Application of Multiplicative Weights.

Cheat

Scissors

Paper

Rock

Application of Multiplicative Weights.

Cheat

Scissors

Paper

Rock

Application of Multiplicative Weights.

Cheat

Scissors

Paper

Rock S

Application of Multiplicative Weights.

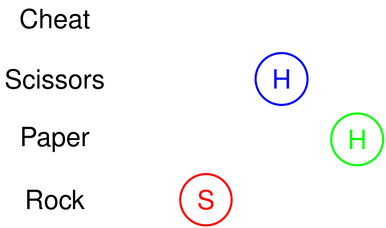
Cheat

Scissors H

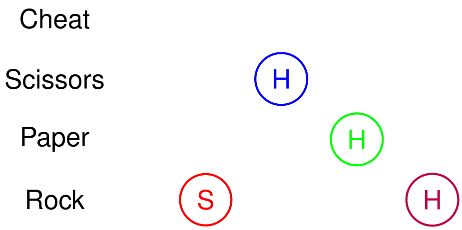
Paper

Rock S

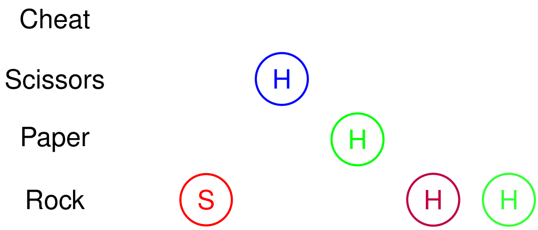
Application of Multiplicative Weights.



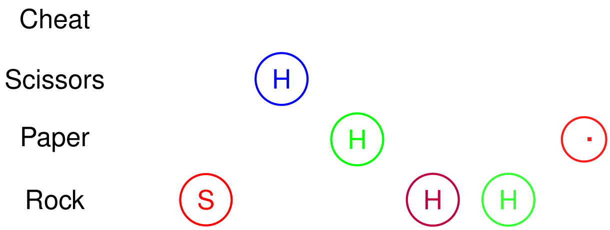
Application of Multiplicative Weights.



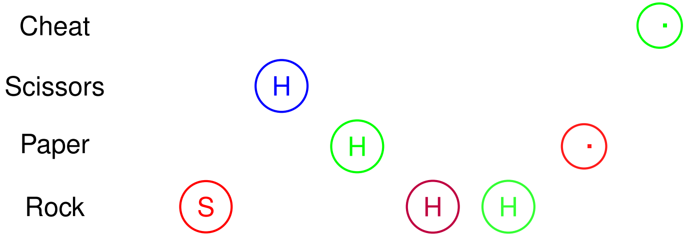
Application of Multiplicative Weights.



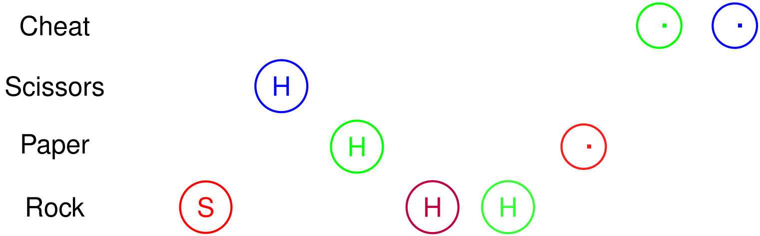
Application of Multiplicative Weights.



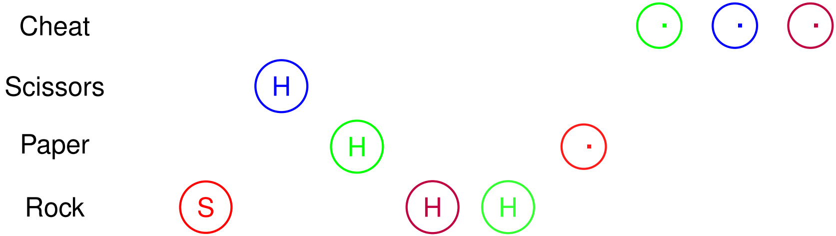
Application of Multiplicative Weights.



Application of Multiplicative Weights.



Application of Multiplicative Weights.



