Application of Multiplicatcative Weights.

Cheat
Scissors
Paper
Rock
Application of Multiplicative Weights.

Cheat
Scissors
Paper
Rock
Application of Multiplicative Weights.

Cheat
Scissors
Paper
Rock

BTW: Only the circles mean anything.
Application of Multiplicative Weights.
Application of Multiplicative Weights.
Application of Multiplicative Weights.
Application of Multiplicative Weights.

Cheat
Scissors
Paper
Rock

BTW: Only the circles mean anything.
Application of Multiplicative Weights.
Application of Multiplicative Weights.
Application of Multiplicative Weights.
Application of Multiplicative Weights.

Cheat
Scissors
Paper
Rock

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

Multiplicativc 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 $y$.
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.
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.
Multiplicative Weights

\[ \Rightarrow \text{ strong duality for Zero-Sum Games.} \]
Column player plays MW distribution.
Row player plays best response.
Output average of column player as \( y \).
Output average of row player as \( x \).
\( \text{MW Alg (Column strategy)} \rightarrow \text{Close to best response against row.} \)
Row \( x \) is best response against \( y \).

Boosting: (Extra.) Barely learning \( \Rightarrow \) really good learning.
Alg that predicts \( \frac{1}{2} + \varepsilon \) of input points.
plus multiplicative weights.
\[ \Rightarrow \text{Alg that predicts} \ 1 - \mu \text{ of input points.} \]

MW Application:
Expert/Input points lose when alg predicts correctly.
Adversary every day is learning algorithm.
Predict majority.
MW analysis \( \Rightarrow \text{most points predicted correctly.} \)
Matrix Reminders.

$\mathbf{A}$ is an $m \times n$ matrix.
Matrix Reminders.

$m \times n$ matrix $A$.

$m$-dimensional vector $x$.

$n$-dimensional vector $y$.

$Ay$ is $m$-dimensional (column) vector.
Matrix Reminders.

$m \times n$ matrix $A$.

$m$-dimensional vector $x$.

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

$m \times n$ matrix $A$.

$m$-dimensional vector $x$.

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

Shorthand when clear from context: $xA \equiv x^T A$. 
$m \times n$ 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 $y$. 
$m \times n$ 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 $(x, y)$:
$p(x, y) = x^T A y$.

That is,
$\sum_i x_i \left( \sum_j a_{ij} y_j \right) = \sum_j \left( \sum_i x_i a_{ij} \right) y_j$.

$x^T A$ 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.
Two person zero sum games.

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

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

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

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

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

$x^T A$ 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.
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: $y = (y_1, \ldots, y_n)$. $\sum_i y_i = 1$.
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 $(x, y)$:
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 $(x, y)$:

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

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

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: $x = (x_1, \ldots, x_m). \sum_i x_i = 1.$
Column mixed strategy: $y = (y_1, \ldots, y_n). \sum_i y_i = 1.$

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

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

That is,

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

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

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

That is,

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

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

$Ay$ is vector of (row) payoffs against column strategy $y$. 
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.
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. $x = [0, 0, 1, \ldots, 0]$. 
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 $(x, y)$:

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

That is,

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

$x^T A$ 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: \((x^*, y^*)\)?

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

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

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

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

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

\[
\max_i A(i) \cdot y^* = (x^*)^T Ay^*. \tag{1}
\]

\[A^{(i)}\] is \(i\)th row.
Equilibrium pair: \((x^*, y^*)\)?

\[
p(x, y) = (x^*)^T A y^* = \min_{y} (x^*)^T A y = \max_{x} x^T A y^*.
\]

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

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

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

\(A^{(i)}\) 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:
Equilibrium: play the boss...

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

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

Payoff?
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: \((\frac{1}{3}, \frac{1}{2}, \frac{1}{6})\).

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

Both only play optimal strategies!

Complementary slackness.

Why play more than one?

