MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.s077 Spring 2018

Partial Lecture 2 Notes and **Problem Set 1** due Thursday 2/15, in class

---

Readings: Slides from Lectures 2. We do not provide detailed notes for Lecture 2. However, the problem statements below do sketch many of the concepts in Lecture 2 in a self-contained manner, and can be viewed as a lecture-note/problem-set combination.

Note: The solutions below, are the solutions to the theoretical part of the assignment. Solution to any subpart, that has a computational piece, can be found in the corresponding pdf files, uploaded on Stellar.

Problem 1. The relation between true and empirical distributions.

(a) Let A_i be the event that, $|\hat{P}_i - p_i| \leq \epsilon$, for each $i = 1, 2, \ldots, k$. Note that, A_i has a dependence on n, through the empirical distribution. Our task, is to show that,

$$\lim_{n\to\infty} \mathbb{P}\left(\bigcap_{i=1}^{k} A_i\right) = 1,$$

under the knowledge that,

$$\lim_{n\to\infty} \mathbb{P}(A_i^c) = 0.$$

Now,

$$\begin{aligned}
\lim_{n\to\infty} \mathbb{P}\left(\bigcap_{i=1}^{k} A_i\right) &= \lim_{n\to\infty} \left(1 - \mathbb{P}\left(\bigcup_{i=1}^{k} A_i^c\right)\right) \\
&\geq 1 - \lim_{n\to\infty} \left(\sum_{i=1}^{k} \mathbb{P}(A_i^c)\right) \\
&= 1 - \sum_{i=1}^{k} \left(\lim_{n\to\infty} \mathbb{P}(A_i^c)\right) \\
&= 1,
\end{aligned}$$

where, the second line, namely the inequality, follows from the union bound.

(b) Fix an $x \in \mathbb{R}$, and define the random variables, $I_1, I_2, \ldots, I_j$ via,

$$I_j = \begin{cases} 1, & \text{if } X_j \leq x, \\ 0, & \text{otherwise.} \end{cases}$$

Since, X_j's are all i.i.d., so do I_j's. Note that, $\mathbb{P}(I_j = 1) = \mathbb{P}(X_j \leq x) = F(x)$. Moreover, $\mathbb{E}[I_j] = F(x)$.

Using these auxiliary random variables, we can write $\hat{F}(x)$ in the following way:

$$\hat{F}(x) = \frac{1}{n} \sum_{j=1}^{k} I_j.$$

Hence, using the (weak) law of large numbers, we have, $\hat{F}(x)$ converges to $F(x)$ in probability, namely,

$$\lim_{n \to \infty} \mathbb{P}\left(\hat{F}(x) - F(x) > \epsilon\right) = 0.$$

Note that, having observed the random variables $X_1, X_2, \ldots, X_n$; $\hat{F}(x)$ corresponds to the empiric frequency of the outcome 1, of the random variables, $I_1, I_2, \ldots, I_n$.

Problem 2.

(a) Since the random variables, $X_1, X_2, \ldots, X_n$ are all i.i.d., their joint distribution, $f^{\theta}_{X_1,\ldots,X_N}(x_1, \ldots, x_n)$, factors as a product, namely,

$$f^{\theta}_{X_1,\ldots,X_N}(x_1, \ldots, x_n) = \prod_{i=1}^{n} f^{\theta}_{X_i}(x_i) = \prod_{i=1}^{n} f^{\theta}_{X}(x_i) = \prod_{i=1}^{n} \theta e^{-\theta x_i} = \theta^n e^{-\theta \sum_{i=1}^{n} x_i}.$$

After taking logarithms,

$$\log f^{\theta}_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = n \log \theta - \theta \left(\sum_{i=1}^{n} x_i \right),$$

and therefore,

$$\frac{d}{d\theta} \log f^{\theta}_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \frac{n}{\theta} - \sum_{i=1}^{n} x_i.$$

If we set the last expression equal to 0, we get,

$$\hat{\theta}_{ML} = \frac{n}{x_1 + x_2 + \cdots + x_n},$$

as desired.