BTW: Only the circles mean anything.

Lecture in a minute

Multiplicative Weights
$\Longrightarrow$ 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) $\rightarrow$
Close to best response against row.
Row x is best response against y.

Lecture in a minute

Multiplicative Weights
$\implies$ 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) $\rightarrow$
Close to best response against row.
Row x is best response against y.

Boosting: (Extra.) Barely learning $\implies$ really good learning.
Alg that predicts $1/2+\varepsilon$ of input points.
plus multiplicative weights.

Lecture in a minute

Multiplicative Weights
$\implies$ 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) $\rightarrow$
Close to best response against row.
Row x is best response against y.

Boosting: (Extra.) Barely learning $\implies$ really good learning.
Alg that predicts $1/2+\varepsilon$ of input points.
plus multiplicative weights.
$\implies$ 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.

Lecture in a minute

Multiplicative Weights
$\implies$ 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) $\rightarrow$
Close to best response against row.
Row x is best response against y.

Boosting: (Extra.) Barely learning $\implies$ really good learning.
Alg that predicts $1/2+\varepsilon$ of input points.
plus multiplicative weights.
$\implies$ 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 $\implies$ most points predicted correctly.

Matrix Reminders.

$m \times n$ matrix A.

Matrix Reminders.

$m \times n$ matrix A.

m-dimensional vector x.

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: $xA \equiv x^T A$.

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.

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.

Two person zero sum games.

$m \times n$ payoff matrix A.

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

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):

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

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):

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

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):

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

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

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^*)?

---

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

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

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

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

Equilibrium.

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

$$p(x,y) = (x^*)^T Ay^* = \min_y (x^*)^T Ay = \max_x \; x^T Ay^*.$$

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

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

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

[1] $A^{(i)}$ is ith 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: $(\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$.

Payoff? Remember: weighted average of pure 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: $(\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$.

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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: $\frac{1}{3} \times -1 + \frac{1}{2} \times 1 + \frac{1}{6} \times 0$

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? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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 $0 \times -\frac{1}{3} + \frac{1}{3} \times (\frac{1}{6}) + \frac{1}{6} \times (\frac{1}{6}) + \frac{1}{2} \times (\frac{1}{6})$

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? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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!

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? Remember: weighted average of pure strategies.

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.

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? Remember: weighted average of pure strategies.

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

Payoff? Remember: weighted average of pure strategies.

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? Limit opponent payoff!

Equilibrium: always?

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

$$p(x,y) = (x^*)^T A y^* = \min_y (x^*)^T A y = \max_x \; x^T A y^*.$$

Equilibrium: always?

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

$$p(x,y) = (x^*)^T A y^* = \min_y (x^*)^T A y = \max_x \; x^T A y^*.$$

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

Best Response

Column goes first:

Best Response

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

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

Best Response

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

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

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

Best Response

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

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

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

Example: Roshambo.

Best Response

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

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.

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 $(0, 0, \ldots, 1, \ldots 0)$.

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 $(0, 0, \ldots, 1, \ldots 0)$.

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.

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.

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

Duality.

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

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. $\square$

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. $\square$

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. $\square$

At Equilibrium (x^*, y^*), payoff v:
row payoffs (Ay^*) all $\leq v$

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. $\square$

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

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. $\square$

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.

$$R = \min_x \max_y (x^T Ay).$$
$$C = \max_y \min_x (x^T Ay).$$

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

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.

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

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

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.

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

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

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.

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

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

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. $\square$

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.

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

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

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.

Aproximate equilibrium ...

Proof of Equilibrium.

Sort of.

Aproximate equilibrium ...

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

Proof of Equilibrium.

Sort of.

Aproximate equilibrium ...

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

$R(y) = \max_x x^T Ay$

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)

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)

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)

$R(y) = C(x)$

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)

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

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

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)

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

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

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

Proof of Equilibrium.

Sort of.

Aproximate equilibrium ...

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

$R(y) = \max_x x^T Ay$

Strategy pair: (x, y)

Equilibrium: (x, y)

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

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

With $R(y) > C(x)$ (weak duality)
$\to$ "Response y to x is within ε 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)
$\rightarrow$ "Response y to x is within ε of best response"
$\rightarrow$ "Response x to y is within ε of best response"

Proof of Equilibrium.

