Application of Multiplicative Weights.

Cheat
Scissors
Paper
Rock

BTW: Only the circles mean anything.
Lecture in a minute

Multiplicative Weights
⇒ strong duality for Zero-Sum Games.
Column player plays MW distribution.
Row player plays best response.
  Output average of column player as $y$.
  Output average of row player as $x$.
MW Alg (Column strategy) →
Close to best response against row.
Row $x$ is best response against $y$.

Boosting: (Extra.) Barely learning ⇒ really good learning.
Alg that predicts $1/2 + \varepsilon$ of input points.
  plus multiplicative weights.
  ⇒ Alg that predicts $1 - \mu$ of input points.
MW Application:
  Expert/Input points lose when alg predicts correctly.
  Adversary every day is learning algorithm.
  Predict majority.
MW analysis ⇒ most points predicted correctly.
Matrix Reminders.

$m \times n$ matrix $A$.

$m$-dimensional vector $x$.

$x^T A$ is $n$-dimensional (column) vector.

Shorthand when clear from context: $xA \equiv x^T A$.

$n$-dimensional vector $y$.

$Ay$ is $m$-dimensional (column) vector.
Two person zero sum games.

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

Row mixed strategy: $x = (x_1, \ldots, x_m)$. $\sum_i x_i = 1$.

Column mixed strategy: $y = (y_1, \ldots, y_n)$. $\sum_i y_i = 1$.

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

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

That is,

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

$x^T A$ is vector of (column) payoffs against row strategy $x$.

$Ay$ is vector of (row) payoffs against column strategy $y$.

Pure strategy plays one row(column) with probability 1.

E.g. $x = [0, 0, 1, \ldots, 0]$.

Recall row maximizes, column minimizes.
Equilibrium.

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

\[ p(x, y) = (x^*)^T Ay^* = \min_y (x^*)^T Ay = \max_x x^T Ay^*. \]

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

No row is better:
\[
\max_i A(i) \cdot y^* = (x^*)^T Ay^*. \tag{1}
\]

No column is better:
\[
\min_j (A^T)(j) \cdot x^* = (x^*)^T Ay^*. \]

\(^1A^{(i)}\) is \(i\)th row.
Equilibrium: play the boss...

\[
A = \begin{bmatrix}
0 & -1 & 1 \\
1 & 0 & -1 \\
-1 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Equilibrium: Row: \((0, \frac{1}{3}, \frac{1}{6}, \frac{1}{2})\). Column: \((\frac{1}{3}, \frac{1}{2}, \frac{1}{6})\).


Row Player.

Strategy 1: \(\frac{1}{3} \times 0 + \frac{1}{2} \times -1 + \frac{1}{6} \times 1 = -\frac{1}{3}\)
Strategy 2: \(\frac{1}{3} \times 1 + \frac{1}{2} \times 0 + \frac{1}{6} \times -1 = \frac{1}{6}\)
Strategy 3: \(\frac{1}{3} \times -1 + \frac{1}{2} \times 1 + \frac{1}{6} \times 0 = \frac{1}{6}\)
Strategy 4: \(\frac{1}{3} \times 0 + \frac{1}{2} \times 0 + \frac{1}{6} \times 1 = \frac{1}{6}\)

Payoff is \(0 \times -\frac{1}{3} + \frac{1}{3} \times \left(\frac{1}{6}\right) + \frac{1}{6} \times \left(\frac{1}{6}\right) + \frac{1}{2} \times \left(\frac{1}{6}\right) = \frac{1}{6}\)

Column player: every column payoff is \(\frac{1}{6}\).

Both only play optimal strategies! Complementary slackness.

Why play more than one? Limit opponent payoff!
Equilibrium pair: \((x^*, y^*)\)?

\[ \rho(x, y) = (x^*)^T Ay^* = \min_y (x^*)^T Ay = \max_x x^T Ay^*. \]

Does an equilibrium pair: \((x^*, y^*)\), exist?
Best Response

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

$$ R = \min_y \max_x (x^T Ay). $$

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 low.

$$ C = \max_x \min_y (x^T Ay). $$

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

Example: Roshambo. Value of $C$?
Duality.

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

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

**Proof:** Better to go second.

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

Equilibrium \(\implies R = C!\)

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

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

Sort of.

Approximate equilibrium ...

\[ C(x) = \min_y x^T Ay \]

\[ R(y) = \max_x x^T Ay \]

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

Equilibrium: \((x, y)\)

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

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

With \(R(y) > C(x)\) (weak duality)

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

How?

(A) Using geometry.
(B) Using a fixed point theorem.
(C) Using multiplicative weights.

Not hard. Even easy. Still, head scratching happens.
Games and experts

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

\[
(\max_x xAy^*) - (\min_y x^*Ay) \leq \epsilon
\]

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

Experts Framework:

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

Multiplicative Weights Method yields loss \(L\) where

\[
L \leq (1 + \epsilon)L^* + \frac{\log n}{\epsilon}
\]
Assume: $A$ has payoffs in $[0, 1]$.

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>$t = 1$</th>
<th>$t = 2$</th>
<th>$t = 3$</th>
<th>$t = 4$</th>
<th>$\cdots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$y_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$y_m$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>

$x$-player (row)

$\propto w^t$ mult. weights

$y$-player (col)
Approximate Equilibrium!

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

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

Claim: $(x^*, y^*)$ are $2\varepsilon$-optimal for matrix $A$ for $T = \frac{\ln n}{\varepsilon^2}$.