Limit opponent payoff!
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: \(\frac{1}{3} \times -1 + \frac{1}{2} \times 1 + \frac{1}{6} \times 0 = \frac{1}{6}\)
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: \(\frac{1}{3} \times 0 + \frac{1}{2} \times 0 + \frac{1}{6} \times 1\)
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: \(\frac{1}{3} \times 0 + \frac{1}{2} \times 0 + \frac{1}{6} \times 1 = \frac{1}{6}\)
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: \(\frac{1}{3} \times 0 + \frac{1}{2} \times 0 + \frac{1}{6} \times 1 = \frac{1}{6}\)
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 \(0 \times -\frac{1}{3} + \frac{1}{3} \times \left(\frac{1}{6}\right) + \frac{1}{6} \times \left(\frac{1}{6}\right) + \frac{1}{2} \times \left(\frac{1}{6}\right)\)
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 $$0 \times -\frac{1}{3} + \frac{1}{3} \times (\frac{1}{6}) + \frac{1}{6} \times (\frac{1}{6}) + \frac{1}{2} \times (\frac{1}{6}) = \frac{1}{6}$$
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 \(0 \times -\frac{1}{3} + \frac{1}{3} \times (\frac{1}{6}) + \frac{1}{6} \times (\frac{1}{6}) + \frac{1}{2} \times (\frac{1}{6}) = \frac{1}{6}\)

Column player: every column payoff is \(\frac{1}{6}\).
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!
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.
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 \(0 \times -\frac{1}{3} + \frac{1}{3} \times (\frac{1}{6}) + \frac{1}{6} \times (\frac{1}{6}) + \frac{1}{2} \times (\frac{1}{6}) = \frac{1}{6}\)

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

Both only play optimal strategies! Complementary slackness.
Why play more than one?
Equilibrium: play the boss...

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

Equilibrium: Row: \((0, \frac{1}{3}, \frac{1}{6}, \frac{1}{6})\). 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: \((x^*, y^*)\)?

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

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

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

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

Column goes first:

Find $y$, where best row is not too high.

$$R = \min_y x^T A y$$

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

Example: Roshambo.

Value of $R$?

Row goes first:

Find $x$, where best column is not low.

$$C = \max_x y \min_y x^T A y$$

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

Example: Roshambo.

Value of $C$?
Best Response

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

$$R = \min_{y} \max_{x} (x^T Ay).$$
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.

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

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

Example: Roshambo.

**Value of $C$?**
Best Response

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

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

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

Example: Roshambo.
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 $R$?
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 $(0,0,\ldots,1,\ldots0)$.

Example: Roshambo. Value of $R$?

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

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

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

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 low.

$$C = \max_x \min_y (x^T A y).$$
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 $(0,0,\ldots,1,\ldots0)$.

Example: Roshambo. Value of $R$?

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

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

Agin: $y$ of form $(0,0,\ldots,1,\ldots0)$. 
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 $(0,0,...,1,...,0)$.
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 $(0,0,...,1,...,0)$.
Example: Roshambo.
Best Response

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

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

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 low.

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

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

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

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

Column payoffs \((x^* A)\) all \(\geq v = \Rightarrow v \geq C\).

\(\Rightarrow R \leq C\)

Equilibrium \(\Rightarrow R = C\)!

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

Doesn't matter who plays first!
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.
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 \( v \):
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 \(\leq v\)
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 \(\leq v\) \(\implies\) \(R \leq v.\)
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 \( \geq v \)
Duality.

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

**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 \( \geq v \) \( \implies \) \( v \geq C \).
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 \(\geq v \implies v \geq C.\)
\(\implies R \leq C\)
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 \( \implies R = C! \)
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!
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 \( R = C! \)
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.
Proof of Equilibrium.

Sort of.

Approximate equilibrium ...
Proof of Equilibrium.

Sort of.

Approximate equilibrium ...

\[ C(x) = \min_y x^T Ay \]
Proof of Equilibrium.

Sort of.

Approximate equilibrium ...

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

\[ R(y) = \max_x x^T Ay \]
Proof of Equilibrium.

Sort of.

Approximate equilibrium ...

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

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

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

\[ R(y) = C(x) \]
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: \((x, y)\)

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

Sort of.

Approximate equilibrium ...

\[ C(x) = \min_y x^T A y \]

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

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

Equilibrium: \((x, y)\)

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

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

\[ \rightarrow \text{“Response } y \text{ to } x \text{ is within } \varepsilon \text{ of best response”} \]
Proof of Equilibrium.

Sort of.

Aproximate 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)

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

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

Sort of.

Aproximate 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)

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

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

How?

(A) Using geometry.
Proof of approximate equilibrium.

How?

(A) Using geometry.

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

How?

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

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

How?

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

How?