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)
$\rightarrow$ "Response y to x is within ε of best response"
$\rightarrow$ "Response x to y is within ε of best response"

Proof of approximate equilibrium.

How?

(A) Using geometry.

Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.

Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.

(C) Using multiplicative weights.

Proof of approximate equilibrium.

How?

(A) Using geometry.

(B) Using a fixed point theorem.

(C) Using multiplicative weights.

(C)

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

$$(\max_x xAy^*) - (\min_y x^*Ay) \le \varepsilon$$

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

Games and experts

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

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

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

Games and experts

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

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

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

Experts Framework:
n Experts,

Games and experts

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

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

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

Experts Framework:
n Experts, T days,

Games and experts

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

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

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

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

Games and experts

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

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

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

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

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

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

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

Multiplicative Weights Method yields loss L where

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

Games and Experts.

Games and Experts.

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

Games and Experts.

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

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

Games and Experts.

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

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

1) m pure 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.

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.
Let y_t be distribution (column strategy) on day t.

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.
Let y_t be distribution (column strategy) on day t.

2) Each day, adversary plays best row response to y_t.

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.
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.

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.
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.

x-player (row)

$t = 1$ $t = 2$ $t = 3$ $t = 4$ $\cdots$

y-player (col)

$\propto w^t$ mult. weights

y_1

y_2

$\vdots$

y_m

Picture of Algorithm.

x-player (row)

		$t=1$	$t=2$	$t=3$	$t=4$	$\cdots$
	y_1	1				
y-player (col)	y_2	0				
$\propto w^t$ mult. weights	$\vdots$	$\vdots$				
	y_m	0				

Picture of Algorithm.

x-player (row)

		$t=1$	$t=2$	$t=3$	$t=4$	$\cdots$
	y_1	1	0			
y-player (col)	y_2	0	0			
$\propto w^t$ mult. weights	$\vdots$	$\vdots$	$\vdots$			
	y_m	0	1			

Picture of Algorithm.

x-player (row)

y-player (col)

$\propto w^t$ mult. weights

	$t=1$	$t=2$	$t=3$	$t=4$	$\cdots$
y_1	1	0	0		
y_2	0	0	1		
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
y_m	0	1	0		

Picture of Algorithm.

x-player (row)

		$t=1$	$t=2$	$t=3$	$t=4$	$\cdots$
	y_1	1	0	0	1	
y-player (col)	y_2	0	0	1	0	
$\propto w^t$ mult. weights	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	y_m	0	1	0	0	

Picture of Algorithm.

x-player (row)

		$t = 1$	$t = 2$	$t = 3$	$t = 4$	$\cdots$
	y_1	1	0	0	1	$\cdots$
y-player (col)	y_2	0	0	1	0	$\cdots$
$\propto w^t$ mult. weights	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	y_m	0	1	0	0	$\cdots$

Approximate Equilibrium!

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

Approximate Equilibrium!

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

Let $x^* = \frac{1}{T}\sum_t x_t$

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

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $y^* = \text{argmin}_{y_t} x_t A y_t$.

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

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

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

Thus, algorithm loss, L, is $\geq T \times R(y^*)$.

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $y^* = \text{argmin}_{y_t} x_t A y_t$.

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $y^* = \text{argmin}_{y_t} x_t A y_t$.

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $y^* = \text{argmin}_{y_t} x_t A y_t$.

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $y^* = \text{argmin}_{y_t} x_t A y_t$.

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

Row payoff: $R(y^*) = \max_x xAy^*$.
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 $T \times x^* A$.
$\rightarrow L^* \leq T \times C(x^*) = T \times \min_y x^* Ay$.

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

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

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

$\square$

Wrapping up duality theorem.

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

Wrapping up duality theorem.

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

Does an equilibrium exist?

Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Something about math here?

Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Other proofs: use geometry, linear programming.

Complexity?

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

Wrapping up duality theorem.

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

Does an equilibrium exist? Yes.

Something about math here? Fixed point theorem.

Other proofs: use geometry, linear programming.

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

Wrapping up duality theorem.

For any ε, there exists an ε-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!

