Analytic: We showed 1. If

\[ \forall \mathbf{y} \in \text{your class loss} \]

\[ \mathbf{A}[\mathbf{y}] \geq \frac{\mathbf{y}^T \mathbf{e}}{m+1} \]

After loss reaquired, set \( w_i = \frac{\mathbf{y}^T \mathbf{e} - (1 - \mathbf{y}^T \mathbf{x})}{m+1} \) for all \( i \).

Initialize \( w_i = m_i \) for all \( i \).

Parameter: \( \rho > 1 \).

Hege/Matching weights Alg:

\[ \begin{align*}
&\text{Algorithm 1:} \\
&\text{Initialize loss} \text{ (over days) to best (smallest)} \\
&\text{value of } N = 1.6. \\
&\text{value of } T = 1. \\
&\text{for } t = 1, 2, \ldots, T \\
&\text{for } i = 1, 2, \ldots, N \\
&\text{Adversary selects losses } \mathbf{L}_t \text{, } \mathbf{f}_t \text{.} \\
&\text{Algorithm chooses probabilites } \mathbf{p}_t \text{, } \mathbf{q}_t \text{.} \\
&\text{At time } t = 1, 2, \ldots, T \\
&\text{Record: Error of set, machines ("experts") } \mathbf{A}_t \text{.} \\
\end{align*} \]

\[ \begin{align*}
\text{At time } t = 1, 2, \ldots, T \\
\text{Online Learning/} \text{multiplicitive weights} \\
\text{vs. solving linear programs via} \\
\text{Lecture 4: Solving linear programs via}
\end{align*} \]
\[ \Rightarrow \ln \left( 1 - \varepsilon l_i^* \right) + \ldots + \ln \left( 1 - \varepsilon l_i^* \right) \leq -\varepsilon (Y_L) + \ln N \tag{3} \]

Now use \( \ln \left( 1 - x \right) \approx -x - \frac{1}{2} x^2 \) for small \( x \) to get
\[ \ln \left( 1 - \varepsilon l \right) \geq -\varepsilon - \varepsilon l^2 \geq -\varepsilon - \varepsilon^2, \quad \text{since} \quad -1 < l < 1 = e^{-1}. \]

\[ \Rightarrow -\varepsilon l_i^* - \varepsilon^2 - \varepsilon l_i^* - \varepsilon^2 - \ldots - \varepsilon l_i^* - \varepsilon^2 \leq -\varepsilon (Y_L) + \ln N. \]

\[ \Rightarrow \varepsilon (Y_L) \leq \varepsilon (l_i^* + \ldots + l_i^*) + \varepsilon^2 T + \ln N \]

\[ \Rightarrow Y_L \leq (\text{total loss of always doing } i^*) + \varepsilon T + \frac{\ln N}{\varepsilon}. \]

\[ \text{[or divided by } T \quad \text{...]} \quad \frac{1}{T} Y_L \leq \frac{1}{T} (\text{loss of always } i^*) + \frac{\ln N}{\varepsilon T} \]

\[ \text{your avg.} \quad \text{avg. loss/day} \quad \text{small } \varepsilon \text{ diminishes over time} \]

\[ \text{loss/day} \quad \text{of "always } i^*" \]

A good choice of \( \varepsilon \): balance \( \varepsilon = \frac{\ln N}{\varepsilon T} \) \( \Leftrightarrow \) \( \varepsilon = \frac{\ln N}{T} \)

\[ \Rightarrow \frac{1}{T} Y_L \leq \frac{1}{T} (B_L) + \frac{\sqrt{\ln N}}{\sqrt{T}} \]

\[ \text{dimmishes over time, small once } T \gg \ln N. \]

Solving LPs with this.

Zero-sum games

Zero-sum games are a kind of problem from Game Theory/Economics. They're a special case of LPs (like Flows), though actually they're kind of "equiv.;

you can prove every LP can be reduced to one. We won't, tho; will just be content to solve them.
Zero-Sum Games (e.g.: Rock-Paper-Scissors)

2 players, Alice & Bob.

N_1 "actions" \ N_2 "actions"

"Payoff matrix" M: \( M_{ij} \) \( N_1 \times N_2 \)

\( M_{ab} \) = how much Alice pays Bob if she plays a, he plays b

\( \text{loss to Alice, gain to Bob; these sum to zero} \)

WLOG, \( |M_{ab}| \leq 1 \) for all \( a,b \).

Who plays first?

\( \rightarrow \) Play "simultaneously", and each may use a "mixed strategy" = probability distr. on actions

If Alice uses \( p_1, p_2, \ldots, p_{N_1} \),
Bob uses \( q_1, q_2, \ldots, q_{N_2} \), Alice’s expected loss is \( \sum_{a,b} p_a q_b M_{ab} \).

What are their "optimal strategies?"

How to compute? \( \text{Not an LP, seemingly...} \)

Alternate version 1: Hard on Alice:

- Alice must first announce her randomized strat. \( p_1, p_2, \ldots, p_{N_1} \).
- Bob may now choose his randomized strat. \( q_1, q_2, \ldots, q_{N_2} \).

But Bob’s expected gain is \( q_1 \left( \sum_a p_a M_{a,1} \right) + \cdots + q_{N_2} \left( \sum_a p_a M_{a,N_2} \right) \).

Bob should just put 100% prob. on whichever of these is largest.

- Bob may as well be deterministic.