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

(C) Not hard.
Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.

(C) Using multiplicative weights.

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

How?

(A) Using geometry.

(B) Using a fixed point theorem.

(C) Using multiplicative weights.

(C)
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
\[
(\max_x xAy^*) - (\min_y x^*Ay) \leq \varepsilon
\]
Games and experts

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

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

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

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

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

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

\[
\left( \max_x x A y^* \right) - \left( \min_y x^* A y \right) \leq \varepsilon \\
R(x^*) - C(y^*) \leq \varepsilon
\]
Games and experts

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

\[(\max_x xA y^*) - (\min_y x^*Ay) \leq \varepsilon\]

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

Experts Framework:

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

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

\[
\begin{align*}
& (\max_x x A y^*) - (\min_y x^* A y) \leq \varepsilon \\
& R(x^*) - C(y^*) \leq \varepsilon
\end{align*}
\]

Experts Framework:

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

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

\[
(\max_x xA^*y^*) - (\min_y x^*Ay) \leq \varepsilon
\]
\[
R(x^*) - C(y^*) \leq \varepsilon
\]

Experts Framework:

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

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

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

\[
\left( \max_x xAy^* \right) - \left( \min_y x^* Ay \right) \leq \varepsilon
\]

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

Experts Framework:

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

Multiplicative Weights Method yields loss \(L\) where

\[
L \leq (1 + \varepsilon) L^* + \frac{\log n}{\varepsilon}
\]
Assume:

$A$ has payoffs in $[0, 1]$. For $T = \log n \epsilon 2$ 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.
Assume: $A$ has payoffs in $[0, 1]$. 

### Games and Experts.

[...content related to 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 column strategies are experts.
Games and Experts.

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

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

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

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

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

For $T = \frac{\log n}{\varepsilon^2}$ 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 $y_t$. 
Assume: $A$ has payoffs in $[0,1]$.  
For $T = \frac{\log n}{\varepsilon^2}$ 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.
Assume: $A$ has payoffs in $[0, 1]$.

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

\begin{align*}
\text{x-player (row)} \\
&t = 1 \quad t = 2 \quad t = 3 \quad t = 4 \quad \cdots \\
y_1 \\
y_2 \\
&\propto w^t \text{ mult. weights} \\
&\vdots \\
y_m
\end{align*}
### x-player (row)

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

### y-player (col)

$\propto w^t$ mult. weights
\begin{align*}
\text{x-player (row)} \\
\begin{array}{c|cc}
 t & 1 & 2 \\
\hline
1 & 1 & 0 \\
2 & 0 & 0 \\
\vdots & \vdots & \vdots \\
 m & 0 & 1 \\
\end{array}
\end{align*}

\text{y-player (col)} \\
\begin{itemize}
\item \( \propto w^t \) mult. weights
\end{itemize}
Picture of Algorithm.

\[ \begin{align*}
\text{x-player (row)} & \\
& \begin{array}{cccc}
 t = 1 & t = 2 & t = 3 & t = 4 & \cdots \\
y_1 & 1 & 0 & 0 & \\
y_2 & 0 & 0 & 1 & \\
\vdots & \vdots & \vdots & \vdots & \\
y_m & 0 & 1 & 0 & \\
\end{array}
\end{align*} \]

\[ \begin{align*}
\text{y-player (col)} & \\
& \begin{array}{c}
 y_1 \quad 1 \quad 0 \quad 0 \\
y_2 \quad 0 \quad 0 \quad 1 \\
\vdots \quad \vdots \quad \vdots \quad \vdots \\
y_m \quad 0 \quad 1 \quad 0 \\
\end{array}
\end{align*} \]

\[ \begin{align*}
\propto w^t \text{ mult. weights}
\end{align*} \]
Picture of Algorithm.

\[
\text{x-player (row)}
\]

\[
\begin{array}{cccccc}
\text{t} = 1 & \text{t} = 2 & \text{t} = 3 & \text{t} = 4 & \cdots \\
y_1 & 1 & 0 & 0 & 1 \\
y_2 & 0 & 0 & 1 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[
\text{y-player (col)} \propto w^t \text{ mult. weights}
\]
$y$-player (col) \[ \begin{array}{cccccc}
\alpha w^t \text{ mult. weights} \\
\vdots \\
y_m & 0 & 1 & 0 & 0 & 0 & \cdots \\
\end{array} \]

$x$-player (row)

\[
\begin{array}{cccccc}
& t = 1 & t = 2 & t = 3 & t = 4 & \cdots \\
y_1 & 1 & 0 & 0 & 1 & \cdots \\
y_2 & 0 & 0 & 1 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & 0 & 1 & 0 & 0 & \cdots \\
\end{array}
\]
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 \( y^* = \arg\min_{y_t} x_t A y_t \).

