Overview

Submit your writeup, including any code, as a PDF via gradescope.[1] We recommend reading through the entire homework beforehand and carefully using functions for testing procedures, plotting, and running experiments. Taking the time to reuse code will help in the long run!

Data science is a collaborative activity. While you may talk with others about the homework, please write up your solutions individually. If you discuss the homework with your peers, please include their names on your submission. Please make sure any handwritten answers are legible, as we may deduct points otherwise.

1. Math Stats

Work through the following exercises, and explain your reasoning in your answer.

(a) Suppose a particular drug test is 99% sensitive and 98% specific. The null hypothesis H_0 is that the subject is not using the drug. Assume a prevalence of $\pi_1 = 0.5\%$, i.e. only 0.5% of people use the drug. Consider a randomly selected individual undergoing testing. Rounding to the nearest three significant figures, find

(i) **(1pt)** the probability of testing positive given H_0.

(ii) **(1pt)** the probability that they are not using the drug given they test positive.

(iii) **(2pt)** the probability of testing positive a second time given they test positive once. You may assume the two tests are statistically independent given drug user status.

(b) Suppose we have a waiting time $T \sim \text{Exponential}(\lambda)$ and wish to test

$$H_0 : \lambda = c \qquad \text{vs} \qquad H_1 : \lambda = 2c$$

for some $c > 0$. In this question, you'll use the *likelihood ratio test* (LRT) to compare these two hypotheses. The LRT considers the ratio of the two density functions:

$$LR(T) = \frac{\mathcal{L}(x|H_1)}{\mathcal{L}(x|H_0)},$$

and rejects H_0 when $LR(T)$ is greater than some threshold η.

We use this test because of the *Neyman-Pearson lemma*, which states that the likelihood ratio test is the most powerful test (in other words, it has the highest power, or TPR) of significance level α. That is, out of all possible tests of H_0 vs H_1 with FPR $= \alpha$, the likelihood ratio test has the highest TPR.

Hint: For this question, you may find it helpful to brush up on computing probabilities involving continuous random variables. Prob 140 textbook, Chapter 15 *provides a helpful refresher.*

(i) **(1pt)** Compute LR(t) explicitly in terms of c.

[1] In Jupyter, you can download as PDF or print to save as PDF

(ii) **(3pt)** Let α be our false positive rate ($0 < \alpha < 1$). Compute the value of the threshold η so that the FPR of the test is equal to α. We say that such a test has *significance level* α. Your answer should be expressed in terms of α and c.

Hint: start by expressing the FPR as a conditional probability, then connect it to the LRT decision rule and the distributions f_0 and f_1.

(iii) **(2pt)** What is the *TPR* of this test? This is also known as the test's *power*. Your answer should be expressed in terms of α and c.

2. Online Experiments

In some applications of multiple testing, it is not possible to collect all p-values before making decisions about which hypotheses should be proclaimed discoveries. For example, when A/B testing a website, p-values arrive in a continual stream, so decisions have to be made in an online fashion, without knowing the p-values of future hypotheses. In this question, we compare an online algorithm for FDR control called LORD with the classical Benjamini-Hochberg (BH) procedure. We will provide an implementation of the LORD algorithm, however, for completeness, we also state the steps of the LORD algorithm below. Don't worry if you don't have intuition for the α_t update; the important thing is that such an update ensures that FDR is controlled at any given time t.

Algorithm 1 The LORD Procedure

```
input: FDR level $\alpha$, non-increasing sequence $\{\gamma_t\}_{t=1}^{\infty}$ such that $\sum_{t=1}^{\infty} \gamma_t = 1$, initial wealth
       $W_0 \leq \alpha$
Set $\alpha_1 = \gamma_1 W_0$
for $t = 1, 2, \ldots$ do
    $p$-value $P_t$ arrives
    if $P_t \leq \alpha_t$, reject $P_t$
    $\alpha_{t+1} = \gamma_{t+1} W_0 + \gamma_{t+1-\tau_1}(\alpha - W_0)\mathbf{1}\{\tau_1 < t\} + \alpha \sum_{j=2}^{\infty} \gamma_{t+1-\tau_j} \mathbf{1}\{\tau_j < t\}$,
    where $\tau_j$ is time of $j$-th rejection $\tau_j = \min\{k : \sum_{l=1}^{k} \mathbf{1}\{P_l \leq \alpha_l\} = j\}$
end
```

While offline algorithms like Benjamini-Hochberg take as input a *set* of p-values, online algorithms take in an *ordered sequence* of p-values. This makes their performance sensitive to p-value ordering. In this exercise we analyze this phenomenon.

