A Motivating Example: Weighted Majority

- Suppose want to bet on football game every week
- Before every game, can go on internet and see what “experts” predict
  - Some experts are very good, others are terrible
  - Also, some might be good on some games and bad on others, or be good at the beginning of the year and bad at the end, etc.
  - So quality of expert might change week-to-week
  - And the experts might be arbitrarily correlated
- Want to somehow take all of their opinions into account and figure out which team to bet on
- We’ll show how to make decisions as we go along so that, in hindsight, we perform about as well as the best of the experts
  - This is tricky because we don’t know which one was the best until the end
  - Simple strategies, like following the expert who has done the best so far, don’t work
  - Could have an expert who does well early in the year, but does terribly thereafter
    - E.g., An expert who, this year, always says, “bet on the Giants”...

**Weighted Majority Algorithm**

- **Initialize:** $w_i^1 = 1$ for all $i$
- **At step $t$:**
  - Prediction for step $t$ given by weighted majority of experts
  - If expert $i$ was wrong, set $w_i^{t+1} = (1 - \epsilon)w_i^t$
  - Otherwise, set $w_i^{t+1} = w_i^t$

- Let $m_i^t = \text{number of mistakes made by expert } i \text{ after } t \text{ steps}$
- Let $m^t = \text{number of mistakes we’ve made after } t \text{ steps}$

**Theorem**

For all $t$ and $i$,

$$m_i^t \leq \frac{2 \ln n}{\epsilon} + 2(1 + \epsilon)m_i^t$$

- So, in particular, holds when $i = \text{the best expert}$
Theorem
For all \( t \) and \( i \),
\[
m^t \leq \frac{2 \ln n}{\epsilon} + 2(1 + \epsilon)m^t_i
\]

Proof:
- Note that \( w_t^i = (1 - \epsilon)^m^t_i \)
- Define potential function:
  \[
  \Phi^t = \sum_i w_t^i
  \]
  \[
  \Phi^1 = n
  \]
  \[
  \Phi^t \geq w_t^i \text{ for all } t \text{ and } i
  \]
- If we made a mistake, more than half of the (weighted) experts must have been wrong
  - So at least half of the total weight must decrease by factor of \( 1 - \epsilon \)
  - So \( \Phi \) decreases by factor of at least \( 1 - \epsilon/2 \) each time we make a mistake
- Gives
  \[
  \Phi^t \leq n(1 - \epsilon/2)^m^t
  \]
- So
  \[
  (1 - \epsilon)^m^t_i \leq n(1 - \epsilon/2)^m^t_i
  \]
- Take logs of both sides and use \(-\ln(1 - x) \leq x + x^2 \) for \( x < 1/2 \)

Notes About the Weighted Majority
- Proof was very simple
  - Basically just noted that if an expert is wrong too many times more than the best expert, his weight will get very small
  - Used no assumptions about sequence of events, correlations between experts, experts having consistent quality, non-adversarial inputs, etc.
- Very general analysis idea, which will underly other MW algorithms
- Two main features:
  - Multiplicative update rule
  - Exponential potential function

Removing a Factor of 2
- We showed \( m^t \leq \frac{2 \ln n}{\epsilon} + 2(1 + \epsilon)m^t_i \)
- Dependence on \( \epsilon \) makes sense
  - Bigger \( \epsilon \) means we follow experts more aggressively, so additive term down but mult. error up
- Factor of 2 in second term is unpleasant
  - Won't really matter for algorithmic applications, but is pretty bad in context of making predictions using expert advice
  - Says we make at most twice as many mistakes as best expert (plus an additive term)
- Why is it there?
  - Consider case when 51% of experts say wrong thing, 49% say right thing
  - We always make a mistake in this case, yet only reduce total weight by 1/4
  - Would be better if prob. made a mistake was tied to amount weight is reduced
  - If experts are almost 50/50, we really should just flip a coin
- Suggests new randomized algorithm

