Applications next!