Homework_11

November 23, 2019

1 Homework 11

1.0.1 Instructions

Your homeworks have two components: a written portion and a portion that also involves code. Written work should be completed on paper, and coding questions should be done in the notebook. You are welcome to LaTeX your answers to the written portions, but staff will not be able to assist you with LaTeX related issues. It is your responsibility to ensure that both components of the homework are submitted completely and properly to Gradescope. Refer to the bottom of the notebook for submission instructions.

```
In [1]: from datascience import *
        from prob140 import *
        import numpy as np
        import matplotlib.pyplot as plt
        plt.style.use('fivethirtyeight')
        %matplotlib inline
        import math
        from scipy import stats
```

1.0.2 1. Functions of Exponential Random Variables

Let X and Y be independent exponential random variables with rates λ and μ respectively.

a) Let $W = \min(X, Y)$. Find the distribution of W. Start by remembering which of the following is easiest when working with minima: the density, the cdf, or the survival function.

b) Let c be a positive constant. Use the results about exponential random variables in Sections 16.1 and 17.2 of the Prob 140 textbook to find $P(X > cY)$ without integration.

c) Use part (b) to find the cdf of $\frac{X}{Y}$.

1.0.3 Solution

1a) We have known since early in the course that for minima the right tail (survival function) is the easiest to work with. For $w > 0$, $P(W > w) = e^{-\lambda w}e^{-\mu w} = e^{-(\lambda+\mu)w}$

This is the survival function of the exponential $(\lambda + \mu)$ distribution. So the distribution of W is exponential $(\lambda + \mu)$.

1b) From Section 16.1 we know that cY has the exponential (μ/c) distribution.

From Section 17.2 we know that if X_1 is exponential (λ_1) and X_2 is exponential (λ_2) independent of X_1, then $P(X_1 > X_2) = \lambda_2/(\lambda_1 + \lambda_2)$.

So $P(X > cY) = \frac{\mu/c}{\mu/c+\lambda} = \frac{\mu}{\mu+c\lambda}$.

1c) Let F be the cdf of X/Y. Then for $w > 0$, $F(w) = P(\frac{X}{Y} < w) = P(X < wY)$ because $Y > 0$. Since wY has the exponential μ/w distribution and is independent of X, $F(w) = P(X < wY) = \lambda/(\mu/w + \lambda) = w\lambda/(\mu + w\lambda)$.

1.0.4   2. Gamma Tail Bound

Let X have the gamma (r, λ) distribution.

a) Show that $P(X \geq 2E(X)) \leq (2/e)^r$.

[Remember that if h is a positive function of x, then the value of x that minimizes h is the same as the value of x that minimizes $\log(h)$.]

b) [CODE] Fix $\lambda = 1$. Display overlaid plots of the following three graphs as functions of r, for r in the interval $(0.5, 15)$:

- $P(X \geq 2E(X))$
- $(2/e)^r$
- Markov's bound on $P(X \geq 2E(X))$

The code uses `plt.plot` which is also used in Lab 9. The expression `stats.gamma.cdf(x, r, scale=1)` evaluates to the cdf of the gamma $(r, 1)$ distribution at the point x.

2a) $P(X \geq 2E(X)) = P(X \geq c)$ where $c = 2E(X) = 2\frac{r}{\lambda}$

$$P(X \geq c) = P(e^{tX} \geq e^{tc}) \leq \frac{M_X(t)}{e^{tc}} = \frac{\left(\frac{\lambda}{\lambda-t}\right)^r}{e^{tc}} \text{ for } t < \lambda \text{ (Chernoff's Bound)}$$

We want to minimize $\frac{\left(\frac{\lambda}{\lambda-t}\right)^r}{e^{tc}}$ over $0 < t < \lambda$

This is the same as minimizing $\log \frac{\left(\frac{\lambda}{\lambda-t}\right)^r}{e^{tc}}$ over $0 < t < \lambda$ over $0 < t < \lambda$

$\log \frac{\left(\frac{\lambda}{\lambda-t}\right)^r}{e^{tc}} = r \log \frac{\lambda}{\lambda-t} - \log e^{tc} = r \log \lambda - r \log(\lambda - t) - tc$