(b) Let the observed values of $X_1, X_2, \ldots, X_n$ be, $x_1, \ldots, x_n$, respectively. We have,

$$\hat{\mathbb{E}}[X] = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \mathbb{E}^{\theta}[X] = \frac{1}{\theta}.$$

Therefore,

$$\hat{\theta}_1 = \frac{n}{x_1 + x_2 + \cdots + x_n}.$$

In particular, the ML estimate that was obtained at the previous part, coincides with θ_1.

For the second part, observe that,

$$\text{var}(X) = \mathbb{E}^{\theta}[X^2] - \Big(\mathbb{E}^{\theta}[X]\Big)^2 = \frac{1}{\theta^2} \implies \mathbb{E}^{\theta}[X^2] = \frac{2}{\theta^2}.$$

Now, using the exact same approach, we get,

$$\hat{\mathbb{E}}[X^2] = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \quad \text{and} \quad \mathbb{E}^{\theta}[X^2] = \frac{2}{\theta^2}.$$

Therefore,

$$\hat{\theta}_2 = \sqrt{\frac{2n}{x_1^2 + \cdots + x_n^2}}.$$

As a further observation, note that due to Cauchy-Schwarz inequality, we have, $(x_1^2 + \cdots + x_n^2)n \geq (x_1 + \cdots + x_n)^2$, hence,

$$\hat{\theta}_2 = \sqrt{\frac{2n}{x_1^2 + \cdots + x_n^2}} \leq \frac{n\sqrt{2}}{x_1 + \cdots + x_n} = \hat{\theta}_1\sqrt{2}.$$

(c) We can explicitly compute for, $\mathbb{P}(X \leq a_\theta)$, by integrating the distribution of X.

$$\mathbb{P}(X \leq a_\theta) = \int_0^{a_\theta} \theta e^{-\theta x} dx = 1 - e^{-\theta a_\theta} = \frac{1}{2},$$

thus,

$$e^{-\theta a_\theta} = \frac{1}{2} \implies a_\theta = \frac{\ln 2}{\theta},$$

where, $\ln(\cdot)$ denotes the logarithm, with respect to base e. Now, with this,

$$\frac{\ln 2}{\hat{\theta}_m} = \hat{a} \implies \hat{\theta}_m = \frac{\ln 2}{\hat{a}}.$$

Problem 3.

(a) Note that,

$$\log f_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = \sum_{i=1}^{n}\left(-\frac{1}{2}\log 2\pi v - \frac{(x_i-\mu)^2}{2v}\right) = -\frac{n}{2}\log 2\pi v + \frac{1}{2v}\sum_{i=1}^{n}(x_i-\mu)^2.$$

In order to obtain the ML estimates for the parameters; we need to set the partial derivative of the log-likelihood, with respect to μ, and v, and set them equal to 0; and solve for the corresponding estimates.

We first compute $\hat{\mu}$, via taking the derivative of the log-likelihood above, with respect to μ, and setting it equal to 0.

$$\frac{\partial}{\partial \mu}\log f_{X_1,\ldots,X_n} = \frac{1}{v}\sum_{i=1}^{n}(x_i-\mu) = 0 \implies \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}x_i.$$

Next, taking the derivative with respect to v, we arrive at,

$$\frac{\partial}{\partial v}\log f_{X_1,\ldots,X_n} = -\frac{n}{2v} - \frac{1}{2v^2}\sum_{i=1}^{n}(x_i-\hat{\mu})^2 = 0 \implies \hat{v} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{\mu})^2.$$

(b) This part is a computational assignment, whose solution can be found under the relevant document in Stellar.

(c) Let, $N(\mu,\sigma^2)$ denote a normal random variable, with mean μ; and variance, σ^2. Since, X_i has a distribution $N(\mu_1,v_1)$, with probability λ; and $N(\mu_2,v_2)$, with probability $1-\lambda$, we have,

$$f^{\theta}_{X_i}(x) = \lambda\cdot\frac{1}{\sqrt{2\pi v_1}}e^{-(x-\mu_1)^2/2v_1} + (1-\lambda)\cdot\frac{1}{\sqrt{2\pi v_2}}e^{-(x-\mu_2)^2/2v_2}.$$

(d) In words, the effect is that, as $v_1 \to 0$; the first normal, involved in the mixture, approaches to a degenerate distribution (mathematically, a delta function; located at x_1). Let us now, formally analyze this. We begin by noting,

$$f^{\theta}_{X}(x) = \frac{1}{2}\frac{1}{\sqrt{2\pi v_1}}e^{-(x-x_1)^2/2v_1} + \frac{1}{2}\frac{1}{\sqrt{2\pi v_2}}e^{-(x-\mu_2)^2/2v_2}.$$