We start by considering three different simulations, in which we have N real-valued observations. A fraction π_0 of them will be drawn from our null distribution, $\mathcal{N}(0, 1)$. The remainder will be drawn from our alternative distribution, $N(3, 1)$. For each point, we'll also keep track of which distribution it was truly drawn from (call this θ_i, where $\theta_i \in \{0, 1\}$). In order to compare the Benjamini-Hochberg and LORD algorithms, we'll generate p-values for our observations.

The three simulations will differ in whether the null observations occur (i) randomly throughout, (ii) all at the beginning, or (iii) all at the end. For each one, you should write a function that takes in the parameter π_0 and returns a simulated array of θ_i values and a simulated array of p-values.

The notation $\Phi(\cdot)$ refers to the CDF of a $\mathcal{N}(0, 1)$ variable.

(a) **(1pt)** When generating p-values under the null (i.e., when $\theta_i = 0$), explain why we can use $P_i \sim$ Unif$[0, 1]$ instead of sampling a $N(0, 1)$ RV and computing its CDF.
Hint: your answer should only be one sentence long.

(b) Now, write three functions of π_0 to generate $N = 1000$ p-values in the following three different ways:

(i) **(2pt)** Generate the p-values in random order. Here is the pseudocode you should follow: make sure you understand what the code is doing!

```
for $i = 1, \ldots, N$ :
  sample $\theta_i \sim$ Bernoulli$(1 - \pi_0)$
  if $\theta_i = 0$ : sample $P_i \sim$ Unif$[0, 1]$
  else : sample $Z_i \sim \mathcal{N}(3, 1)$, set $P_i = \Phi(-Z_i)$.
```

(ii) **(2pt)** Now, generate the p-values with all the null observations at the beginning. First sample $\pi_0 N$ p-values under the null distribution, and then sample the remaining ones assuming the alternative is correct. The pseudo-code is:

```
for $i = 1, \ldots, \pi_0 N$ :
  set $\theta_i = 0$, sample $P_i \sim$ Unif$[0, 1]$
for $i = \pi_0 N + 1, \ldots, N$ :
  set $\theta_i = 1$, sample $Z_i \sim \mathcal{N}(3, 1)$, set $P_i = \Phi(-Z_i)$.
```

(iii) **(2pt)** Finally, reverse the approach we adopted in the previous part: first sample $N - \pi_0 N$ p-values obtained under the alternative hypothesis, and then sample the remaining ones from a true null. The pseudo-code is:

```
for $i = 1, \ldots, N - \pi_0 N$ :
  set $\theta_i = 1$, sample $Z_i \sim \mathcal{N}(3, 1)$, set $P_i = \Phi(-Z_i)$
for $i = N - \pi_0 N + 1, \ldots, N$ :
  set $\theta_i = 0$, sample $P_i \sim$ Unif$[0, 1]$.
```

(iv) **(0pt, optional)** Vectorize your code for the previous three questions so that you don't use any for loops at all. Make sure your code is correct (see below) before you attempt to optimize it!

Remark: We have provided three functions defined in `test_q2b.py` to check if the three arrays of p-values that you created look correct. The functions are named `check_1`, `check_2` and `check_3`, and they take as input, respectively, the array of p-values generated in (ii), (iii) and (iv), and the value of π_0 you used.

(c) **(4pt)** Run the Benjamini-Hochberg procedure with $\alpha = 0.05$ for settings (i), (ii), (iii) on the whole batch; generate all of N p-values, and then apply BH. Compute the false discovery proportion (FDP) and sensitivity. Repeat this experiment 100 times to estimate FDR as the average FDP over 100 trials, as well as the average sensitivity. Do this for all $\pi_0 \in \Pi_0 := \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Make the following plots:

- FDR estimated over 100 trials on the y-axis against $\pi_0 \in \Pi_0$ on the x-axis, for the three different scenarios (i), (ii) and (iii).
- Expected sensitivity estimated over 100 trials on the y-axis against $\pi_0 \in \Pi_0$ on the x-axis, for the three different scenarios (i), (ii) and (iii).

What can you tell about the sensitivity and FDR of BH in the three different scenarios?

(c) (3pt) Now also run the LORD algorithm with $\alpha = 0.05$ on three p-value sequences, given as in (i), (ii) and (iii), respectively. Repeat the steps you have followed in part (b) and make the same plots. For which of the three scenarios (i), (ii), (iii) does LORD achieve highest average sensitivity? Can you give an intuitive explanation for this?

(d) (2pt) How do the sensitivity and FDR of BH compare to the sensitivity and FDR of LORD?