CSE 312 – Problem Set 7

Due Wednesday, May 27, 11:59pm

Spring 2026

Instructions

Solutions format and late policy. See PSet 1 for details on expectations for solutions, collaboration policy, late policy, etc. The same requirements and policies still apply. Also follow the typesetting instructions from the prior PSets.

Solutions submission. You must submit your solution via Gradescope. In particular:

- Problem 1 will be done on Gradescope.
- Submit the solutions to problems 2-6 under "PSet 7 [Written]" as on previous problem sets. This will be a *single* PDF file containing the solution to problems 2-6 in the homework. Each numbered task should be solved on its own page (or pages). Follow the prompt on Gradescope to link tasks to your pages. Do not write your name on the individual pages – Gradescope will handle that.
- For the programming part (Problem 7), submit your code under "PSet 7 [Coding]" as a file called `min_hash.py`.

1 Gradescope Questions (15 points)

Please do these gradescope questions.

2 The Leetcode Grind (25 points)

You are grinding Leetcode to prepare for software engineering interviews for H hours, where H is a random variable, **equally likely** to be 1, 2 or 3. The number of `Difficult` problems D that you successfully solve is random and depends on how long you practice. We are told that

$$\mathbb{P}(D = d \mid H = h) = \frac{1}{h}, \qquad \text{for } d = 1, \dots, h\ .$$

a) (5 points) Find the joint distribution of D and H. (Hint: use the chain rule).

b) (5 points) Find the marginal distribution of D.

c) (5 points) Find the conditional distribution of H given that $D = 1$ (that is, $\mathbb{P}(H = h \mid D = 1)$ for each possible h in $\{1, 2, 3\}$). Use the definition of conditional probability and the results from previous parts.

d) (10 points) Suppose that we are told that you solved 1 or 2 `Difficult` problems. Note that these are two mutually exclusive events. Find the expected number of hours you practiced conditioned on this event, defined as follows:

$$\mathbb{E}\left[H \mid D = 1 \cup D = 2\right] = \sum_{h=1}^{3} h \cdot \mathbb{P}(H = h \mid D = 1 \cup D = 2)$$

3 Joint densities I (16 points)

a) (4 points) Let Z be a uniformly random point in the unit radius disk centered at the origin of a plane in two dimensions. Let X (resp. Y) be the x-coordinate (respectively y-coordinate) of Z. Give a simple argument (of at most a few sentences) to show that X and Y are not independent.

b) (12 points) Suppose that X and Y (not the same as in the previous part of this problem) have the following joint density:

$$f_{X,Y}(x,y) = \begin{cases} a & 0 < x \leq 0.5,\ 0 < y \leq 0.5 \\ b & 0 < x \leq 0.5,\ 0.5 < y < 1 \\ b & 0.5 < x \leq 1,\ 0 \leq y \leq 0.5 \\ a & 0.5 < x < 1,\ 0.5 < y < 1, \end{cases}$$

where a and b are constants.
- For what values of a and b are the marginal distributions of X and Y uniform on (0,1)?
- For what values of a and b are X and Y independent?
Note that some of your answers may simply be restrictions on the values a and b can take on, rather than specific values. Justify your answers.

4 Joint Densities II (10 points)

Let X, Y, Z be independent and uniformly distributed over (0,1) Compute

$$\mathbb{P}(X \geq YZ).$$

Your answer should be a number. Hint: You just need to integrate over the relevant region of the joint density $f_{X,Y,Z}(x,y,z)$. Carefully choose the order of integration.

5 Sticks I (10 points)

We are given a line segment, $[0,1]$. Two darts are each independently thrown uniformly at random within the line segment. What is the probability that the value of the first dart is at least three times the value of the other? Hint: Use the continuous law of total probability, conditioning on the value of the other dart.

6 Covariance (15 points)

(We may or may not talk about covariance in class, but it is important for you to know about.) Also, note that some portions of this problem are covered in Section 5.3 of the book. Regardless of that fact, be sure you include all details of your derivations for the following questions. For any two random variables X, Y the *covariance* is defined as

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])].$$

In this problem, if you prefer, you may assume that X and Y are discrete random variables.

a) (3 points) Show that

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y].$$

b) (3 points) Show that for any two random variables

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y).$$

c) (3 points) If $\mathbb{E}[Y|X = x] = x$ show that $\text{Cov}(X, Y) = \text{Var}(X)$.

d) (3 points) If X, Y are independent show that $\text{Cov}(X, Y) = 0$.

e) (3 points) If X and Y have $\text{Cov}(X, Y) > 0$, we say that X and Y are positively correlated. If $\text{Cov}(X, Y) < 0$, we say that X and Y are negatively correlated. Suppose that $\Omega_X = \{0, 1\}$, $\Omega_Y = \{0, 1\}$ and $\Omega_{X,Y} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Give a valid joint probability mass function for X and Y for which X and Y are positively correlated. Then give a different joint probability mass function for X and Y (same ranges) for which X and Y are negatively correlated.

7 Distinct Elements [Coding] (20 points)

Recall the setup for the `MinHash` algorithm presented in class. The universe of is the set $\mathcal{U}$ (think of this as the set of 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** Unif$(0, 1)$ random variable. That is, $h(x_1), h(x_2), ..., h(x_N)$ for any N **distinct** elements are iid continuous 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" Unif$(0, 1)$ random variable).

Then, the `MinHash` algorithm is realized by the following pseudocode, which explains its two key functions:

a) `update(`x`)`: How to update your variable when you see a new stream element.

b) `estimate()`: At any given time, how to estimate the number of distinct elements you've seen so far.

Note that this differs from the syntax used on the slides, but captures the same algorithm.
MinHash Operations

function initialize()
  val $\leftarrow \infty$

function update(x)
  val $\leftarrow \min\{\texttt{val}, h(x)\}$

function estimate()
  return round $\left(\frac{1}{\texttt{val}} - 1\right)$

for $i = 1, \ldots, N$: *// Loop through all stream elements*
  update(x_i) *// Update our single float variable*

return estimate() *// An estimate for n, the number of distinct elements.*

To help you out with the following questions, we have set up an edstem lesson. However, you are required to upload your final solution to Gradescope (see instructions above).

a) Implement the functions **update** and **estimate** in the `MinHash` class of `min_hash.py`.

b) The estimator we used in part a) has high variance, and therefore it may not always give good answer. As outlined in class, we improve this by considering k variables

$$\text{val}_1, \text{val}_2, \ldots, \text{val}_k$$

where each of val_i, $1 \leq i \leq k$ is an i.i.d. random variable with the distribution of the minimum of $m \leq N$ independent Unif$(0, 1)$ variables, obtained by hashing the N elements in the stream with independent hash functions $h^1, \ldots, h^k$. Our final estimate will then be

$$\hat{n} = \text{round}\left(\frac{1}{\widehat{\text{val}}} - 1\right) \quad \text{where} \quad \widehat{\text{val}} = \frac{1}{k}\sum_{i=1}^{k} \text{val}_i.$$

Implement the functions **update** and **estimate** in the `MultMinHash` class of `min_hash.py` using the improved estimator.

Refer to Section 9.5 of the book for more details on the distinct elements algorithm.