DS-GA 1002 - Homework 3

Eric Niblock

September 21th, 2020

1. (Triangular pdf) We are interested in fitting a model with a parametric pdf equal to

$$f_w(x) = \begin{cases} \frac{2x}{w^2} & 0 \leq x \leq w \\ 0 & otherwise \end{cases} \tag{1}$$

where the parameter w is nonnegative.

(a) The observed values are 1.25, 0.4, 1.5, 1, 1.2. What are the possible values of the parameter w?

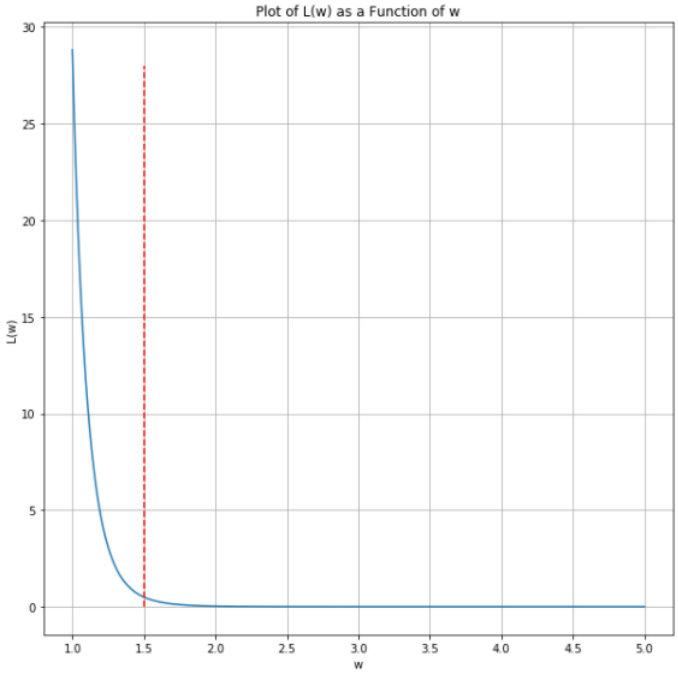
Since every value of x must be less than or equal to w if it is to produce some value concerning the pdf, we want to make sure that $w \geq x_i$ for each x_i. Since 1.5 is that largest value given for x, we should ensure that w lies somewhere in the domain $1.5 \leq w$

(b) Compute the likelihood function corresponding to these data and sketch it.

We would like to generate a likelihood estimation for parameter w, so, assuming independent samples,

$$\mathcal{L}_{\{x_1,\ldots,x_n\}}(w) = \prod_{i=1}^{n} \frac{2x_i}{w^2} = \left(\frac{2}{w^2}\right)^n \prod_{i=1}^{n} x_i = \frac{28.8}{w^{10}} \tag{2}$$

Observe the plot below which shows the likelihood estimate as a function of w. The red dotted line shows the constraint from part (a).



(c) What is the maximum likelihood estimate of w?

Taking the derivative of this function and setting it equal to zero will not yield the maximum of this function. Instead, we find the local maximum by observing that $\mathcal{L}_{\{x_1,...,x_n\}}(w)$ is greatest when w is smallest. Given the constraint from (a), we know that $w \geq max(\{x_1, ...x_n\})$, so in this case,

$$w_{ML} = max(\{x_1, ..., x_5\}) = 1.5 \tag{3}$$

(d) If we observe 100 independent samples that are generated according to the parametric model with a fixed value of w, do you think that there is any chance that the ML estimate of w is correct? Justify your answer intuitively.

The probability of our estimate of w being correct is approximately zero. This is because w is essentially a random variable, hence, $P(w_{ML} = w) = 0$. Furthermore, we can imagine that even our estimation of w would change if we had one observation that was greater than 1.5.

(e) Let us assume $w = 2$. Generate a sample from a random variable following the model using a uniform sample from the interval [0, 1] equal to 0.64.

First, we find the cdf from the given pdf, noting that $w = 2$,

$$F_w(x) = \int_0^x \frac{2x`}{4} dx` = \frac{x^2}{4} \tag{4}$$

Then, we can generate the inverse of this function,

$$F_w^{-1}(u) = 2\sqrt{u} \tag{5}$$

So, if we want to generate a sample corresponding to 0.64, we have,

$$F_w^{-1}(0.64) = 2\sqrt{0.64} = 1.6 \tag{6}$$

2. (Generating a uniform distribution) Assume that we can generate samples from a continuous random variable $\tilde{a}$ with an invertible cdf $F_{\tilde{a}}$.