Randomized Weighted Majority Algorithm
- At time \( t \), let expert \( i \) have weight \( w_t^i \)
- Play distribution \( D^t = \{p_1^t, \ldots, p_n^t\} \), where
  \[
  p_i^t = \frac{w_t^i}{\sum_k w_k^t}
  \]
- Initialize: \( w_1^i = 1 \) for all \( i \)
- At step \( t \):
  - Pick an expert according to distribution \( D^t \) and use it make prediction
  - If expert \( i \) was wrong, set
    \[
    w_t^{i+1} = (1 - \epsilon)w_t^i
    \]
  - Otherwise, set \( w_t^{i+1} = w_t^i \)
- Same type of analysis shows expected number of errors we make in first \( t \) steps is at most
  \[
  \frac{\ln n}{\epsilon} + \frac{1}{1 - \epsilon} m^t_i < \frac{\ln n}{\epsilon} + (1 + \epsilon)m^t_i \quad \text{when } \epsilon < 1/2
  \]
A More General Setting

- Suppose now set $P$ of events/outcomes no longer binary (and could even be infinite)
- Also, suppose no longer just “right or wrong”, but get arbitrary number as penalty
- If outcome $j$, expert $i$ pays penalty $M(i, j)$
  - Will assume all $M(i, j) \in [-\ell, \rho]$, for $\ell < \rho$
  - Call $\rho$ the width
- We’ll pick an expert to follow each step
- Our strategy will be randomized
  - Necessary in this new setting
  - Why?
- Want strategy so that expected penalty is not much worse than best expert (in hindsight)
- Maybe doesn’t sound that surprising, but pretty impressive when pick right way to instantiate “experts”
  - E.g., Experts ↔ stocks, probabilities ↔ fractions of a portfolio
  - Says you can buy/sell stocks over time s.t. perform about as well as the single best stock in the portfolio!

- From now on, $\epsilon$ will always be $\leq 1/2$
- Let $M(D^i, j^i)$ denote our expected penalty for outcome $j^i$
- Same type of analysis as before gives

### Theorem

For any $i$ and $T$,

$$\sum_{t=1}^{T} M(D^i, j^i) \leq \frac{\rho \ln(n)}{\epsilon} + (1 + \epsilon) \left[ \sum_{j^i \geq 0} M(i, j^i) \right] + (1 - \epsilon) \left[ \sum_{j^i < 0} M(i, j^i) \right]$$

where $\geq 0$ and $< 0$ in summations select rounds with penalties of given sign.

- In particular, if we take $\epsilon \leq \min\{ \frac{\rho}{4\ell^2}, \frac{1}{2} \}$ and pick any $\delta$, then after
  $$T = \frac{16\rho^2 \ln(n)}{\delta^2}$$
  rounds, can show average error per round obeys
  $$\frac{\sum_{t=1}^{T} M(D^i, j^i)}{T} \leq \delta + \frac{\sum_{t=1}^{T} M(i, j^i)}{T}$$

- Note dependence of number of rounds on width $\rho$
- If $\ell = 0$, can actually get running time prop to $\rho$ instead of $\rho^2$

#### Multiplicative Weights Update Algorithm

- Initialize: $w^i_1 = 1$ for all $i$
- At step $t$:
  - Pick an expert according to distribution $D^i$ and use it make prediction
  - If expert $i$ was wrong, set
    $$w^i_{t+1} = (1 - \epsilon)w^i_t$$
  - Otherwise, set $w^i_{t+1} = w^i_t$
  - Let $j^i \in P$ be outcome
  - Update weight of expert $i$ by:
    $$w^i_{t+1} = \begin{cases} w^i_t(1 - \epsilon)^{M(i, j^i)/\rho} & \text{if } M(i, j^i) \geq 0 \\ w^i_t(1 + \epsilon)^{-M(i, j^i)/\rho} & \text{if } M(i, j^i) < 0 \end{cases}$$