Take the derivative with respect to t, set to 0, and solve for the minimizing value t^*:

$\frac{r}{\lambda-t^*} - c = 0$

$\frac{r}{\lambda-t^*} = c = \frac{2r}{\lambda}$

$\frac{\lambda}{2} = \lambda - t^*$

$t^* = \frac{\lambda}{2}$

Plugging t^* into $\frac{\left(\frac{\lambda}{\lambda-t}\right)^r}{e^{tc}}$, we get $\frac{2^r}{e^{\frac{\lambda}{2}\cdot\frac{2r}{\lambda}}} = (2/e)^r$

Therefore, $P(X \geq 2E(X)) \leq (2/e)^r$

```
In [2]: # Answer to 2b
        r = np.arange(0.05, 15, 0.1)

        markov_bound = 0.5*np.ones(len(r))

        part_a_bound = (2/np.exp(1))**r

        exact = make_array()
        for i in r:
```
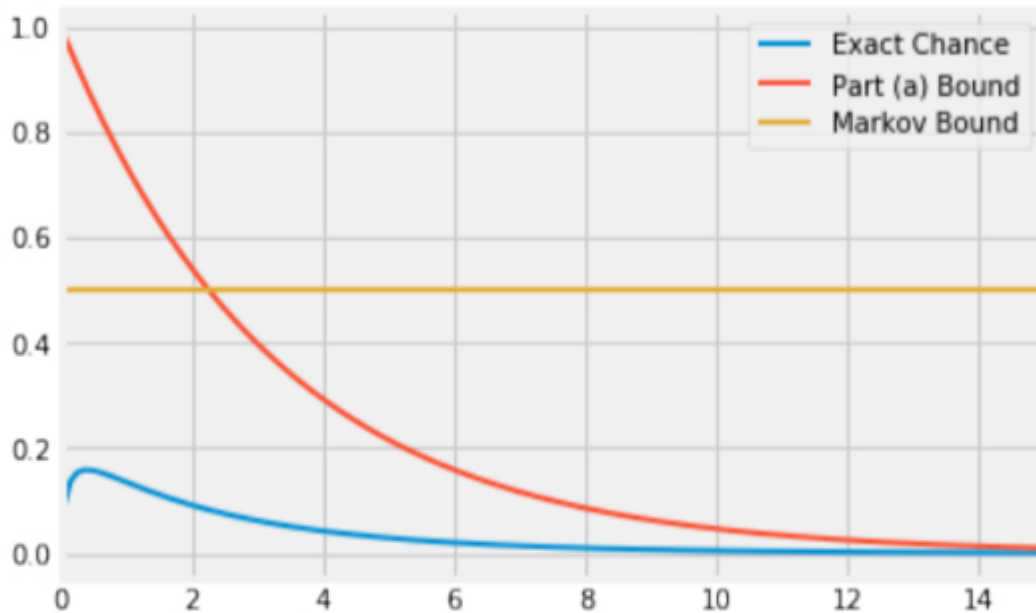
2

```
exact = np.append(exact, 1 - stats.gamma.cdf(2*i, i, scale=1))

plt.plot(r, exact, lw=2, label='Exact Chance')
plt.plot(r, part_a_bound, lw=2, label='Part (a) Bound')
plt.plot(r, markov_bound, lw=2, label='Markov Bound')
plt.legend()
plt.xlim(0, 15);
```


c) For which values of r is the bound in part (a) better than Markov's bound? Show the analytical calculation. Then use a code cell to find the numerical range and confirm that it is consistent with the graphs in (b).

2c) The bound in part (a) is better than the Markov bound for r values greater than the value for r when the two bounds are equal (see the graph from 2b).

