Note: Your TA probably will not cover all the problems. This is totally fine. The discussion worksheets are deliberately made long so they can serve as a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 Movie Critics

There are five well-known online movie critics. Every Friday, each critic predicts whether the new blockbuster will be a *Hit* or a *Flop* at the box office. For four Fridays in a row, the five critics made the following predictions:

Week	Movie	Critic A	Critic B	Critic C	Critic D	Critic E	Outcome
1	Zootopia 2	Hit	Flop	Hit	Hit	Flop	Hit
2	Lilo & Stitch	Flop	Flop	Hit	Flop	Hit	Flop
3	Wicked: For Good	Hit	Flop	Flop	Flop	Hit	Flop
4	KPop Demon Hunters	Hit	Hit	Flop	Flop	Hit	Hit

(a) Based on the last 4 weeks, which critic is the best expert *in hindsight* (i.e., has the fewest mistakes)? If there is a tie, list all tied critics.

(b) Suppose we took the majority prediction of all 5 critics over the last 4 weeks. How many mistakes would we have made if we made this prediction for the last 4 weeks' movies?

(c) Next week, a new blockbuster, *Avatar: Fire and Ash*, is being released. You have to decide between the following two potential strategies to predict whether the upcoming blockbuster is a Hit or a Flop:

Best-so-far. Copy the prediction of the critic that had made the fewest mistakes *so far*.

Majority. Take the majority answer of the 5 critics' predictions for *Avatar: Fire and Ash*.

Which of these two strategies is better to follow? Briefly explain why.

2 Halving Algorithm for K Outcomes

So far in lecture, we have only considered the case when experts are predicting either a "Yes" or a "No". Let us now consider the more general case where they are predicting among more than 2 possibilities. Suppose there are n experts, and on each round $t = 1, 2, \ldots, T$ there are $K \geq 2$ possible outcomes, which we denote by the set $\{1, 2, \ldots, K\}$. On each round:

- Each expert i predicts one of the K outcomes, denoted $\hat{y}_{t,i} \in \{1, \ldots, K\}$.
- The algorithm must output a single prediction $\hat{y}_t \in \{1, \ldots, K\}$.
- Then the true outcome $y_t \in \{1, \ldots, K\}$ is revealed.
- An expert i incurs a loss of 0 if $\hat{y}_{t,i} = y_t$ (i.e., they predicted correctly) and 1 otherwise.

We will generalize the deterministic Halving algorithm given in lecture from the binary setting to the K outcome setting presented here.

Assume throughout this problem that there exists an expert among the n experts who makes *no mistakes* (i.e., has total loss 0 over T rounds).

Recall that in the binary case, the Halving algorithm maintains the set of *surviving* experts who have not yet made any mistakes, predicts by majority vote among these experts, and then eliminates all experts that predicted incorrectly.

(a) How should the algorithm use the expert predictions and the current set of surviving experts to produce a single prediction $\hat{y}_t \in \{1, \ldots, K\}$? Give a natural rule in terms of how many surviving experts predict each outcome.

(b) Briefly argue why your algorithm from the previous part is a direct extension of the binary Halving algorithm, in the sense that it only depends on whether each expert was correct or incorrect, not *which* incorrect outcome they gave.

(c) Show that the number of mistakes that this algorithm would make is at most $\log_{K/(K-1)} n$. Note that when $K = 2$, this recovers the same mistake upper bound as was given in lecture.

3 Multiplicative Weights Warmup

You are choosing among *three caching strategies* for a web service: Strategy A, B, and C. On each day t, each strategy either performs well (loss 0) or poorly (loss 1).

The losses of the 3 strategies over the 5 days are:

Day t	1	2	3	4	5
A	0	1	0	1	0
B	1	0	1	0	1
C	0	0	1	1	0

You run the multiplicative weights algorithm with parameter $\varepsilon = \frac{1}{2}$. Weights are initialized to $w_A^{(1)} = w_B^{(1)} = w_C^{(1)} = 1$. On each day $t = 1, \ldots, T$, you:

1. Compute the total weight $W^{(t)} = w_A^{(t)} + w_B^{(t)} + w_C^{(t)}$.
2. Pick a strategy i with probability $p_i^{(t)} = w_i^{(t)}/W^{(t)}$.
3. Observe the loss vector $c^{(t)} = (c_A^{(t)}, c_B^{(t)}, c_C^{(t)})$, where $c_A^{(t)}, c_B^{(t)}, c_C^{(t)} \in [0, 1]$.
4. Update your weights using the rule $w_i^{(t+1)} = w_i^{(t)}\big(1 - \varepsilon\, c_i^{(t)}\big)$.

