Comments on last lecture.

Easy to come up with several Nash for non-zero-sum games.
Is the game framework only interesting in some infinite horizon?
No.
Minimize worst expected loss. Best defense.
Any prior distribution on opponent. Best offense.
Rational players should play this way!
“Infinite horizon” is just an assumption of rationality.

Today

Finish Maximum Weight Matching Algorithm.
Exact algorithm with dueling players.
Multiplicative Weights Framework.
Very general framework of toll/congestion algorithm.

Some details/Runtime

Add 0 value edges, so that optimal solution contains perfect matching.
Beginning “Matcher” Solution: \( M = \{ \} \).
Feasible! Value = 0.
Beginning “Coverer” Solution: \( p(u) = \text{maximum incident edge for } u \in U, 0 \text{ otherwise.} \)
Main Work:
- breadth first search from unmatched nodes finds cut.
Update prices (find minimum delta.)
Simple Implementation:
- Each bfs either augments or adds node to \( S \) in next cut.
- \( O(n) \) iterations per augmentation.
- \( O(n) \) augmentations.
- \( O(n^2m) \) time.

Matching/Weighted Vertex Cover

Maximum Weight Matching.
Given a bipartite graph, \( G = (U, V, E) \), with edge weights \( w : E \rightarrow \mathbb{R}^+ \), find a maximum weight matching.
A matching is a set of edges where no two share an endpoint.

Minimum Weight Cover.
Given a bipartite graph, \( G = (U, V, E) \), with edge weights \( w : E \rightarrow \mathbb{R}^+ \), find an vertex cover function of minimum total value.
A function \( p : V \rightarrow \mathbb{R}^+ \) for all edges, \( e = (u, v) \).
\( p(u) + p(v) \geq w(e) \).
Minimize \( \sum_{v \in U \cup V} p(u) \).
Optimal solutions to both if
for \( e \in M \), \( w(e) = p(u) + p(v) \) (Defn: tight edge.) and perfect matching.

Example

Weight legend:
Black 1, green 2, blue 3.
Tight edges for initial prices.
Tight edges for final prices.
No augmenting path. Red edges.
No augmenting path. Red edges.
Red edges reachable! \( (S, T) \)
Blue edges minimally non-tight.
New augmenting path. Blue edges minimally non-tight.
Blue edges minimally non-tight.
Some more tight edges.
And \( S \) shows a “new” augmenting edge.
And another augmentation...
...and another augmentation...
...and finally: a perfect matching.

All matched edges tight.
Perfect matching. Feasible price function. Values the same. Optimal!
Notice:
no weights on the right problem.
retain previous matching through price changes.
retains edges in failed search through price changes.
The multiplicative weights framework.

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!
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.
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!

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!

Back to mistake bound.

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 \( \rightarrow \) \( \leq n/2 \) perfect experts
\( \vdots \)
\( \geq 1 \) perfect expert \( \rightarrow \) at most \( \log n \) mistakes!
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

Goal: Best expert makes \( m \) mistakes.
Potential function: \( \sum w_i. \) Initially \( n \).
For best expert, \( b \), \( w_b \geq \frac{1}{m} \).
Each mistake:
- total weight of incorrect experts reduced by \(-\frac{1}{2}\)?
- each incorrect expert weight multiplied by \( \frac{1}{2} \)?
- total weight decreases by factor of \( \frac{1}{2} \)?
- factor of \( \frac{1}{2} \)?
- mistake \( \rightarrow \) half weight with incorrect experts.
Mistake \( \rightarrow \) potential function decreased by \( \frac{1}{2} \).
We have
\[
\frac{1}{2^m} \leq \sum w_i \leq \left( \frac{3}{4} \right)^m n.
\]

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 performance: \( T/2 \) mistakes.
Pattern (A): \( T-1 \) mistakes.
Factor of (almost) two worse!

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

Randomization!!!

Analysis: continued.

\[
\frac{1}{2^m} \leq \sum w_i \leq \left( \frac{3}{4} \right)^m n.
\]
\( m \) - best expert mistakes \( M \) algorithm mistakes.
\[
\frac{1}{2^m} \leq \left( \frac{3}{4} \right)^m M.
\]
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 \( 1 - \epsilon \) for incorrect experts...
\[
(1 - \epsilon)^m \leq (1 - \epsilon^2)^M n.
\]
Massage...
\[
M \leq 2(1 + \epsilon) m + \frac{2m \ln n}{\epsilon}
\]
Approaches a factor of two of best expert performance!

Randomized analysis.

Some formulas:
For \( \epsilon \leq 1, x \in [0, 1] \),
\[
(1 + \epsilon)^x \leq (1 + x \epsilon) \quad (1 - \epsilon)^x \leq (1 - x \epsilon)
\]
For \( \epsilon \in [0, \frac{1}{2}] \),
\[
-x - \epsilon^2 \leq \ln(1 - \epsilon) \leq -x
\]
\[
-x - \epsilon^2 \leq \ln(1 + \epsilon) \leq x
\]
Proof idea: \( \ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots \)
**Randomized Algorithm**

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

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

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

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

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

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

Proof:

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

$$= \sum_i w_i \left(1 - \epsilon \frac{\sum_i w_i \ell_i}{\sum_i w_i}\right)$$

$$= W(t)(1-\epsilon L_t)$$

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: $2(1+\epsilon)m + \frac{\log n}{\epsilon}$

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

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

Strategy:

Choose proportional to weights multiply weight by $(1-\epsilon)$loss.

Multiplicative weights framework!

Applications next!

**Analysis**

$$(1-\epsilon)L^* \leq W(T) \leq n \prod_i (1-\epsilon L_i)$$

Take logs

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

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

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

And

$$\sum L_i \leq (1+\epsilon)L^* + \frac{\ln n}{\epsilon}$$

$\sum L_i$ is total expected loss of algorithm.

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

No factor of 2 loss!

**Gains.**

Why so negative?

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

Multiplicative weights with $(1+\epsilon)g_i$.

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

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

Scaling:

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

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