$(2/e)^r = 0.5$
$(2e^{-1})^r = 0.5$
$\log(2e^{-1})^r = \log 0.5$
$r \log(2e^{-1}) = \log 0.5$
$r(\log 2 - 1) = \log 0.5$
$r \geq \frac{\log 0.5}{\log 2 - 1}$

```
In [3]:  # Calculation for 2c
         np.log(0.5)/(np.log(2) - 1)

Out[3]:  2.258891353270929
```

1.0.5   3. Waiting for a Random Coin to Land Heads

Let X be a random proportion. Given $X = p$, let T be the number of tosses till a p-coin lands heads.

a) Let $P(X = 1/10) = 1/4$, $P(X = 1/7) = 1/4$, and $P(X = 1/3) = 1/2$. Find $E(T)$.

b) Find $E(T)$ if X has the beta (r, s) density for some $r > 1$. Simplify all integrals and Gamma functions in your answer.

c) Let X have the beta (r, s) density. For fixed $k > 0$, find the posterior density of X given $T = k$. Identify it as one of the famous ones and state its name and parameters.

1.0.6   Solution

3a) Given $X = x$, the conditional distribution of T is geometric (x).

So $E(T \mid X = x) = \frac{1}{x}$ and hence $E(T|X) = \frac{1}{X}$

$E(T) = E(E(T|X)) = E(\frac{1}{X}) = 10 \cdot \frac{1}{4} + 7 \cdot \frac{1}{4} + 3 \cdot \frac{1}{2} = \frac{23}{4}$

3b) Again, $E(T|X) = \frac{1}{X}$

$E(T) = E(E(T|X)) = E(\frac{1}{X}) = \int_0^1 \frac{1}{x} \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} x^{r-1}(1-x)^{s-1} dx = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \int_0^1 x^{r-2}(1-x)^{s-1} dx =$

$\frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \int_0^1 x^{(r-1)-1}(1-x)^{s-1} dx$

The integral is a beta integral because $r > 1$ and hence $r - 1 > 0$. So

$E(T) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \cdot \frac{\Gamma(r-1)\Gamma(s)}{\Gamma(r-1+s)} = \frac{\Gamma(r+s)}{\Gamma(r)} \cdot \frac{\Gamma(r-1)}{\Gamma(r-1+s)}$

Use $\Gamma(w + 1) = w\Gamma(w)$ to get

$E(T) = \frac{\Gamma(r+s)}{\Gamma(r)} \cdot \frac{\Gamma(r-1)}{\Gamma(r-1+s)} = \frac{r+s-1}{r-1}$

3c) For $0 < x < 1$, $f_{X|T=k}(x) \propto x^{r-1}(1-x)^{s-1}(1-x)^{k-1} x = x^{(r+1)-1}(1-x)^{(s+k-1)-1}$

So the posterior density is beta $(r + 1, s + k - 1)$. The updating is the same as with the binomial likelihood: add the number of heads to r and add the number of tails to s.

1.0.7   4. Discrete and Continuous Random Selections

Fix a positive integer n, and let p be strictly between 0 and 1.

Suppose Person A picks a number uniformly in the interval $(0, n)$. Let X be the number Person A picks.

Suppose that independently of Person A, Person B picks an integer Y according to the binomial (n, p) distribution, for example by using `stats.binom.rvs(n, p, size=1)` or by tossing a p-coin n times and recording the number of heads.

Find $P(X < Y)$.

For each $y \in \{0, 1, 2, \dots, n\}$, $P(X < Y \mid Y = y) = \frac{y}{n}$. So $P(X < Y) = E(P(X < Y \mid Y)) = E(\frac{Y}{n}) = p$

Another way of writing this:

$P(X < Y) = \sum_y P(Y = y)P(X < Y \mid Y = y) = \sum_{y=0}^{n} \frac{y}{n} P(Y = y) = E(\frac{Y}{n}) = p$

1.0.8   5. Maximum Likelihood Estimate of the Exponential Rate

For $n > 1$, let $X_1, X_2, \dots, X_n$ be i.i.d. exponential (λ) variables.

a) Let $\hat{\lambda}_n$ be the maximum likelihood estimate (MLE) of the parameter λ. Find $\hat{\lambda}_n$ in terms of the sample mean $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. The subscript n in $\bar{X}_n$ is there to remind us that we have the average of n values. It doesn't refer to the nth sampled value X_n.