For this problem, you can ignore the actual random choice and just work with the probabilities and expected losses.

(a) Fill in the following table of weights and probabilities for each day $t = 1, \ldots, 5$. (You do not have to simplify fractions aggressively, but keep them readable.)

Day t	$w_A^{(t)}$	$w_B^{(t)}$	$w_C^{(t)}$	$p_A^{(t)}$	$p_B^{(t)}$	$p_C^{(t)}$
1						
2						
3						
4						
5						

(b) For each day, compute the *expected loss* of the algorithm. Namely, compute the value

$$\mathbb{E}[\ell^{(t)}] = \sum_{i \in \{A,B,C\}} p_i^{(t)} \, c_i^{(t)},$$

and the *total expected loss* over all 5 days.

(c) Compute the total loss of each single fixed strategy (A, B, C) over the 5 days. Which strategy is best in hindsight, and what is its loss?

(d) What is the algorithm's *regret* on this loss sequence? In other words, compute the value

$$\text{Regret} = \text{Algorithms expected total loss} - \min\{\text{loss(A)}, \text{loss(B)}, \text{loss(C)}\}.$$

Experts problem

- There are n experts and T rounds.
- On each round $t = 1, \dots, T$:
  - Each expert i makes a prediction.
  - Based on their predictions, we pick an expert i and follow their prediction.
  - After the outcome is revealed, each expert i suffers a loss $l_i^{(t)} \in [0, 1]$ as well as us.
- **Goal:** perform almost as well as the best expert in hindsight.

Definitions

- $l_i^{(t)}$ = loss of expert i on day t.
- $w_i^{(t)}$ = weight of expert i before the t'th round.
- $p_i^{(t)}$ = probability that we picked expert i on round t. Note that $\sum_{i=1}^n p_i^{(t)} = 1$.
- Our loss on day t:

$$\text{loss}^{(t)} = \sum_{i=1}^{n} p_i^{(t)}\, l_i^{(t)}.$$

- Total regret up to day T:

$$R_T = \sum_{t=1}^{T} \sum_{i=1}^{n} p_i^{(t)} l_i^{(t)} \; - \min_{i \in \{1,\dots,n\}} \sum_{t=1}^{T} l_i^{(t)}.$$

  (Our total expected loss minus the total loss of the best expert.)

Multiplicative Weights Algorithm (losses in $[0, 1]$)

- **Parameter:** Choose a learning rate $0 < \epsilon \leq \frac{1}{2}$.
- **Initialization:** $w_i^{(1)} = 1 \quad$ for all $i = 1, \dots, n$.
- For each round $t = 1, 2, \dots, T$:
  1. **Compute the probabilities** using the formula

  $$p_i^{(t)} = \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} \quad \text{for all } i = 1, 2, \dots, n.$$

     Think of $p_i^{(t)}$ as the 'trust' we place in expert i's prediction at the t'th round.
  2. **Make a decision** by sampling an index $i \in \{1, \dots, n\}$ with probability $p_i^{(t)}$
  3. **Observe outcome and losses** by getting the values $l_i^{(t)} \in [0, 1]$ for all experts $i = 1, 2, \dots, n$.
  4. **Update our weights** using the update rule

  $$w_i^{(t+1)} = w_i^{(t)} \, (1 - \epsilon)^{\, l_i^{(t)}} \quad \text{for all } i = 1, 2, \dots, n$$

     So experts with larger loss $l_i^{(t)}$ get multiplied by a smaller factor and lose more weight.

Regret guarantee

Assuming $0 < \epsilon \leq \frac{1}{2}$ and all losses $l_i^{(t)}$ lie in $[0, 1]$, our total regret is at most

$$R_T \ \leq \ \epsilon T + \frac{\ln n}{\epsilon}.$$

A common choice is $\epsilon = \sqrt{\frac{\ln n}{T}}$. When plugged into the bound above, we get the total regret upper bound

$$R_T \leq 2\sqrt{T \ln n}.$$

Thus, by following the multiplicative weights algorithm, the *average* regret R_T/T goes to 0 as T grows. This means the algorithm eventually does almost as well as the best expert chosen in hindsight.

Intuition for Multiplicative Weights

- Start by trusting all experts equally.
- Each time an expert does badly, their weight is multiplied by a factor less than 1, so their influence shrinks.
- Experts that consistently do well keep a relatively large weight and dominate the prediction.
- This is a soft version of eliminating bad experts: instead of throwing them away, we gradually discount them.