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

Row payoff: \( R(y^*) = \max_{x} x A y^* \).

Loss on day \( t \), \( x_t A y_t \geq R(y^*) \) 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.

\( \rightarrow \) \( L^* \leq T \times C(x^*) = T \times \min_{y} x^* A y \).

Multiplicative Weights: \( L \leq (1 + \varepsilon) L^* + \ln n \varepsilon T \times R(y^*) \leq (1 + \varepsilon) T \times C(x^*) + \ln n \varepsilon T \rightarrow R(y^*) \leq (1 + \varepsilon) C(x^*) + \ln n \varepsilon T \rightarrow R(y^*) - C(x^*) \leq 2\varepsilon C(y^*) + \ln n \varepsilon T \)
Approximate Equilibrium!

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

Let \( x^* = \frac{1}{T} \sum_t x_t \)
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 $y^* = \arg\min_{y_t} x_t A y_t$. 
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 $T = \frac{\ln n}{\varepsilon^2}$. 
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: $R(y^*) = \max_x xAy^*$.
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ by the choice of $y^*$. 
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 $y^* = \arg\min_y x_tAy_t$.

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 $t$, $x_tA_y \geq R(y^*)$ by the choice of $y^*$.
Thus, algorithm loss, $L$, is $\geq T \times R(y^*)$. 
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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.
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $\sum_t x_t A$
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $T \times x^* = \sum_t x_t$
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 $T \times x^* A$. 
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: $R(y^*) = \max_x x Ay^*$.

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 $T \times x^* A$.

$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$. 
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $T \times x^* = \sum_t x_t$
→ best column against $T \times x^* A$.
→ $L^* \leq T \times C(x^*) = T \times \min_y x^* A y$. 
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 $y^* = \arg\min_y x_t Ay_t$.

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:
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 \( y^* = \text{argmin}_y x_t A y_t \).

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

Row payoff: \( R(y^*) = \max_x x A y^* \).

Loss on day \( t \), \( x_t A y_t \geq R(y^*) \) 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 \( T \times x^* A \).

\( \rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* A y \).

Multiplicative Weights: \( L \leq (1 + \varepsilon)L^* + \frac{\ln n}{\varepsilon} \)
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $T \times x^* A$.

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

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

$T \times R(y^*) \leq (1 + \varepsilon) T \times C(x^*) + \frac{\ln n}{\varepsilon}$
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 $y^* = \text{argmin}_{y_t} x_t Ay_t$.

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

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

Loss on day $t$, $x_tA y_t \geq R(y^*)$ 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 $\sum_t x_tA$ 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: $L \leq (1 + \epsilon)L^* + \frac{\ln n}{\epsilon}$

$T \times R(y^*) \leq (1 + \epsilon) T \times C(x^*) + \frac{\ln n}{\epsilon} \rightarrow R(y^*) \leq (1 + \epsilon) C(x^*) + \frac{\ln n}{\epsilon T}$
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $T \times x^* A$.
- $\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* A y$.

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

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

$\rightarrow R(y^*) - C(x^*) \leq \varepsilon C(y^*) + \frac{\ln n}{\varepsilon T}.$
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 $y^* = \arg\min_{y_t} x_t A y_t$.

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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 $T \times x^* A$.

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

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

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

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

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

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

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

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

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

Loss on day $t$, $x_t A y_t \geq R(y^*)$ 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.

\[ \text{best column against } \sum_t x_t A \text{ and } T \times x^* = \sum_t x_t \rightarrow \text{best column against } T \times x^* A. \]
\[ \rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* A y. \]

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.
Wrapping up duality theorem.

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.
Does an equilibrium exist?
Wrapping up duality theorem.

For any $\epsilon$, there exists an $\epsilon$-Approximate Equilibrium. Does an equilibrium exist? Yes.
Wrapping up duality theorem.