(a) What is the distribution of the random variable $\tilde{b} = F_{\tilde{a}}(\tilde{a})$?

Let's attempt to find the cdf of $\tilde{b}$. So, we have,

$$\begin{aligned} F_{\tilde{b}}(b) &= P(\tilde{b} \leq b) = P(F_{\tilde{a}}(\tilde{a}) \leq b) = P(\tilde{a} \leq F_{\tilde{a}}^{-1}(b)) \\ &= F_{\tilde{a}}(F_{\tilde{a}}^{-1}(b)) = b \end{aligned} \tag{7}$$

The derivative of the above result would yield 1, corresponding to the pdf. So the distribution of $\tilde{b}$ is therefore uniform on the interval [0,1].

(b) Using the answer to the previous question and a result from the lecture notes, propose a method to generate an exponential random variable with parameter λ_2 from the samples of another exponential random variable with parameter λ_1. The algorithm should simplify to a very simple procedure.

Let X and Y be two random variables such that $f_X(x) = \lambda_1 e^{-\lambda_1 x}$ if $x \geq 0$ and $f_Y(y) = \lambda_2 e^{-\lambda_2 y}$ if $y \geq 0$. If we call U the uniform distribution between 0 and 1, we then have, $U = F_Y(Y)$, and from lecture notes, we have, $F_Y(Y) = 1 - e^{-\lambda_1 Y}$. Furthermore, we also know from the notes that,

$$F_X^{-1}(U) = \frac{1}{\lambda_2} log\left(\frac{1}{1-U}\right) \tag{8}$$

We then replace U with $F_Y(Y)$, yielding,

$$\begin{aligned} F_X^{-1}(U) &= \frac{1}{\lambda_2} log\left(\frac{1}{1-(1-e^{-\lambda_1 Y})}\right) \\ &= -\frac{1}{\lambda_2} log(e^{-\lambda_1 Y}) \\ &= \frac{\lambda_1}{\lambda_2} Y \end{aligned} \tag{9}$$

This simple procedure can be used to generate an exponential random variable with parameter λ_2 from the samples of another exponential random variable with parameter λ_1.

3. (Halloween) In Halloween Laura and her brother Mike arrive at a house where they offer them a bowl with 2 chocolate bars. Mike grabs a random number of chocolate bars; he grabs 0, 1, or 2 with the same probability. Laura then grabs some chocolate bars out of the remaining ones; also with uniform probability (there is the same probability that she grabs 0, 1, etc.).

(a) Model the number of bars grabbed by Mike and the number of bars grabbed by Laura as random variables and compute their joint pmf.

The number of candy bars grabbed by Mike will be associated with random variable $\tilde{m}$ and the number grabbed by Laura will be $\tilde{l}$. Now, if we just examine Mike, we have,

$$p_{\tilde{m}}(0) = p_{\tilde{m}}(1) = p_{\tilde{m}}(2) = \frac{1}{3} \tag{10}$$

Then, we know by the chain rule for discrete random variables that,

$$p_{\tilde{m},\tilde{l}}(m, l) = p_{\tilde{m}}(m) p_{\tilde{l}|\tilde{m}}(l|m) \tag{11}$$

It is trivial to calculate the conditional probabilities associated with Laura, after Mike has chosen, because, again, they are uniform,

$$p_{\tilde{l}|\tilde{m}}(0|0) = p_{\tilde{l}|\tilde{m}}(1|0) = p_{\tilde{l}|\tilde{m}}(2|0) = \frac{1}{3} \tag{12}$$

$$p_{\tilde{l}|\tilde{m}}(0|1) = p_{\tilde{l}|\tilde{m}}(1|1) = \frac{1}{2} \tag{13}$$

$$p_{\tilde{l}|\tilde{m}}(0|2) = 1 \tag{14}$$

So, the joint pmf is given by

$$\begin{array}{lll} p_{\tilde{m},\tilde{l}}(0,0) = 1/9 & p_{\tilde{m},\tilde{l}}(0,1) = 1/9 & p_{\tilde{m},\tilde{l}}(0,2) = 1/9 \\ p_{\tilde{m},\tilde{l}}(1,0) = 1/6 & p_{\tilde{m},\tilde{l}}(1,1) = 1/6 & p_{\tilde{m},\tilde{l}}(2,0) = 1/3 \end{array} \tag{15}$$

With every other $p_{\tilde{m},\tilde{l}}(m, l) = 0$.