Alice’s goal: minimize \( \max \left\{ \sum_a p_a M_{a,1}, \ldots, \sum_a p_a M_{a,N_2} \right\} \)

s.t. \( p_1, \ldots, p_{N_1} \geq 0 \)

\( p_1 + \cdots + p_{N_1} = 1 \)

An LP! Say its opt. value is \( L_{\text{hard}} \): least expected loss in hard ver. for Alice.
Alt ver 2: Easy on Alice:
- Bob must first announce his randomized strat \( q_1, \ldots, q_N \).
- Alice may now choose hers.
- May as well be deterministic.
- Bob's optimal strategy is given by an LP: \( \sum a_{ij} z_j = \text{maximization} \)
call its opt. value \( L_{\text{easy}} \), least expected loss in this easy ver.

\[
L_{\text{easy}} \leq L \leq L_{\text{hard}}
\]

if Alice goes first expected Alice-loss when they play (optimal strats) simultaneously

von Neumnan Minimax Theorem: \( L_{\text{easy}} = L_{\text{hard}} \). \( = L \).

Proof 1 (hawk): The two LPs, for \( L_{\text{easy}}, L_{\text{hard}}, \) are duals.

Proof 2 ... [We'll show it, & we'll in fact show an algorithm to find the optimal strategies achieving \( L \).]

Need to show \( L_{\text{hard}} \leq L_{\text{easy}}. \)

Consider your Alice's actions as \( i = 1, \ldots, N \), the slots/experts.

Play Hedge for a while...

On day \( t \), \( P_t = \cdots = P_1 \).

Let adversary "think twice".

Consider playing "Hard on Alice" version \( T \) days in a row.

Treat Alice's/your options \( a = 1, \ldots, N \) as slots/experts.

You/Alice will play \( P_t, \ldots, P_{t+T-1} \) according to Hedge strategy on round \( t \).

Adversary/Bob will play "best response" in Zero Sum Game, \( b_t \), on round \( t \).

\( \Rightarrow \) yields a "loss vector" for Hedge, \( L_t = M_i b_t \).

Alice updates with this loss vector.
e.g. Z.S.G.:  

<table>
<thead>
<tr>
<th></th>
<th>$w_0$</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>2</td>
<td>-6</td>
<td>+9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>day</th>
<th>Alice mixed strat</th>
<th>Bob's response</th>
<th>Alice's expected loss</th>
<th>Alice's expected loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(\frac{1}{2}, \frac{1}{2})$</td>
<td>2</td>
<td>$(-\frac{3}{2} + \frac{1}{2})$</td>
<td>$= +\frac{3}{2}$</td>
</tr>
<tr>
<td>2</td>
<td>$w_1 = 4.24$, $w_2 = 1.93$</td>
<td>2</td>
<td>$(-\frac{3}{2} + \frac{1}{2})$</td>
<td>$= +\frac{3}{2}$</td>
</tr>
<tr>
<td>3</td>
<td>$(.56, .44)$</td>
<td>2</td>
<td>$(-\frac{3}{2} + \frac{1}{2})$</td>
<td>$= +\frac{3}{2}$</td>
</tr>
<tr>
<td>4</td>
<td>$(.62, .38)$</td>
<td>2</td>
<td>$(-\frac{3}{2} + \frac{1}{2})$</td>
<td>$= +\frac{3}{2}$</td>
</tr>
<tr>
<td>5</td>
<td>$(.77, .23)$</td>
<td>1</td>
<td>$(+\frac{3}{2}, -\frac{1}{2})$</td>
<td>( \quad )</td>
</tr>
<tr>
<td>6</td>
<td>$(.77, .23)$</td>
<td>1</td>
<td>$(+\frac{3}{2}, -\frac{1}{2})$</td>
<td>( \quad )</td>
</tr>
</tbody>
</table>

\( \frac{1}{T} (\text{Your Alice loss}) \geq L_{\text{hard}}, \) since You/Alice had to go first each time.

In Hedge ver., what - in hindsight - would be the best single play for You/Alice?

Define \( q_1 = \frac{\text{frac times Bob's resp 1}}{T} \), \( q_2 = \frac{\text{frac times Bob resp 2}}{T} \), \( 2w_v = \frac{M_v}{T} \).

\( \frac{1}{T} (\text{Best Loss}) \) is avg. value of best response by Alice to Bob playing mixed strat \( 2w_v \) ... \( 2w_v \) !

\( \implies L_{\text{hard}} \leq L_{\text{easy}}. \)

after \( T \) rounds, we conclude

\[
L_{\text{hard}} \leq \frac{1}{T} (\text{Your loss}) \leq \frac{1}{T} (\text{Best Loss}) + \varepsilon + \frac{\ln N}{\varepsilon T} \leq L_{\text{easy}} + \varepsilon + \frac{\ln N}{\varepsilon T}.
\]

\( \implies \) must have \( L_{\text{hard}} \leq L_{\text{easy}} \), because if \( L_{\text{hard}} > L_{\text{easy}} + \delta \), we could make \( \varepsilon < \frac{\delta}{2} \), then \( T \)

large enough so \( \frac{\ln N}{\varepsilon T} < \frac{\delta}{2} \),

get \( L_{\text{hard}} \leq L_{\text{easy}} + \delta, \implies \varepsilon. \)
within a few optimal steps, and then $P$ are

Carn all algorithmically find them:

only $c + \ln n$ steps from the optimal key = $c$ time.

Moreover, the average $p$ in a best plays

8 the game, the average $p$ in Alice plays