DSAI 512 Fall 2025
HW #2

Yiğit Ateş - 2025776009
Institute for Data Science and Artificial Intelligence, Boğaziçi University

I. Problem 1

Drawing marbles with replacement, where:

- μ: probability of drawing red
- $1-\mu$: probability of drawing green
- $N = 10$: sample size
- ν: fraction of red in sample
- $\mu \in \{0.05, 0.5, 0.8\}$
- $P(\nu = 0)$: the probability that no red marbles are drawn.

A. 1.a

Since draws are independent with replacement:

$$P(\nu = 0) = (1-\mu)^{10} \tag{1}$$

$$\begin{aligned} \mu = 0.05: \quad & P(\nu = 0) = (1-0.05)^{10} \approx 0.5987 \\ \mu = 0.5: \quad & P(\nu = 0) = (1-0.5)^{10} \approx 0.0010 \\ \mu = 0.8: \quad & P(\nu = 0) = (1-0.8)^{10} \approx 1.024e-7 \end{aligned}$$

B. 1.b

Let event $\mathcal{B}$: at least one sample has $\nu = 0$
Let event $\overline{\mathcal{B}}$: none of the samples has $\nu = 0$
Using the complement rule:

$$P(\mathcal{B}) = 1 - P(\overline{\mathcal{B}}) = 1 - \left(1-(1-\mu)^{10}\right)^{1000} \tag{2}$$

$$\begin{aligned} \mu = 0.05: \quad & 1-(1-0.5987)^{1000} \approx 1.0000 \\ \mu = 0.5: \quad & 1-(1-0.0010)^{1000} \approx 0.6323 \\ \mu = 0.8: \quad & 1-(1-1.024e-7)^{1000} \approx 0.0001 \end{aligned}$$

C. 1.c

Same calculation with $B =$ 1,000,000 experiments:

$$P(\mathcal{B}) = 1 - \left(1-(1-\mu)^{10}\right)^{1000000} \tag{3}$$

$$\begin{aligned} \mu = 0.05: \quad & 1-(1-0.5987)^{1000000} \approx 1.0000 \\ \mu = 0.5: \quad & 1-(1-0.0010)^{1000000} \approx 1.0000 \\ \mu = 0.8: \quad & 1-(1-1.024e-7)^{1000000} \approx 0.0973 \end{aligned}$$

D. 1.d

When we observe the $\mu = 0.5$ case from 1.a, we can see the bad event ($\nu = 0$) is very unlikely to happen, with a probability of only 0.0010. This is similar to running a single train_test_split experiment with a fixed random_state from scikit-learn. However, when we increase the number of experiments or in other words try many hypotheses, we see the probability of at least one bad event occuring jumps to 0.6323 (for $B = 1000$) and then to 1.000 (for $B = 1,000,000$). This means as the number of hypotheses grows, it becomes almost certain that at least one of them will mislead us. Since we always look for the best hypothesis, we're very likely to get fooled by that one. To give a more vivid example, if our random_state in train_test_split gives us a misleading sample like 10 red marbles while the truth is $\mu = 0.5$, a hypothesis g that says 'always guess red' will get a perfect score ($E_{in} = 0$). This hypothesis will be chosen because it looks the best, leading us to be fooled. This is the essence of overfitting as in sample error (E_{in}) is low, but its true error (E_{out}) is much higher. This is why the Hoeffding inequality, by itself, isn't enough. It only defines the bound for one hypothesis. That's why introducing the Union Bound is necessary. It states that **the total probability of at least one of our many hypotheses being misleading is no more than the sum of all their individual probabilities.**

II. Problem 2

A. 2.a

No, it does not guarantee better performance. If we assume that our dataset is biased towards mostly low temperature like $\{1, -1, -1, \ldots, -1\}$, Algorithm A will choose h_2 as it will get the best score with 96% accuracy. However, in the real world if the true distribution is nearly balanced ($p \approx 0.5$), because of this biased distribution in D, our accuracy will fall to approximately 50% and since this is not better than random guessing, it is not guaranteed.

B. 2.b

If all samples have +1 results, then Algorithm A chooses h_1 as it yields the best result and Algorithm B chooses h_2. In a scenario where Algorithm B performs better in the real world, it is needed that $f(x) = -1$ probability should be greater than $f(x) = +1$:

$$\begin{aligned} 1 - p &> p \\ 1 &> 2p \\ 0.5 &> p \end{aligned}$$

In this specific case, if $0.5 > p$, then it is possible.

C. 2.c

For case 1 where $p = 0.9$, Algorithm A performs better if it selects h_1. For Algorithm A to choose h_1, the dataset D must observe that h_1 outperforms h_2, which occurs only when the number of $+1$ samples exceeds the number of -1 samples. We can express this probability as $P(k \geq 13)$, where k represents the number of $+1$ samples in the 25 readings.