For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.
Does an equilibrium exist? Yes.
Something about math here?

$T = \ln n \varepsilon^2 \rightarrow O(\text{nm log } n \varepsilon^2)$. 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."
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.
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.
Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Other proofs: use geometry, linear programming.

Complexity?
Wrapping up duality theorem.

For any $\epsilon$, there exists an $\epsilon$-Approximate Equilibrium.
Does an equilibrium exist? Yes.
Something about math here? Fixed point theorem.
Other proofs: use geometry, linear programming.
Complexity?
\[ T = \frac{\ln n}{\epsilon^2} \]
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?

\[
T = \frac{\ln n}{\varepsilon^2} \rightarrow O(n m \frac{\log n}{\varepsilon^2}).
\]
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?

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

Basically linear!
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?

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

Versus Linear Programming: $O(n^3m)$
For any $\varepsilon$, there exists an $\varepsilon$-Approximate Equilibrium.

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Other proofs: 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.
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?

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

Basically linear!

Versus Linear Programming: $O(n^3 m)$ Basically quadratic.

(Faster linear programming: $O(\sqrt{n+m})$ linear solution solves.)
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?

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

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

Still much slower
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?

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

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.
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?

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

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.


Also works with both using multiplicative weights.

“In practice.”
Boosting.

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

Get labelled dataset.
Boosting...

Get labelled dataset.
Learn model 1.
Boosting...

Get labelled dataset.
Learn model 1.
Take points previous models don’t learn well.
Boosting...

Get labelled dataset.
Learn model 1.
Take points previous models don’t learn well.
Learn model 2.
Boosting...

Get labelled dataset.
Learn model 1.
Take points previous models don’t learn well.
Learn model 2.
Repeat.
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?
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?
Boosting...

Get labelled dataset.
Learn model 1.
Take points previous models don’t learn well.
Learn model 2.
Repeat.
Combine models 1, \ldots n, for better predictor?
How? Majority Vote.
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.

Useless.

A bit more than 1/2 Correct would be better.

Weak Learner: Classify ≥ 1/2 + ε points correctly.

Not really important but...
Boosting Example.

Learning just a bit.
Example: set of labelled points, find hyperplane that separates.
Boosting Example.

Learning just a bit.

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

```
-  +  +
-  +  +
+   -  -
-  +  -
```

Looks hard.
Boosting Example.

Learning just a bit.

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

\begin{verbatim}
- + +
- + +
+ - -
- + -
\end{verbatim}

Looks hard.

$\frac{1}{2}$ of them?
Boosting Example.

Learning just a bit.

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

\[
\begin{array}{ccc}
- & + & + \\
- & + & + \\
+ & - & - \\
- & + & - \\
\end{array}
\]

Looks hard.

1/2 of them? Easy.
Boosting Example.

Learning just a bit.

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

Looks hard.

1/2 of them? Easy.

Arbitrary line.
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.
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.
Boosting Example.

Learning just a bit.

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

\[ \begin{array}{ccc}
\phantom{-} & + & + \\
- & - & + \\
+ & + & - \\
+ & - & - \\
- & + & \\
\end{array} \]

Looks hard.

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

Learning just a bit.

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

\[+\quad -\quad -\quad +\quad +\]

Looks hard.

1/2 of them? Easy.

Arbitrary line. And Scan.

Useless. A bit more than 1/2 Correct would be better.
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.
Boosting Example.

Learning just a bit.

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

\[
\begin{array}{c|c
   - & + \\
   - & + \\
   + & - \\
   - & - \\
\end{array}
\]

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 ...
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/Strong Learner

Input: \( n \) labelled points.

Weak Learner:
Input: $n$ labelled points.

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

Input: $n$ labelled points.

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

Strong Learner:
Weak Learner/Strong Learner

Input: $n$ labelled points.

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

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

Input: $n$ labelled points.

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

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

Same thing?
Weak Learner/Strong Learner

Input: $n$ labelled points.

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

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

Same thing?

Can one use weak learning to produce strong learner?
Input: $n$ labelled points.

Weak Learner:
produce hypothesis correctly classifies $\frac{1}{2} + \epsilon$ 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.)
Poll.

Given a weak learning method (produce ok hypotheses.)
produce a great hypothesis.
Given a weak learning method (produce ok hypotheses.)
produce a great hypothesis.
Can we do this?
Poll.