(b) Compute the marginal pmf of the number of bars grabbed by Laura.

We compute the marginal pmf by summing over the various values of $\tilde{m}$ as follows

$$p_{\tilde{l}}(0) = p_{\tilde{m},\tilde{l}}(0,0) + p_{\tilde{m},\tilde{l}}(1,0) + p_{\tilde{m},\tilde{l}}(2,0) = \frac{11}{18} \tag{16}$$

$$p_{\tilde{l}}(1) = p_{\tilde{m},\tilde{l}}(0,1) + p_{\tilde{m},\tilde{l}}(1,1) = \frac{5}{18} \tag{17}$$

$$p_{\tilde{l}}(2) = p_{\tilde{m},\tilde{l}}(0,2) = \frac{1}{9} \tag{18}$$

This fully describes the marginal pmf, with all other outcomes having a probability of zero.

(c) What is the conditional pmf of the number of bars grabbed by Mike if we know that Laura grabbed 1 bar?

We find,

$$p_{\tilde{m}|\tilde{l}}(0|1) = \frac{p_{\tilde{m},\tilde{l}}(0,1)}{p_{\tilde{l}}(1)} = \frac{2}{5} \tag{19}$$

$$p_{\tilde{m}|\tilde{l}}(1|1) = \frac{p_{\tilde{m},\tilde{l}}(1,1)}{p_{\tilde{l}}(1)} = \frac{3}{5} \tag{20}$$

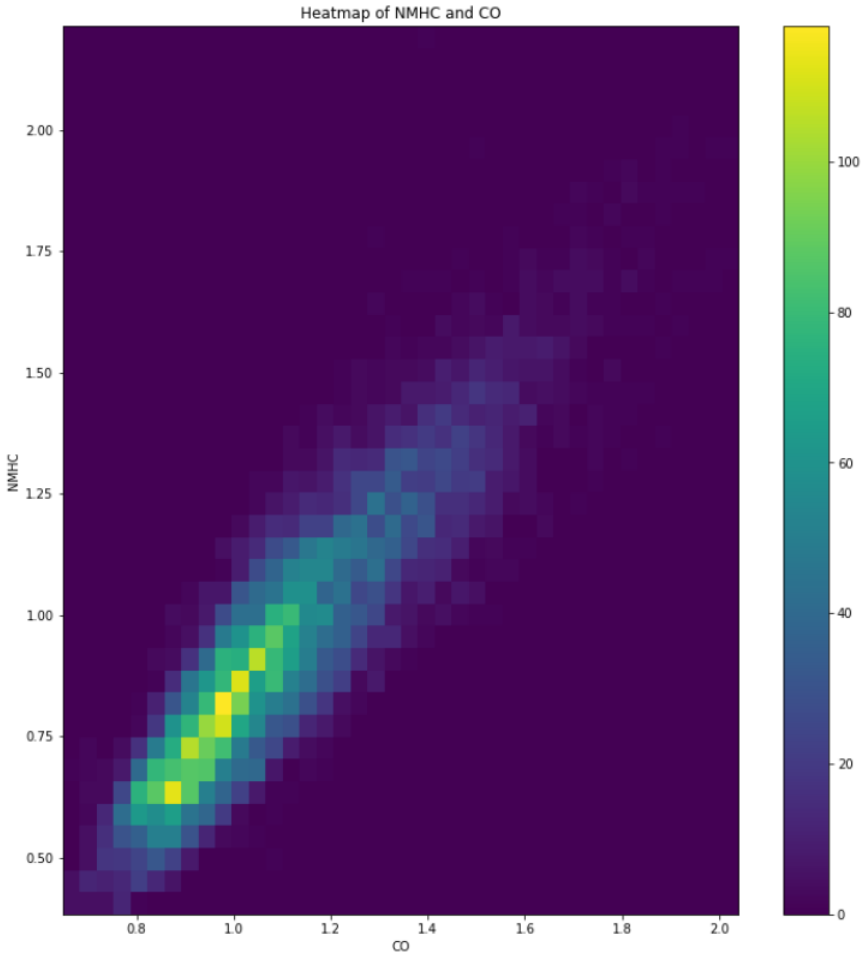
This fully describes the conditional pmf, with all other outcomes having a probability of zero.