In particular,

$$f^{\theta}_{X}(x_1) = \frac{1}{2}\frac{1}{\sqrt{2\pi v_1}} + \frac{1}{2}\frac{1}{\sqrt{2\pi v_2}}e^{-(x-\mu_2)^2/2v_2}.$$

We now verify the limiting behaviour. As v_1 goes to zero, the first term goes to ∞. Thus, the maximum value of the likelihood is infinity, and is attained by considering an arbitrarily narrow distribution on top of a single data point. It is intuitively clear that such a solution is not "sound"

(e,f) These parts are computational, and their solutions can be found under the relevant document in Stellar.

(g) With $v = \sigma^2$, we note that, $\mathbb{E}[Y^2] = \text{var}(Y) + (\mathbb{E}[Y])^2 = \sigma^2 + \mu^2$. Suppose now that, $Y = \mu + \sigma Z$, where, Z is a standard normal random variable. Now, using the identity, $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$, we can compute,

$$\begin{aligned}\mathbb{E}[Y^3] &= \mathbb{E}[(\mu + \sigma Z)^3] \\ &= \mathbb{E}[\mu^3 + 3\mu^2\sigma Z + 3\mu\sigma^2 Z^2 + \sigma^3 Z^3] \\ &= \mu^3 + 3\mu\sigma^2 \\ &= \mu^3 + 3\mu v,\end{aligned}$$

where, the second-to-last equality follows from the fact that, $\mathbb{E}[Z] = \mathbb{E}[Z^3] = 0$, for a standard normal random variable Z; together with the linearity of expectation.

Now, we are given that, $Y_1 \sim N(\mu_1, v_1)$, and, $Y_2 \sim N(\mu_2, v_2)$; and that, X is distributed with respect to Y_1, with probability λ; or with respect to Y_2, with probability $1 - \lambda$. Let, I be an indicator random variable, where, $I = 1$, if and only if, $X \sim Y_1$. Clearly, $\mathbb{P}(I = 1) = \lambda$. We can obtain $\mathbb{E}[X^5]$, using conditioning on I, as follows:

$$\begin{aligned}\mathbb{E}[X^5] &= \mathbb{E}[X^5|I = 1]\mathbb{P}(I = 1) + \mathbb{E}[X^5|I = 0]\mathbb{P}(I = 0) \\ &= \mathbb{E}[Y_1^5]\lambda + \mathbb{E}[Y_2^5](1 - \lambda) \\ &= \lambda(\mu_1^5 + 10\mu_1^3 v_1 + 15\mu_1 v_1^2) + (1 - \lambda)(\mu_2^5 + 10\mu_2^3 v_2 + 15\mu_2 v_2^2),\end{aligned}$$

where above, the first equality makes use of the law of total expectation, the second line follows from the fact that, conditioned on $I = 1$, X is distributed, according to Y_1 (and similarly, conditioned on $I = 0$, the distribution is according to that of Y_2); and the last line is obtained, by plugging in the formulas for the fifth moments.

(h) This part is a computational assignment, whose solution can be found under the relevant document in Stellar.

(i) Not necessarily. It might be true that, we are looking at a single crab family, which might differ by another factor (e.g., age, gender etc.).

Problem_2d_sol

February 7, 2018

Problem 2(d) The data file with 25 samples is provided as the file data_2d.csv. Your task is to produce numerical estimates of: i. θ_{ML} (part a) ii. $\hat{\theta}_1$ (part b) iii. $\hat{\theta}_2$ (part b) iv. $\hat{\theta}_m$ (part c)

Use the space below to write code to read the data file, and produce all four estimates noted above. Your code should print the four values (and identify them appropriately).