For case 2 where $p = 0.1$, Algorithm A performs better if it selects h_2. This occurs when the number of -1 samples exceeds the number of $+1$ samples. We can express this probability as $P(k < 13)$.

Since these show binomial distribution we can define and compute them:

$$P(k \geq 13) = \sum_{k=13}^{25} \binom{25}{k} (0.9)^k (1-0.9)^{25-k} \approx 1.0000$$

$$P(k < 13) = \sum_{k=0}^{12} \binom{25}{k} (0.1)^k (1-0.1)^{25-k} \approx 1.0000$$

D. 2.d

Let P_A be the probability that Algorithm A chooses the correct hypothesis and P_B be the probability that Algorithm B chooses the correct hypothesis. Since B always chooses the opposite of A, B is correct when A is wrong denoted by:

$$P_B = 1 - P_A$$

The condition for B to be more likely to perform better is $P_B > P_A$:

$$\begin{aligned} 1 - P_A &> P_A \\ 1 &> 2P_A \\ 0.5 &> P_A \end{aligned}$$

The problem is to find the range of p for which $P_A < 0.5$.

If $p > 0.5$: The correct hypothesis is h_1.

A chooses h_1 if $k \geq 13$. Thus, $P_A = P(k \geq 13)$. The mean of the distribution is $E[k] = np = 25p > 12.5$. Since the mean is greater than 12.5, the probability is concentrated on this side, so $P(k \geq 13) > 0.5$.

If $p < 0.5$: The correct hypothesis is h_2.

A chooses h_2 if $k < 13$. Thus, $P_A = P(k < 13)$. The mean of the distribution is $E[k] = np = 25p < 12.5$. Since the mean is less than 12.5, the probability is concentrated on this side, so $P(k < 13) > 0.5$.

If $p = 0.5$: Both hypotheses are equally correct.

$P_A = P(k < 13) = P(k \geq 13) = 0.5$.

The condition $P_A < 0.5$ is never met. Therefore, there is no solution for the case where B is more likely to perform better. This shows that Algorithm A is always a better (or equal) strategy than deliberately contradicting the result despite having limited data.

E. 2.e

Algorithm C (Cautious) chooses h_1 or h_2 with 50% probability, regardless of the data D. Its expected out-of-sample accuracy is:

$$E[\text{Acc}_C] = P(\text{picks } h_1) \cdot \text{Acc}(h_1) + P(\text{picks } h_2) \cdot \text{Acc}(h_2)$$

$$E[\text{Acc}_C] = (0.5) \cdot (p) + (0.5) \cdot (1-p)$$

$$E[\text{Acc}_C] = 0.5p + 0.5 - 0.5p = 0.5$$

Algorithm C's accuracy is always 50%, i.e., random guessing.

As p varies, $E[\text{Acc}_C]$ is constant at 0.5.From 2.d, we know Algorithm A's probability of being correct (P_A) is always ≥ 0.5, while B's (P_B) is always ≤ 0.5. This means A's expected accuracy will be ≥ 0.5 and B's will be ≤ 0.5.Thus, Algorithm A (the "smart" learner) always performs better than or equal to Algorithm C (the random learner). Algorithm C always performs better than or equal to Algorithm B (the contrarian).

Condition:The "smart" learner (A) performs as well as the "purely random" learner (C) when $E[\text{Acc}_A] = E[\text{Acc}_C] = 0.5$. This occurs only when the underlying problem is also purely random, at $p = 0.5$. In this case, both h_1 and h_2 have 50% accuracy, so any choice (or random choice) yields the same 50% expected accuracy.

F. 2.f

This toy scenario illustrates:The danger of overinterpreting small datasets: The scenario in 2.b, where D consists of 25 +1 samples, shows this perfectly. An algorithm "learning" from this data (Algorithm A) would be 100% confident that h_1 is correct. However, if the true p is 0.1, this learned hypothesis is catastrophically wrong. The small dataset was "unlucky" and extremely unrepresentative of the real world.The probabilistic nature of "learning" success or failure: "Learning" is not a guaranteed success. As shown in 2.c and 2.d, the success of Algorithm A (P_A) is a probability. This probability depends on $n = 25$ (the data size) and p (the "real world" difficulty). If $p = 0.9$, learning is almost guaranteed to succeed ($P_A \approx 1$). If $p = 0.5$, learning is no better than a coin flip ($P_A = 0.5$).The need for generalization bounds: Since learning can fail, we need tools to quantify the risk of failure. A generalization bound (like Hoeffding or Union bound, as seen in Problem 1.d) provides a mathematical link between what we see in sample (E_{in}) and what we can expect out-of-sample (E_{out}). It tells us the probability that our dataset D is misleading us, allowing us to state with a certain confidence (δ) that the "true" error is not too far from the error we measured.

III. Code Repository

Computations will be available at *this GitHub repository* after the deadline.

IV. Acknowledgements

Gemini 2.5 Pro was used to assist in converting handwritten mathematical notation to LaTeX format and ensuring my analogies make sense within the context.