4

5a)

$$Lik(\lambda) = \prod_{i=1}^{n} \lambda e^{\lambda X_i} = \lambda^n e^{-\lambda \sum_{i=1}^{n} X_i}$$

$$L(\lambda) = \log \lambda^n + \log e^{-\lambda \sum_{i=1}^{n} X_i} = n \log \lambda - \lambda \sum_{i=1}^{n} X_i$$

$$\frac{d}{d\lambda} L(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^{n} X_i = 0$$

$$\frac{n}{\hat{\lambda}_n} = \sum_{i=1}^{n} X_i$$

$$\hat{\lambda}_n = \frac{n}{\sum_{i=1}^{n} X_i} = \frac{1}{\bar{X}_n}$$

b) This part is a simulation study in which you already know λ and are examining how close your estimate comes to this known value. Take $\lambda = 0.25$. We have assigned it the name `lam` because `lambda` means something else in Python.

Repeat the following simulation 10,000 times. - Generate a sample of 400 i.i.d. exponential (`lam`) variables. Use `stats.expon.rvs(scale = 1/lam, size=400)`. - Calculate your MLE $\hat{\lambda}_{400}$ based on this sample.

Then draw the histogram and find the mean of your 10,000 MLEs.

In [4]: # Your answer to 5b

```
lam = 0.25

# Define a function that returns the MLE based on
# one i.i.d. sample of size 400
# from the exponential distribution with lambda = 0.25
def mle_from_one_sample():
    sample = stats.expon.rvs(scale=1/lam, size=400)
    return 1 / np.mean(sample)

# Construct an array of 10,000 MLEs
mle_400 = make_array()
for i in range(10000):
    mle_400 = np.append(mle_400, mle_from_one_sample())

# Plot a histogram of the 10,000 MLEs
Table().with_column('MLE', mle_400).hist(bins=20)
```

5
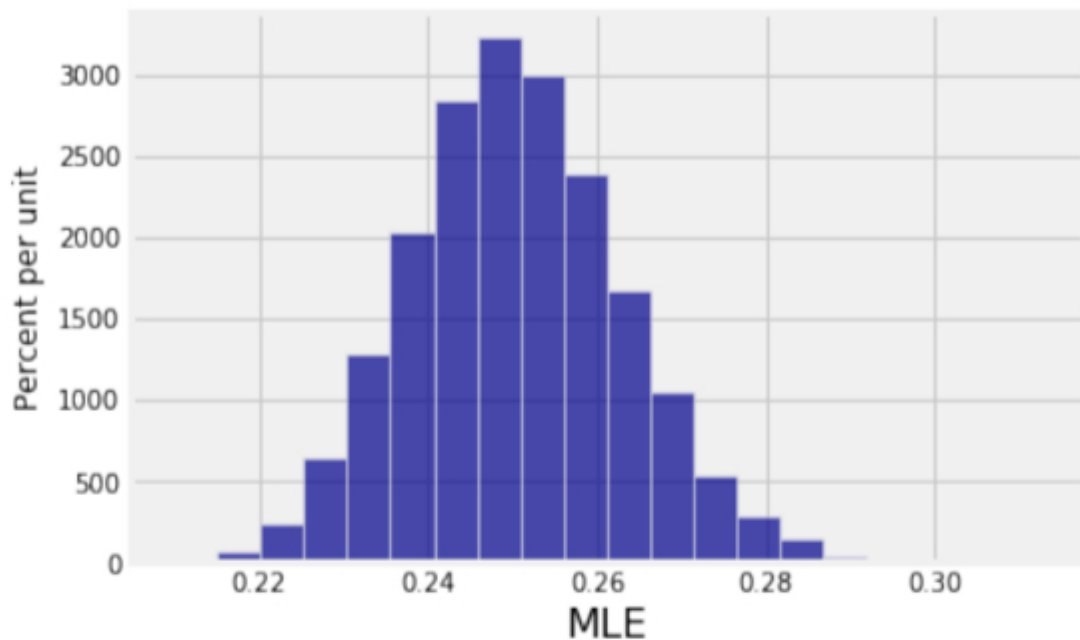
c) Use facts about sums and linear transformations to find the distribution of $\bar{X}_n$ with little or no calculation. Recognize it as one of the famous ones and provide its name and parameters. Use it to find $E(\hat{\lambda}_n)$.