In [2]:
```
# import libraries
import numpy as np
import pandas as pd

# read the data
df = pd.read_csv("data_2d.csv", header=None)
observed_data = df[0].values

n = len(observed_data)
sum_data = sum(observed_data)
sum_data_squared = sum([x**2 for x in observed_data])
med_data = np.median(observed_data)

theta_ml = n / sum_data
theta_1 = theta_ml
theta_2 = np.sqrt(2 * n / sum_data_squared)
theta_m = np.log(2) / med_data

print("The maximum likelihood estimate is {}.".format(theta_ml))
print("The first moment matching estimate is {}.".format(theta_1))
print("The second moment matching estimate is {}.".format(theta_2))
print("The median matching estimate is {}.".format(theta_m))
```

```
The maximum likelihood estimate is 2.623542901104985.
The first moment matching estimate is 2.623542901104985.
The second moment matching estimate is 2.8650511519588524.
The median matching estimate is 2.4866175374041273.
```

Problem_3

February 8, 2018

Problem 3 The data for this problem is provided as a csv file called crab_data.csv. You need to provide answers to parts (b), (e) and (f). Refer to the Problem Set 1 pdf.

```
In [9]: # import libraries
        import numpy as np
        import pandas as pd
        from scipy.stats import norm
        import matplotlib.pyplot as plt
        from sklearn import mixture

        # read data as pandas dataframe
        df = pd.read_csv("crab_data.csv")
```

3(b) i. Provide the ML estimates of the parameter vector: $\hat{\theta} = (\hat{\mu}, \hat{v})$.

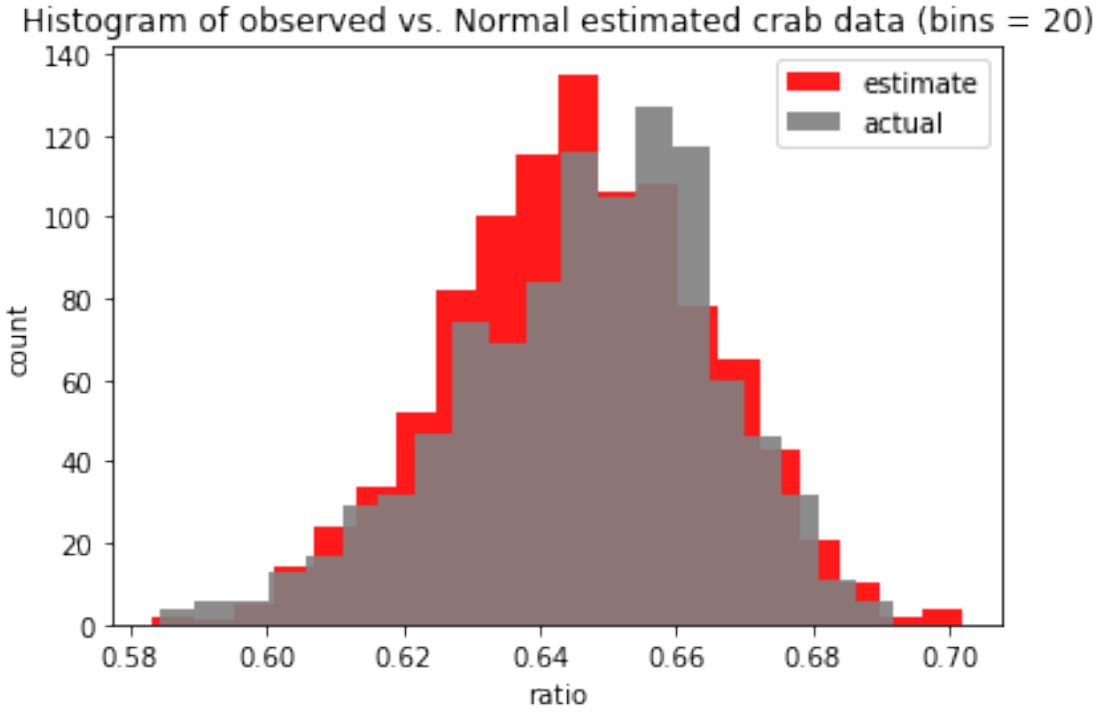
ii. Next, generate a sample of the same length as the number of observations in the crab dataset. This sample should should come from a Normal distribution with the same mean and variance you just estimated, i.e. $(\hat{\mu}, \hat{v})$. You may find the numpy library's random.normal() function helpful. The documentation can be accessed at: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html