**Approximately Solving Zero-Sum Games**

- Can use this to construct algorithms by creating appropriate experts and payoffs
- We’ll do a bunch of these
- First example: approximately solving zero-sum games
  - Two players, a “row player $R$ and a “column player $C$
  - Each has a (possibly different) finite set of pure strategies
  - Given payoff matrix $M$ whose rows are indexed by $R$’s strategies and whose cols are indexed by $C$’s strategies
  - If $R$ plays strategy $i$ and $C$ plays strategy $j$, then payoff from $R$ to $C$ is given by $M(i, j)$
    - For simplicity, let’s assume normalized s.t. all $M(i, j) \in [0, 1]$
  - Allowed to play mixed strategies = probability distributions over strategies
  - Let $D$ and $P$ be row and column mixed strategies, resp.
  - If col player knows $D_i$, will want to play strategy that gives $\max_j M(D, j)$
  - If row player knows $P_i$, will want to play strategy that gives $\max_j M(P, j)$
  - von Neumann’s Minimax Theorem says:
    $$\lambda^* := \min_D \max_j M(D, j) = \max_P \min_i M(i, P)$$
  - $\lambda$ is called value of the game
    - Goal is to find it up to some additive error $\delta$
Approximately Solving Zero-Sum Games with M-W

\[ \lambda^* := \min_D \max_j M(D, j) = \max_P \min_i M(i, P) \]

- Experts ↔ pure strategies for \( R \)
- Events ↔ pure strategies for \( C \)
- Penalty paid by expert \( i \) when have event \( j \) is \( M(i, j) \)
- For any given \( D \), we assume know how to find column strategy \( j \) that maximizes \( M(D, j) \)
  - This is always \( \geq \lambda^* \)
- So imagine following scenario, in which play game for many rounds:
  - Every round, we choose a distrib. over experts ↔ mixed strategy \( D \) for \( R \)
  - Get back the worst possible event ↔ col. strat. that maximizes \( M(D, j) \)
  - We pay penalty \( M(D, j) \)
- Exactly setup from earlier, and can choose \( D \) using M-W

Let's see what our M-W analysis gives us

For any distribution \( D \), know \( \sum_t M(D, j_t) \geq \min_i \sum_t M(i, j_t) \)
  - Since distribution is just a weighted average of pure strategies
- Also know that, for all \( t \), our distribution at time \( t \) has \( M(D_t, j_t) \geq \lambda^* \)
  - Since column player plays his best strategy given our choice of \( D \)
- M-W theorem thus gives that, after \( T = 16 \ln(n)/\delta^2 \) rounds and for any \( D \),
  \[ \lambda^* \leq \frac{\sum_{t=1}^T M(D_t, j_t)}{T} \leq \delta + \min \left\{ \frac{\sum_{t=1}^T M(i, j_t)}{T} \right\} \leq \delta + \frac{\sum_{t=1}^T M(D_t, j_t)}{T} \]
  - Take \( D = \) optimal row strategy \( D^* \), get \( \lambda^* \leq \frac{\sum_{t=1}^T M(D_t, j_t)}{T} \leq \delta + \lambda^* \)
  - So \( \frac{\sum_{t=1}^T D_t}{T} \) is approximately optimal strategy, and \( \frac{\sum_{t=1}^T M(D_t, j_t)}{T} \) is approximately value of the game
  - Could also have used best \( D_t \) instead of average one

Solving Linear Programs

- It turns out that solving any LP can be cast as solving a zero-sum game
  - Will discuss something even more general next lecture
  - Need to make sure to pay attention to width
- How parameters that dictate rate of convergence compare to ellipsoid or interior point methods is very interesting
- We get error \( \delta \) after something \( O(\ln n/\delta^2) \) iterations
  - Don't forget: in the width 1 case
  - Otherwise, need to throw in a \( \rho^2 \)
- Ellipsoid / IP alg get error \( \delta \) after \( \text{poly}(n) \ln(1/\delta) \)
  - And only depend logarithmically on what would correspond to the width
- So interesting tradeoff: IP alg are much better w.r.t. error and size of numbers, whereas M-W alg are much better w.r.t. dimension.
- More on this next lecture