Row payoff: $R(y^*) = \max_x xAy^*$.

Loss on day $t$, $x_t Ay_t \geq R(y^*)$ by the choice of $y^*$. Thus, algorithm loss, $L$, is $\geq T \times R(y^*)$.

Best expert: $L^*$ - best column against the row distributions played.

Best column against $\sum_t x_t A$ and $T \times x^* = \sum_t x_t$

$\rightarrow$ best column against $T \times x^* A$.

$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

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

$\rightarrow R(y^*) - C(x^*) \leq 2\varepsilon.$
Wrapping up duality theorem.

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Other proofs: use geometry, linear programming.

Complexity?

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

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.


Also works with both using multiplicative weights.

“In practice.”
Boosting.

This is for fun. Not testable.
Boosting...

Get labelled dataset.
Learn model 1.
Take points previous models don’t learn well.
Learn model 2.
Repeat.
Combine models 1, … $n$, for better predictor?
How? Majority Vote.
How to analyse?
Boosting Example.

Learning just a bit.

Example: set of labelled points, find hyperplane that separates.

- - + +
- + + +
+ - - +
- + - -

Looks hard.

1/2 of them? Easy.
Arbitrary line. And Scan.

Useless. A bit more than 1/2 Correct would be better.
Weak Learner: Classify $\geq \frac{1}{2} + \varepsilon$ points correctly.
Not really important but ...
Weak Learner/Strong Learner

Input: $n$ labelled points.

Weak Learner:
   produce hypothesis correctly classifies $\frac{1}{2} + \varepsilon$ fraction

Strong Learner:
   produce hypothesis correctly classifies $1 - \mu$ fraction

Same thing?

Can one use weak learning to produce strong learner?

Boosting: use a weak learner to produce strong learner.
Poll.

Given a weak learning method (produce ok hypotheses.) produce a great hypothesis.

Can we do this?

(A) Yes
(B) No

If yes. How?

Multiplicative Weights!

The endpoint to a line of research.
Experts are points. ‘Adversary’ is weak learner.
Points want to be misclassified.
Learner wants to maximize probability of classifying random point correctly.
Strong learner algorithm will come from adversary.
Do $T = \frac{2}{\gamma^2} \ln \frac{1}{\mu}$ rounds

1. Row player: multiplicative weights $(1 - \varepsilon)$ on points.
2. Column: run weak learner on row distribution.
3. Hypothesis $h(x)$: majority of $h_1(x), h_2(x), \ldots, h_T(x)$.

Claim: $h(x)$ is correct on $1 - \mu$ of the points !!!
Cool!
Really? Proof?
Adaboost proof.

**Claim:** $h(x)$ is correct on $1 - \mu$ of the points!

Let $S_{bad}$ be the set of points where $h(x)$ is incorrect.

majority of $h_t(x)$ are wrong for $x \in S_{bad}$.

Consider $W(t) = \sum_i w_i^t$.

$x \in S_{bad}$ is a good expert – loses less than $\frac{1}{2}$ the time.

$$W(T) \geq (1 - \epsilon)^{\frac{T}{2}} |S_{bad}|$$

Each day, weak learner gets $\geq \frac{1}{2} + \gamma$ payoff.

$\rightarrow L_t \geq \frac{1}{2} + \gamma$.  

$$W(t+1) \leq (\frac{1}{2} - \gamma) W(t) + (\frac{1}{2} + \gamma)(1 - \epsilon)W(t)$$
$$\leq W(t)(1 - \epsilon(\frac{1}{2} + \gamma)) \leq W(t)e^{-\epsilon(\frac{1}{2} + \gamma)}.$$  

$\rightarrow W(T) \leq ne^{-\epsilon(\frac{1}{2}+\gamma)T}$

Combining

$$|S_{bad}|(1 - \epsilon)^{T/2} \leq W(T) \leq ne^{-\epsilon(\frac{1}{2}+\gamma)T}$$
Calculation..

\[ |S_{bad}|(1 - \varepsilon)^{T/2} \leq ne^{-\varepsilon(\frac{1}{2} + \gamma)T} \]

Set \( \varepsilon = \gamma \), take logs.

\[ \ln \left( \frac{|S_{bad}|}{n} \right) + \frac{T}{2} \ln(1 - \gamma) \leq -\gamma T \left( \frac{1}{2} + \gamma \right) \]

Again, \( -\gamma - \gamma^2 \leq \ln(1 - \gamma) \),

\[ \ln \left( \frac{|S_{bad}|}{n} \right) + \frac{T}{2}(-\gamma - \gamma^2) \leq -\gamma T \left( \frac{1}{2} + \gamma \right) \rightarrow \ln \left( \frac{|S_{bad}|}{n} \right) \leq -\frac{\gamma^2 T}{2} \]

And \( T = \frac{2}{\gamma^2} \ln \mu \),

\[ \rightarrow \ln \left( \frac{|S_{bad}|}{n} \right) \leq \ln \mu \rightarrow \frac{|S_{bad}|}{n} \leq \mu. \]

The misclassified set is at most \( \mu \) fraction of all the points.

The hypothesis correctly classifies \( 1 - \mu \) fraction of the points !

**Claim:** Multiplicative weights: \( h(x) \) is correct on \( 1 - \mu \) of the points !
Some details...

Weak learner learns over distributions of points not points.
Make copies of points to simulate distributions.
Standard method in practice for machine learning for combining repeated base learning algorithms.