iii. Finally, you need to produce a histogram of the random observations you generated using the ML estimates. Set the bins = 20, and use color = "red". On the same plot, also produce the histogram of the original data (empirical pdf) using bins = 20. Set the color of that histogram to be "gray".

```
In [10]: data = df.ratio.values
         n = len(data)

         mu_est = sum(data) / n
         var_est = sum([(x - mu_est)**2 for x in data]) / n
         sd_est = np.sqrt(var_est)

         print("i.")
         print("The ML estimate for the mean is: {}.".format(mu_est))
         print("The ML estimate for the variance is: {}.".format(var_est))
         print("\n")

         print("ii.")
```

```
samples_est = np.random.normal(mu_est, sd_est, (n,1))
print("Generated samples from the estimated normal!")
print("\n")

print("iii.")
plt.hist(samples_est, bins = 20, color='red', alpha = 0.9, label = 'estimate');
plt.hist(data, bins = 20, color='gray', alpha = 0.9, label = 'actual');
plt.xlabel('ratio');
plt.ylabel('count');
plt.title('Histogram of observed vs. Normal estimated crab data (bins = 20)')
plt.legend(loc = 'upper right');

def get_cdf(samples, start, end, n):
    samples = sorted(samples)
    sep = (end - start) / n
    N = len(samples)

    it = 0
    cdf = []

    for i in range(0, n + 1):
        x = start + sep * i
        while(it < N and samples[it] <= x):
            it += 1
        cdf.append(it / N)
    return cdf

def goodness_of_fit(f1, f2):
    return max([abs(x-y) for x,y in zip(f1,f2)])

N = 1000 # number of points to evaluate
actual_cdf = get_cdf(data,0,1,N)
est_cdf = [norm.cdf(i / N, mu_est, sd_est) for i in range(N + 1)]
gof = goodness_of_fit(actual_cdf, est_cdf)
print("The goodness of fit is {}.".format(gof)) # Precision depends on N
```

```
i.
The ML estimate for the mean is: 0.6467528146897326.
The ML estimate for the variance is: 0.0003626483163559565.


ii.
Generated samples from the estimated normal!


iii.
The goodness of fit is 0.06077041298623187.
```



3b (continued) Compare visually the two histograms generated above.

The crab data seems to resemble the shape of a Normal distribution at first, specifically the bell structure. However upon close inspection we see that there is a large domain where the estimated normal greatly varies from the observed distribution which suggests this isn't a satisfactory model. In particular there is a difference in the rates of decay to the left and right of the mean in the true distribution which is not the case for a Normal model.

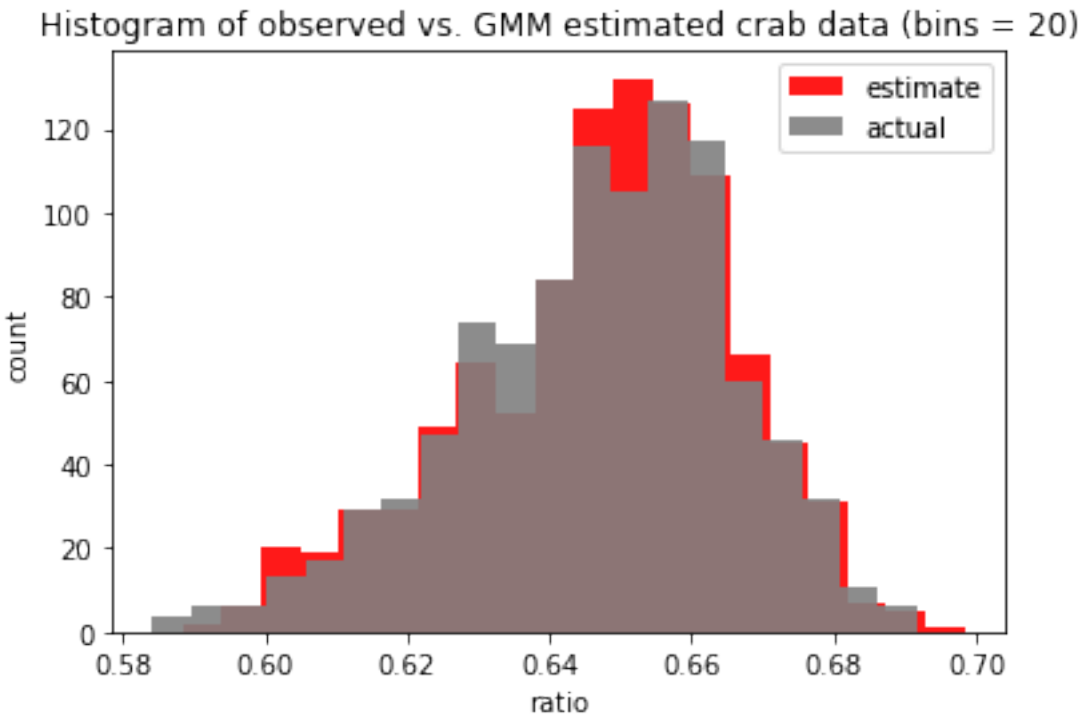
3(e) i. Generate a sample of the same length as the number of observations in the crab dataset. This sample should should come from a mixture of two Normal distributions with the EM algorithm's means and variance provided in the Pset pdf, i.e. $(\hat{\mu_1}, \hat{v_1}, \hat{\mu_2}, \hat{v_2})$ and the proportions of samples from each Normal distribution is dictated by $\hat{\lambda}$. You may find the numpy library's random.normal() function helpful. The documentation can be accessed at: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html

