Homework 5: Continuous Random Variables

For each problem, remember you must briefly explain/justify how you obtained your answer, as correct answers without an explanation will not receive full credit. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide.

In general, your goal in an explanation is to write enough that a student from class who has attended lecture, but not read the problem yet, could understand your approach, verify your reasoning, and believe your answer is correct. While we do not usually need to see arithmetic, you must include enough work that in principle one could rederive your answer with only a scientific calculator.

Unless a problem states otherwise, you should leave your answer in terms of factorials, combinations, etc., for instance 26^7 or $26!/7!$ or $26 \cdot \binom{26}{7}$ are all good forms for final answers.

Instructions as to how to upload your solutions to gradescope are on the course web page.

Remember that you must tag your written problems on Gradescope.

Submission: You must upload a **pdf** of your written solutions to Gradescope under "HW 5 [Written]". (Instructions as to how to upload your solutions to gradescope are on the course web page.) The use of latex is *highly recommended*. (Note that if you want to hand-write your solutions, you'll need to scan them. We will take off points for hand-written solutions that are difficult to read due to poor handwriting and neatness.)

Your code for Problem 6 will be submitted under "HW 5 [Coding]" as a file called `min_hash.py`.

Optionally, your code for Problem 7 will be submitted under "HW 5 [Extra Credit Coding]" as a file called `bloom.py`.

Due Date: This assignment is due at 11:59 PM Wednesday August 6th.

Academic Integrity: Please read the full academic integrity policy. If you work with others (and you should!), you must still write up your solution independently and name all of your collaborators in the separate question on gradescope.

0. Extra Instructions :o

For calculations that require evaluating integrals (unless we indicate otherwise), you must

(a) Show the integral to evaluate (e.g., $\int_0^2 z \cdot 2dz$)

(b) Show an antiderivative and the values to evaluate at (e.g., $z^2|_0^2$)

(c) Plug in the values and simplify (e.g., $2^2 - 0^2 = 4$)

This is not a problem, so nothing needs to be submitted here.

1. The Classic Flea Problem [18 points]

A flea of negligible size is trapped in a large, spherical, inflated beach ball with radius b. At this moment, it is equally likely to be at any point within the ball. Let X be the distance of the flea from the center of the ball. For X, find

(a) the cumulative distribution function F. [5 points]

(b) the probability density function f. [5 points]

(c) the expected value. [4 points]

(d) the variance. [4 points]

Reminder: the volume of a sphere of radius r is $\frac{4}{3}\pi r^3$.

2. Exponential Darts [12 points]

You throw a dart at a circular target of radius r. Let X be the distance of your dart's hit from the center of the target. You have improved and your aim is such that $X \sim \text{Exponential}(5/r)$. (Note that it is possible for the dart to completely miss the target.)

(a) As a function of r, determine the value m such that $\mathbb{P}(X < m) = \mathbb{P}(X > m)$. [6 points]

(b) For $r = 15$, give the value of m to 3 decimal places. [2 points]

(c) What is the probability that you miss the target completely? Give your answer to 4 decimal places. [4 points]

3. Normal, normal, normal [12 points]

For each question below, if you need to use the CDF of a normal, be sure to round the z-score to the hundredths-place and use the table linked on the webpage.

(a) Suppose that X is normally distributed with mean 30 and standard deviation 26. Calculate the probability that $8 < X < 65$.

(b) The weight of a baby at birth is normally distributed with mean 3.25 kilograms and standard deviation 800 grams (approximately). What fraction of babies would you predict weigh more than 4.5 kilograms at birth?

4. Distinct Elements Analysis [5 points]

In this problem you will do some theoretical analysis for the code you will write in the next problem. We encourage you to read the motivation (and potentially the whole section) about the problem in the textbook for context.

Recall the setup for the problem: YouTube wants to count the number of ***distinct*** views for a video. By "distinct view" we mean the number of different people who have watched a video (so the same person watching a video three times counts only once). You could do this by storing User IDs for every user who has watched the video, but YouTube doesn't want to store all the user ID's, as that would use too much memory. The textbook describes a way to estimate this number.

We modeled the problem as follows: we see a **stream** of 8-byte integers (user ID's), $x_1, x_2, \ldots, x_N$, where x_i is the user ID of the i-th view to a video, but there are only n *distinct* elements ($1 \leq n \leq N$), since some people rewatch the video, even multiple times. We don't know what the number of views N is.

Let $U_1, \ldots, U_m$ be m iid samples from the continuous $\texttt{Unif}(0, 1)$ distribution, and let $X = \min\{U_1, \ldots, U_m\}$.

In this problem, you should compute $\text{Var}(X)$. For this problem, you may start with the CDF of X found on page 334 of the textbook, and you may use the fact from the textbook that $\mathbb{E}[X] = \frac{1}{m+1}$. You may find it helpful to do this computation yourself first for practice before moving on to the variance.

You may want to use WolframAlpha (or any similar tool) to solve any integrals along the way. For this problem, you **do not** need to show the antiderivative but you should **still show the integral to evaluate**.

5. Distinct Elements [Coding]

In this problem we are going to be writing the code to go with the Distinct Elements Analysis.

The scenario for the problem is the following:

YouTube wants to count the number of ***distinct*** views for a video, but doesn't want to store all the user ID's.

We modelled the problem as follows: we see a **stream** of 8-byte integers (user ID's), $x_1, x_2, \ldots, x_N$, where x_i is the user ID of the i^{th} view to a video, but there are only n *distinct* elements ($1 \leq n \leq N$), since some people rewatch the video, even multiple times. We don't know what the number of views N is.

