CSE 312: Foundations of Computing II Winter 2024

Problem Set 8

Due: Wednesday, March 6, by 11:59pm (Tasks 1-7)

Task 9 [Coding] is due on Friday, March 8 at 11:59pm

Instructions

Solutions format and late policy. See PSet 1 for further details. The same requirements and policies still apply. Also follow the typesetting instructions from the prior PSets.

Collaboration policy. The written problems (Tasks 1-7) on this pset may be done with a **single partner**. In this case, only one person will submit the written part on Gradescope and add their partner as a collaborator. You must do Task 9 on your own.

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

- For the solutions to Tasks 1-7, submit under "PSet 8 [Written]" a **single** PDF file containing the solutions to Tasks 1-7 (for you and your partner). 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 names on the individual pages – Gradescope will handle that. *Tasks 1-7 are due on Wednesday, March 6th at 11:59*
- Task 8 is purely optional and will not be graded, so no need to submit.
- For the programming part (Task 9), submit your code under "PSet 8 [Coding]" as a file called min_hash.py. *Task 9 is due on Friday, March 8th at 11:59*, and late days are counted separately for this task.

Task 1 – Sticks [10 pts]

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.

Task 2 – More Sticks [10 pts]

A stick of length 1 is broken at a uniformly random position. What is the expected length of the shorter part of the stick?
Hint: Use the law of total expectation conditioning on the position where the stick is broken.

Task 3 – Collisions [15 pts]

A total of m items are to be sequentially inserted into a hash table of size n, where each item is mapped independently but this time, the items are not mapped uniformly into the table. Instead, an item is mapped to cell j of the table with probability p_j, for $j = 1, \ldots, n$ (where $\sum_{j=1}^{n} p_j = 1$). We say that a collision occurs whenever an item is mapped into a nonempty cell. Find $E(X)$, where X is the number of collisions that occur during this process. You can leave your answer as an unsimplified sum.
Hint: Write $X = X_1 + X_2 + \ldots + X_m$ where X_i is an indicator random variable that is 1 if there is a collision when the i-th item is put into a cell and 0 otherwise; then use linearity of expectation. To compute $E(X_i)$, use the law of total expectation, conditioning on which cell it is placed into.

Task 4 – Ducks [15 pts]

Ten hunters are waiting for ducks to fly by. A flock of ducks flies overhead with the number of ducks in the flock a Poisson random variable with mean 5. Suppose that each hunter chooses one of the ducks to aim at uniformly at random and independent of the choices of the other hunters. The hunters all fire at the same time. If each hunter independently hits their chosen target with probability 0.3, use the law of total expectation to compute the expected number of ducks that are hit.

Task 5 – Lazy Grader [12 pts]

Prof. Lazy decides to assign final grades in CSE 312 by ignoring all the work the students have done and instead using the following probabilistic method: each student independently will be assigned an A with probability θ, a B with probability 3θ, a C with probability $\frac{2}{3}$, and an F with probability $\frac{1}{3} - 4\theta$. When the quarter is over, you discover that only 10 students got an A, 35 got a B, 40 got a C, and 15 got an F.

Find the maximum likelihood estimate for the parameter θ that Prof. Lazy used. Give an exact answer as a simplified fraction. You do not need to check second order conditions.

Task 6 – Continuous MLE [24 pts]

You do not need to check second order conditions in the following.

a) Let $x_1, x_2, \ldots, x_n$ be independent samples from an exponential distribution with unknown parameter λ. What is the maximum likelihood estimator for λ?

b) Given $\theta > 0$. Suppose that $x_1, \ldots, x_n$ are i.i.d. realizations (aka samples) from the model

$$f(x;\theta) = \begin{cases} \theta x^{\theta-1} & 0 < x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Find the maximum likelihood estimate for θ.

Task 7 – (Un)biased Estimation [10 pts]

Let $x_1, \ldots, x_n$ be independent samples from Unif$(0, \theta)$, the continuous uniform distribution on $[0, \theta]$. Then, consider the estimator $\hat{\theta}_{\text{first}} = 2x_1$, i.e., our estimator ignores the samples $x_2, \ldots, x_n$ and just outputs twice the value of the first sample.
Is $\hat{\theta}_{\text{first}}$ unbiased?

Task 8 – Covariance – extra problem for your benefit only [0 pts]

We have unfortunately run out of time to spend a proper amount of time on the important concept of *covariance*. If you'd like to get ahead of the game, we highly recommend doing this problem. (E.g., this concept will be used in the ML class, CSE 446). This problem will **NOT be graded.** Solutions will be posted though.

Note that some portions of this problem are covered in Section 5.3 of the book. 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) Show that

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

b) Show that for any two random variables

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

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

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

e) 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.

Task 9 – Distinct Elements [Coding] (Due March 8, 11:59pm) [20 pts]

Note: This task is due Friday, March 8. Late days calculated separately for this task.
Consider the setup for the MinHash algorithm that will be presented in class on Friday, March 1. 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** $\text{Unif}(0,1)$ random variable. That is, $h(x_1), h(x_2), ..., h(x_N)$ for any N **distinct** elements are iid continuous $\text{Unif}(0,1)$ 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" $\text{Unif}(0,1)$ random variable).

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

1. $\text{UPDATE}(x)$: How to update your variable when you see a new stream element.
2. $\text{ESTIMATE}()$: 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 $\leftarrow \min\{\text{val}, h(x)\}$
function ESTIMATE() **return** round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▷ 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 **a)** has high variance, and therefore it may not always give good answer. As outlined in class, we improve this by considering k variables

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

where each of val_i, $1 \leqslant i \leqslant k$ is an i.i.d. random variable with the distribution of the minimum of $m \leqslant N$ independent $\mathrm{Unif}(0,1)$ 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

$$\hat{n} = \frac{1}{\widehat{\mathsf{val}}} - 1 \quad \text{where} \quad \widehat{\mathsf{val}} = \frac{1}{k}\sum_{i=1}^{k} \mathsf{val}_i.$$

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.