Wrapping up duality theorem.

For any ε, there exists an ε-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: $O(n^3 m)$

Wrapping up duality theorem.

For any ε, there exists an ε-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: $O(n^3 m)$ Basically quadratic.

Wrapping up duality theorem.

For any ε, there exists an ε-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 ε, there exists an ε-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: $O(n^3 m)$ Basically quadratic.
(Faster linear programming: $O(\sqrt{n+m})$ linear solution solves.)
Still much slower

Wrapping up duality theorem.

For any ε, there exists an ε-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.

Wrapping up duality theorem.

For any ε, there exists an ε-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.

Dynamics: best response, update weight, best response.

Wrapping up duality theorem.

For any ε, there exists an ε-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.

Dynamics: best response, update weight, best response.

Also works with both using multiplicative weights.

Wrapping up duality theorem.

For any ε, there exists an ε-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: $O(n^3 m)$ 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."

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 $1, \ldots n$, for better predictor?

Boosting...

Get labelled dataset.

Learn model 1.

Take points previous models don't learn well.

Learn model 2.

Repeat.

Combine models $1, \ldots n$, 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 $1, \ldots n$, for better predictor?

How? Majority Vote.

How to analyse?

Boosting Example.

Learning just a bit.

Boosting Example.

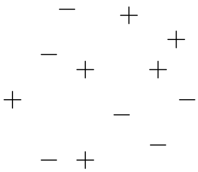
Learning just a bit.

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

Boosting Example.

Learning just a bit.

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

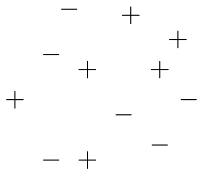


Looks hard.

Boosting Example.

Learning just a bit.

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



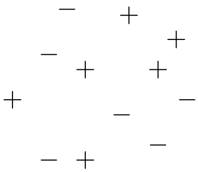
Looks hard.

$1/2$ of them?

Boosting Example.

Learning just a bit.

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



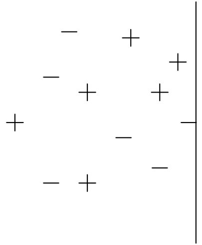
Looks hard.

$1/2$ of them? Easy.

Boosting Example.

Learning just a bit.

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



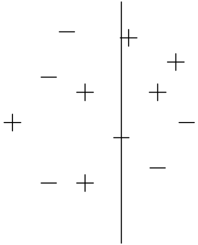
Looks hard.

$1/2$ of them? Easy.
Arbitrary line.

Boosting Example.

Learning just a bit.

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



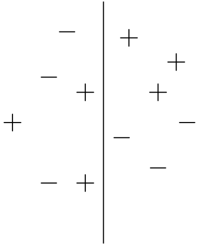
Looks hard.

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

Boosting Example.

Learning just a bit.

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



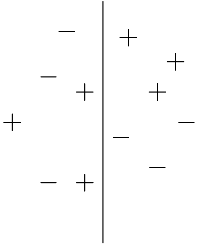
Looks hard.

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

Boosting Example.

Learning just a bit.

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



Looks hard.

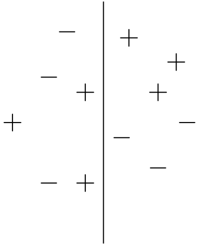
$1/2$ of them? Easy.
Arbitrary line. And Scan.

Useless.

Boosting Example.

Learning just a bit.

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



Looks hard.

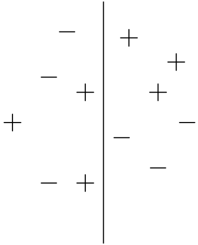
$1/2$ of them? Easy.
Arbitrary line. And Scan.

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.

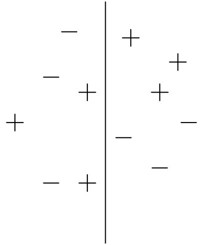
Useless. A bit more than $1/2$ Correct would be better.

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.



Looks hard.

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

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

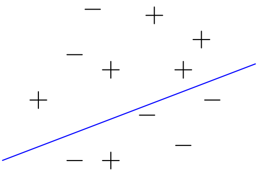
Weak Learner: Classify $\geq \frac{1}{2} + \varepsilon$ points correctly.