4. **(Air Quality)** ***air_quality.csv*** **contains hourly sensor readings of concentrations of various chemicals found in the air outside an Italian city (see the UCI repository for more details). The units are $\frac{g}{m^3}$. We are interested in a non-parametric estimation of the 2D probability distribution of carbon monoxide (CO) and Non Metallic HydroCarbons (NMHC) using multivariate Gaussian kernels.**

   (a) **Plot the heatmap of the dataset. [Recommended: matplotlib or seaborn's heatmap functions will be useful here.]**

   The heatmap was constructed using the code below, with a grid of 40×40 bins. The colorbar on the side relates the color of each bin with the number of data points that fall into each bin.

```
import pandas as pd
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

f = pd.read_csv( r'C:\Users\Eric\Downloads\air_quality.csv')

x = np.array(f['CO'])
y = np.array(f['NMHC'])

heatmap, xedges, yedges = np.histogram2d(x, y, bins=40)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

plt.clf()
plt.figure(figsize = (13,13))
plt.title('Heatmap of NMHC and CO')
plt.xlabel('CO')
plt.ylabel('NMHC')
a = plt.imshow(heatmap.T, extent=extent, origin='lower')
cbar = plt.colorbar(a)
plt.show()
```

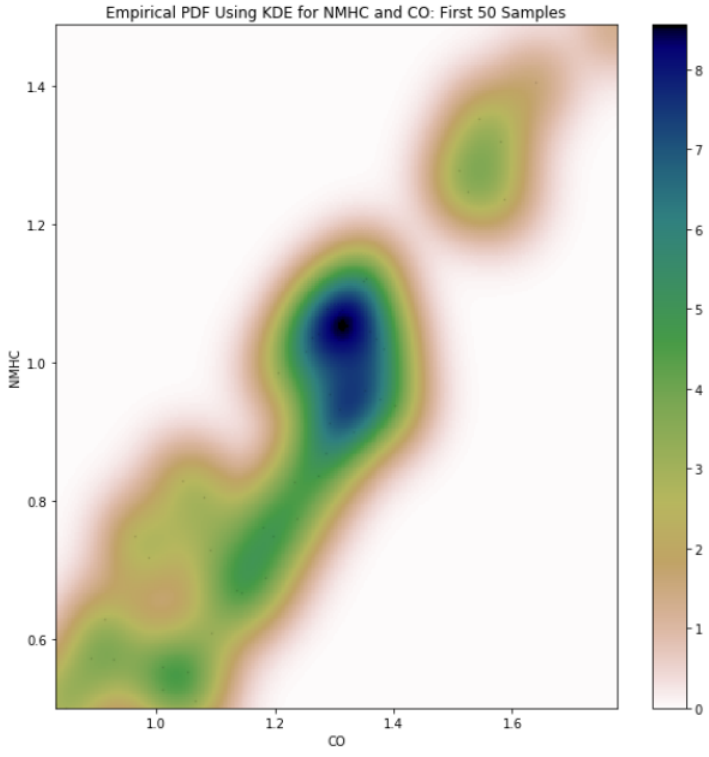


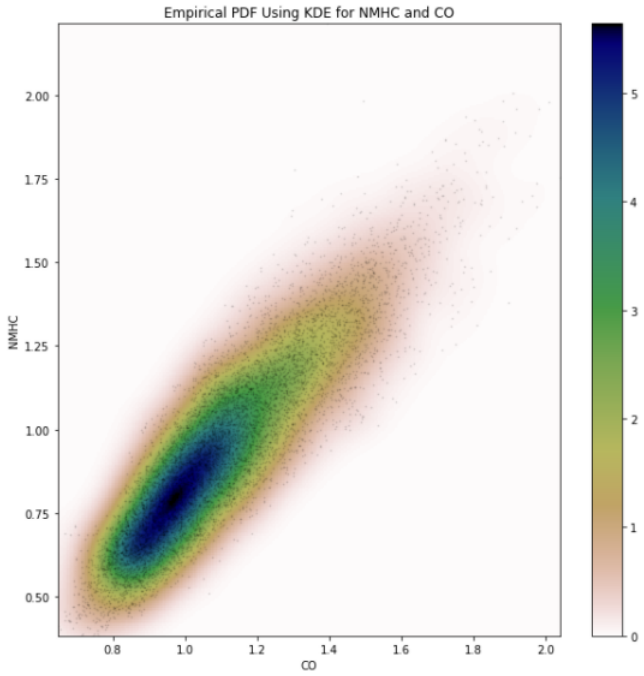
(b) Plot the estimated probability density function applying a 2D Gaussian Kernel i) using the first 50 data points and ii) using the whole dataset. You can try multiple bandwidths, but be sure to include one plot using bandwidth h=0.05. [Note: You may use seaborn's kdeplot or sklearn's KernelDensity].