Suppose the universe of user ID's is the set $\mathcal{U}$ (think of this as all 8-byte integers), and we have a single ***uniform*** hash function $h : \mathcal{U} \to [0, 1]$. That is, for an integer y, pretend $h(y)$ is a **continuous** $\texttt{Unif}(0, 1)$ random variable. That is, $h(y_1), h(y_2), ..., h(y_k)$ for any k **distinct** elements are iid continuous $\texttt{Unif}(0, 1)$ random variables, but since the hash function always gives the same output for some given input, if, for example, the i^{th} user ID x_i and the j-th user ID x_j are the same, then $h(x_i) = h(x_j)$ (i.e., they are the "same" $\texttt{Unif}(0, 1)$ random variable).

Pseudocode is provided which explains the two key functions that you will implement:

- `UPDATE(x)`: How to update your variable when you see a new stream element.
- `ESTIMATE()`: At any given time, how to estimate the number of distinct elements you've seen so far.

Your task for this problem is to implement the algorithm in python. Starter code is available on the associated Ed lesson.

6. Real-World: Modeling Assumptions [12 points]

The tools of this class are useful to computer scientists, but many of them are useful beyond just "classic" computer science. In order to use the powerful tools of probability, we need to make assumptions to let our mathematical tools model the real world. Things like "this coin is perfectly fair" or "the coin flips are all independent" are usually not perfectly true.[1] Indeed, occasionally these assumptions are ways that people "lie with statistics" or provide evidence for claims that aren't actually true.

In this question, you will critique the modeling assumptions made in an analysis.

Find an analysis (e.g., via a blog post/article) that uses probability and statistics tools you're familiar with from this course. By "analysis," we mean any estimate of a "real-world" probability, along with the assumptions that lead to that number. You might want to look at the examples in section 6.1 for what we mean.

We expect most of the answers to this section will be short (2-3 sentences), but you are free to write more if your resource is more complicated.

(a) Provide a link to (or somehow let us access) the analysis you're critiquing. [3 points]

(b) What is the fundamental claim of the analysis? I.e., what conclusion do they draw at the end of their analysis? [3 points]

(c) What modelling assumptions do they use? (For example, do they assume some occurrences are independent? Do they assume a set of events all have equal probability? Do they assume they know the probability? Do they use a variable from the zoo?) [3 points]

(d) Do you believe their analysis is correct? If so, choose an assumption from C and explain why you agree with it. If not, what assumption would you change? [3 points]

6.1. Some Ideas

You are free (and encouraged!) to find your own examples outside this list if you have a topic you are passionate about, but if you can't think of anything, you may use any of these as starting points. In many cases, there are already critiques of poor statistical/probability analyses online – it's ok to look at these critiques, as long as you tell us if you're using any and put everything in your own words.

- Some think that the probability of encountering at least one shiny Pokémon does not change with seeing more Pokémon.

[1] e.g., if you flip a coin repeatedly, the result of the last flip is probably how the coin will appear on your hand before you flip it, which will make the results not quite independent.

- Richard Lustig, 7-Time Lottery Winner, Gives Tips On Winning The Powerball Jackpot. You might find that his advice doesn't make the best assumptions. See article here.
- Is this an accurate estimate of the probability of being struck by lightning in the US? Could being in different states have an impact on this estimate?
- Every year millions of people predict the outcomes of the NCAA men's basketball tournament. It is commonly said that the probability of a perfect bracket is $\frac{1}{2^{63}}$, (since there are 2^{63} ways the 63 games could play out) and therefore no one will ever predict a perfect bracket. (Robbie Weber, an Allen School faculty, has a blog post about this one) Here is a source using that number

7. Bloom filters [Extra Credit Coding]

Google Chrome has a huge database of malicious URLs, but it takes a long time to do a database lookup (think of this as a typical `Set`). They want to have a quick check in the web browser itself, so a space-efficient data structure must be used. A **Bloom filter** is a **probabilistic data structure** which only supports the following two operations:

- `add(x)`: Add an element x to the structure.
- `contains(x)`: Check if an element x is in the structure. If either returns "definitely not in the set" or "could be in the set".

It does **not** support the following two operations:

- delete an element from the structure.
- return an element that is in the structure.

The idea is that we can check our Bloom filter to see if a URL is in the set. The Bloom filter is always correct in saying a URL definitely isn't in the set, but may have false positives – it may say that a URL is in the set when it isn't. Only in these rare cases does Chrome have to perform an expensive database lookup to know for sure.

Suppose that we have k **bit arrays** $t_1, \dots, t_k$ each of length m (all entries are 0 or 1), so the total space required is only km bits or $km/8$ bytes (as a byte is 8 bits). Suppose that the universe of URL's is the set $\mathcal{U}$ (think of this as all strings with less than 100 characters), and we have k ***independent and uniform*** hash functions $h_1, \dots, h_k : \mathcal{U} \to \{0, 1, \dots, m-1\}$. That is, for an element x and hash function h_i, pretend $h_i(x)$ is a **discrete** Unif$[0, m-1]$ random variable. Suppose that we implement the `add` and `contains` function as follows:

```
1: function initialize(k,m)
2:     for $i = 1, \dots, k$: do
3:         $t_i$ = new bit array of m 0's
4: function add(x)
5:     for $i = 1, \dots, k$: do
6:         $t_i[h_i(x)] = 1$
7: function contains(x)
       return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$
```

Refer to Section 9.4 of the textbook for more details on Bloom filters. Starter code is available on the associated Ed lesson.

8. Feedback + Collaboration [1 point]

Answer these questions on the separate Gradescope box for this question.

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- Which students did you collaborate with for this homework?

- Is the work that you are submitting your own and does not violate the academic integrity policy outlined in the syllabus?
- How many hours did you spend working on this assignment (excluding any extra credit questions, if applicable)? Report your estimate to the nearest hour.
- Which problem did you spend the most time on?
- Any other feedback for us?