ii. Now, you need to produce a histogram of the random observations you generated using the EM estimates noted above. Set the bins = 20, and use color = "red". On the same plot, also produce the histogram of the original data (empirical pdf) using bins = 20. Set the color of that histogram to be "gray".

```
In [11]: print("i.")
         crab_gmm = mixture.GaussianMixture(n_components = 2,
                                            covariance_type = 'spherical')
         crab_gmm.fit(data.reshape(n,1))
         w1, w2 = crab_gmm.weights_
```

```
        [mu1], [mu2] = crab_gmm.means_
        v1, v2 = crab_gmm.covariances_
        sd1, sd2 = np.sqrt(v1), np.sqrt(v2)

        print("The estimated weights are: {}, {}.".format(w1,w2))
        print("The estimated means are: {}, {}.".format(mu1, mu2))
        print("The estimated variances  are: {}, {}.".format(v1, v2))

        gmm_samples = crab_gmm.sample(n)[0]

        print("Generated samples from the estimated mixture distribution!")
        print("\n")

        print("ii.")
        plt.hist(gmm_samples, bins = 20, color='red', alpha = 0.9, label = 'estimate');
        plt.hist(data, bins = 20, color='gray', alpha = 0.9, label = 'actual');
        plt.xlabel('ratio');
        plt.ylabel('count');
        plt.title('Histogram of observed vs. GMM estimated crab data (bins = 20)')
        plt.legend(loc = 'upper right');

        def mix_cdf(x):
            return w1 * norm.cdf(x, mu1, sd1) + w2 * norm.cdf(x, mu2, sd2)

        est_gmm_cdf = est_cdf = [mix_cdf(i / N) for i in range(N + 1)]
        gmm_gof = goodness_of_fit(actual_cdf, est_gmm_cdf)
        print("The goodness of fit is {}.".format(gmm_gof))
```

```
i.
The estimated weights are: 0.35146428211323905, 0.648535717886761.
The estimated means are: 0.6289415986473527, 0.6564053381934423.
The estimated variances  are: 0.00025615995258723614, 0.00015680564420825116.
Generated samples from the estimated mixture distribution!


ii.
The goodness of fit is 0.018515814632845018.
```



3e (continued) Compare visually the two histograms generated above.

Compared to the previous Normal model, the estimate obtained from the Gaussian mixture model (GMM) seems to better fit the data. There are less regions where the true distribution differs greatly from the estimated distribution. We still have the bell structure but now the estimated distribution better fits the tails of the true distribution.

3(f) Compute the statistics $\hat{m}^k, 1 \leq k \leq 5$.

```
In [12]: def emp_moment(data, k):
             return sum([x ** k for x in data]) / len(data)

         for i in range(1,6):
             print("The empirical {}th moment for the crab data is {}.".format(i, emp_moment(d
```

```
The empirical 1th moment for the crab data is 0.6467528146897326.
The empirical 2th moment for the crab data is 0.4186518516254457.
The empirical 3th moment for the crab data is 0.27122990835932553.
The empirical 4th moment for the crab data is 0.1758675029276196.
The empirical 5th moment for the crab data is 0.11412761692520862.
```