d) Is $\hat{\lambda}_n$ an unbiased estimate of λ? If it is biased, does it overestimate on average, or does it underestimate? Is it asymptotically unbiased? That is, does $E(\hat{\lambda}_n)$ converge to λ as $n \to \infty$?

e) Your simulation in **b)** should show an approximately normal distribution for $\hat{\lambda}_n$. The Central Limit Theorem says the distribution of $\bar{X}_n$ should be approximately normal, but is $\hat{\lambda}_n$ a linear function of $\bar{X}_n$? If not, why is its distribution approximately normal?

[This one needs a little thought but no calculation. Think about what a smooth function looks like over a small interval in its domain, and then think about why small intervals are relevant here.]

5c) $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n}X_i = \frac{1}{n}S_n$.

S_n has gamma (n, λ) distribution because the X_i's are i.i.d. gamma $(1, \lambda)$ random variables.

We know that for $c > 0$, cS_n has gamma $(n, \lambda/c)$ distribution.

Set $c = \frac{1}{n}$ to see that $\bar{X}_n$ has the gamma $(n, n\lambda)$ distribution.

$E(\hat{\lambda}_n) = E(\frac{1}{\bar{X}_n}) = \int_0^\infty \frac{1}{y}\frac{(n\lambda)^n}{\Gamma(n)}y^{n-1}e^{-n\lambda y}dy = \frac{(n\lambda)^n}{\Gamma(n)}\int_0^\infty y^{(n-1)-1}e^{-n\lambda y}dy = \frac{(n\lambda)^n}{\Gamma(n)}\cdot\frac{\Gamma(n-1)}{(n\lambda)^{n-1}} = \frac{n\lambda}{n-1}$

So, $E(\hat{\lambda}_n) = \frac{n}{n-1}\lambda$

5d) Looking at part c, we see that $E(\hat{\lambda}_n) \neq \lambda$, so is not an unbiased estimator. It overestimates λ on average ($\frac{n}{n-1} > 1$).

It is asymptotically unbiased because $E(\hat{\lambda}_n) = \frac{n}{n-1}\lambda \to \lambda$ as $n \to \infty$

5e) We know that linear functions of normal variables are normal. By Taylor's theorem, every smooth function is approximately linear over small intervals. When n is large, the distribution of $\bar{X}_n$ is approximately normal and is concentrated on a small interval because $SD(\bar{X}_n)$ has $\sqrt{n}$ in the denominator. Over that small interval, the graph of $g(x) = 1/x$ looks roughly linear. So with overwhelming probability, $\hat{\lambda}_n$ behaves like an approximately linear function of an approximately normal variable. So the distribution of $\hat{\lambda}_n$ is approximately normal.

1.1 Submission Instructions

Many assignments throughout the course will have a written portion and a code portion. Please follow the directions below to properly submit both portions.

1.1.1 Written Portion

- Scan all the pages into a PDF. You can use any scanner or a phone using an application. Please **DO NOT** simply take pictures using your phone.
- Please start a new page for each question. If you have already written multiple questions on the same page, you can crop the image or fold your page over (the old-fashioned way). This helps expedite grading.
- It is your responsibility to check that all the work on all the scanned pages is legible.

1.1.2 Code Portion

- Save your notebook using File > Save and Checkpoint.
- Generate a PDF file using File > Download as > PDF via LaTeX. This might take a few seconds and will automatically download a PDF version of this notebook.

    - If you have issues, please make a follow-up post on the general HW 11 Piazza thread.

1.1.3 Submitting

- Combine the PDFs from the written and code portions into one PDF. Here is a useful tool for doing so.
- Submit the assignment to Homework 11 on Gradescope.
- **Make sure to assign each page of your pdf to the correct question.**
- **It is your responsibility to verify that all of your work shows up in your final PDF submission.**

1.1.4 We will not grade assignments which do not have pages selected for each question.