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:

Weak Learner/Strong 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 $\frac{1}{2} + \varepsilon$ fraction

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

Weak Learner/Strong Learner

Input: n labelled points.

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

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

Same thing?

Can one use weak learning to produce strong learner?

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.)

Poll.

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

Poll.

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

Poll.

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!

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.

Boosting/MW Framework

Experts are points.

Boosting/MW Framework

Experts are points. ‘Adversary” is weak learner.

Boosting/MW Framework

Experts are points. ‘Adversary” is weak learner.

Points want to be misclassified.

Boosting/MW Framework

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.

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.

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

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.

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.
3. Hypothesis $h(x)$:

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.
3. Hypothesis $h(x)$: majority of $h_1(x), h_2(x), \ldots, h_T(x)$.

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.
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

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.
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 !

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.
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 ! !

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.
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 ! ! !

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.
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!

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.
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?

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.
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.

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 $x \in S_{bad}$.

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

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

$x \in S_{bad}$ is a good expert

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.

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.

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

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 $h_t(x)$ 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.

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

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

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.

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

$$\begin{aligned} W(t+1) &\leq (\tfrac{1}{2}-\gamma)W(t) + (\tfrac{1}{2}+\gamma)(1-\varepsilon)W(t) \\ &\leq W(t)(1-\varepsilon(\tfrac{1}{2}+\gamma)) \end{aligned}$$

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.

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

$$
\begin{aligned}
W(t+1) &\leq (\tfrac{1}{2} - \gamma)W(t) + (\tfrac{1}{2} + \gamma)(1-\varepsilon)W(t) \\
&\leq W(t)(1 - \varepsilon(\tfrac{1}{2} + \gamma)) \leq W(t)e^{-\varepsilon(\frac{1}{2}+\gamma)}.
\end{aligned}
$$

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.

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

$$\begin{aligned} W(t+1) &\leq (\tfrac{1}{2} - \gamma) W(t) + (\tfrac{1}{2} + \gamma)(1-\varepsilon) W(t) \\ &\leq W(t)(1 - \varepsilon(\tfrac{1}{2} + \gamma)) \leq W(t) e^{-\varepsilon(\frac{1}{2}+\gamma)}. \end{aligned}$$

$\rightarrow$

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.

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

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

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

$$\begin{aligned} W(t+1) &\leq (\tfrac{1}{2} - \gamma) W(t) + (\tfrac{1}{2} + \gamma)(1-\varepsilon) W(t) \\ &\leq W(t)(1 - \varepsilon(\tfrac{1}{2} + \gamma)) \leq W(t) e^{-\varepsilon(\frac{1}{2}+\gamma)}. \end{aligned}$$

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

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.

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

$$\begin{aligned} W(t+1) &\leq (\tfrac{1}{2} - \gamma) W(t) + (\tfrac{1}{2} + \gamma)(1-\varepsilon) W(t) \\ &\leq W(t)(1 - \varepsilon(\tfrac{1}{2} + \gamma)) \leq W(t) e^{-\varepsilon(\frac{1}{2}+\gamma)}. \end{aligned}$$

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

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.

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

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

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

Combining

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

Calculation..

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

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

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.

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

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

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

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

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

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

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

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

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

Again, $-\gamma - \gamma^2 \le \ln(1-\gamma)$,

$$\ln\left(\frac{|S_{bad}|}{n}\right) + \frac{T}{2}(-\gamma-\gamma^2) \le -\gamma T(\tfrac{1}{2}+\gamma) \;\rightarrow \ln\left(\frac{|S_{bad}|}{n}\right) \le -\frac{\gamma^2 T}{2}$$

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

$$\rightarrow \ln\left(\frac{|S_{bad}|}{n}\right) \le \ln\mu \rightarrow \frac{|S_{bad}|}{n} \le \mu.$$

Calculation..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

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 μ 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.

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

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

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

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 μ 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.

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

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

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

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 μ 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.

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

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

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

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 μ 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..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

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 μ 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..

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

Set $\varepsilon = \gamma$, take logs.

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

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

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

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 μ 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.

Some details...

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.