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

Can we do this?

(A) Yes

(B) No
Given a weak learning method (produce ok hypotheses.) produce a great hypothesis.

Can we do this?

(A) Yes

(B) No

If yes.
Poll.

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

Can we do this?

(A) Yes

(B) No

If yes. How?
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!
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.
Boosting/MW Framework

Experts are points.
Experts are points. ‘Adversary” is weak learner.
Boosting/MW Framework

Experts are points. ‘Adversary” is weak learner.
Points want to be misclassified.
Experts are points. ‘Adversary” is weak learner.

Points want to be misclassified.

Learner wants to maximize probability
Boosting/MW Framework

Experts are points. ‘Adversary” is weak learner.

Points want to be misclassified.

Learner wants to maximize probability of classifying random point correctly.

Cool!

Really?

Proof?
Boosting/MW Framework

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.
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
Boosting/MW Framework

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.
Boosting/MW Framework

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.
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 $h(x)$:
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 \( h_1(x), h_2(x), \ldots, h_T(x) \).
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
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 \( T = \frac{2}{y^2} \ln \frac{1}{\mu} \) 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!
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!!
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 !!!
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!
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 $(1 - \epsilon)$ 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?
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
Adaboost proof.

Claim: \( h(x) \) is correct on \( 1 - \mu \) of the points!
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.
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 $x \in S_{bad}$.
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 \( W(t) = \sum_i w_i^t \).
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 $h_i(x)$ are wrong for $x \in S_{bad}$.

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

$x \in S_{bad}$ is a good expert.
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 $h_i(x)$ 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.
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.

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

\[
W(T) \geq (1 - \varepsilon)^T |S_{bad}|
\]

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

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

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

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

\[
W(T) \geq (1 - \varepsilon) \frac{T}{2} |S_{bad}|
\]

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

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

\( W(t + 1) \)
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 $h_i(x)$ 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.

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

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

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

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

\[
W(T) \geq (1 - \varepsilon)^\frac{T}{2} |S_{bad}|
\]

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

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

\[
W(t + 1) \leq (\frac{1}{2} - \gamma)W(t) + (\frac{1}{2} + \gamma)(1 - \varepsilon)W(t)
\leq W(t)(1 - \varepsilon(\frac{1}{2} + \gamma))
\]
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. The majority of $h_i(x)$ are wrong for $x \in S_{bad}$.

Consider $W(t) = \sum_i w^t_i$.

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

$$W(T) \geq (1 - \varepsilon)^T |S_{bad}|$$

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

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

$W(t+1) \leq (\frac{1}{2} - \gamma)W(t) + (\frac{1}{2} + \gamma)(1 - \varepsilon)W(t)$

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

\[ W(T) \geq (1 - \varepsilon)^T |S_{bad}| \]

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

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

\[ W(t + 1) \leq (\frac{1}{2} - \gamma)W(t) + (\frac{1}{2} + \gamma)(1 - \varepsilon)W(t) \]

\[ \leq W(t)(1 - \varepsilon(\frac{1}{2} + \gamma)) \leq W(t)e^{-\varepsilon(\frac{1}{2} + \gamma)}. \]

\[ \rightarrow \]
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.

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

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

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

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

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

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

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

$W(t + 1) \leq (\frac{1}{2} - \gamma) W(t) + (\frac{1}{2} + \gamma)(1 - \varepsilon) W(t)$

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

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

Combining
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.

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

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

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

$$W(t + 1) \leq (\frac{1}{2} - \gamma)W(t) + (\frac{1}{2} + \gamma)(1 - \varepsilon)W(t)$$

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

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

Combining

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The hypothesis correctly classifies $1 - \mu$ fraction of the points!
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 \( T = \frac{2}{\gamma^2} \ln \mu \),
\[ \left| S_{bad} \right|(1 - \varepsilon)^{T/2} \leq n e^{-\varepsilon(\frac{1}{2} + \gamma)T} \]

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

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

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

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

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

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.
Calculation..

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

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

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

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

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

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

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

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

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.
Weak learner learns over distributions of points not points.
Make copies of points to simulate distributions.
Some details...

Weak learner learns over distributions of points not points.

Make copies of points to simulate distributions.
Conclusion.

Standard method in practice for machine learning for combining repeated base learning algorithms.