Below is the code used to generate the empirical pdfs of the first 50 data points, as well as all of the data points (first and second plot, respectively). Both plots use a bandwidth of $h = 0.05$. The colorbar on the side represents the likelihood of being in a certain region of the plot.

```
from sklearn.neighbors import KernelDensity

def kde2D(x, y, bw):

    xbins=300j
    ybins=300j

    xx, yy = np.mgrid[x.min():x.max():xbins,
                      y.min():y.max():ybins]

    xy_sample = np.vstack([yy.ravel(), xx.ravel()]).T
    xy_train  = np.vstack([y, x]).T

    kde = KernelDensity(bandwidth=bw)
    kde.fit(xy_train)

    z = np.exp(kde.score_samples(xy_sample))
    return xx, yy, np.reshape(z, xx.shape), kde
```

```
import numpy as np
import matplotlib.pyplot as plt

x = f['CO']
y = f['NMHC']

xx, yy, zz, kde = kde2D(x, y, .05)

plt.figure(figsize=(10,10))
plt.title('Empirical PDF Using KDE for NMHC and CO')
plt.pcolormesh(xx, yy, zz, cmap=plt.cm.gist_earth_r)
plt.colorbar()
plt.scatter(x, y, c='k',s=0.8, alpha=0.1)
plt.xlim(x.min(),x.max())
plt.ylim(y.min(),y.max())
```






```
area = (xx[1][0] - xx[0][0])*(yy[0][1] - yy[0][0])
print('Approximate Sum of Volume Under Distribution: ' + str(sum(sum(zz))*area))
```

```
Approximate Sum of Volume Under Distribution: 0.9968614604745094
```

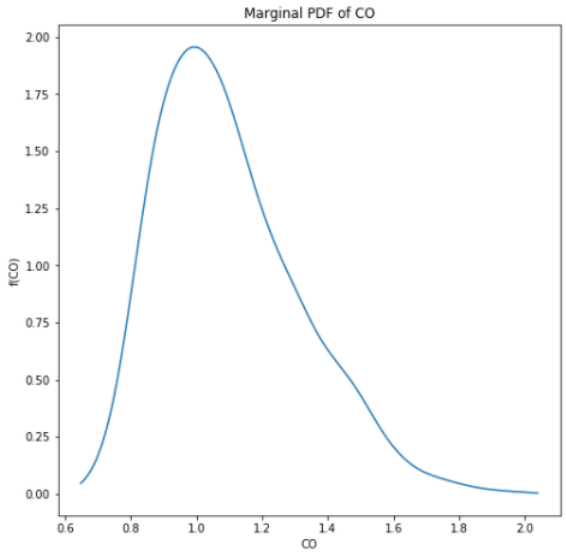
The strip of code above was used as a "sanity check". We calculated the area of each small box used to plot the empirical pdf, and multiplied by the "height" or likelihood at each corresponding point. As expected, this volume calculation yielded approximately one.

(c) Using the pdf estimated via KDE using the Gaussian kernel in part (b), plot the marginal pdf of CO. Describe your approach.

The following code was used to produce the marginal pdf of CO. In order to do so, we summed all of the values of the likelihood (or the height) of the pdf at each value of CO. This amounts to summing the values of the likelihood along the line at a value of CO, for each value of CO. We then normalized the values.

```
temp = np.array([sum(e) for e in zz])
new_x = [xx[i][0] for i in range(len(xx))]
mar =  temp/(sum(temp)*(new_x[1]-new_x[0]))
```

```
plt.figure(figsize=(8,8))
plt.plot(new_x,mar)
plt.title('Marginal PDF of CO')
plt.xlabel('CO')
plt.ylabel('f(CO)')
```

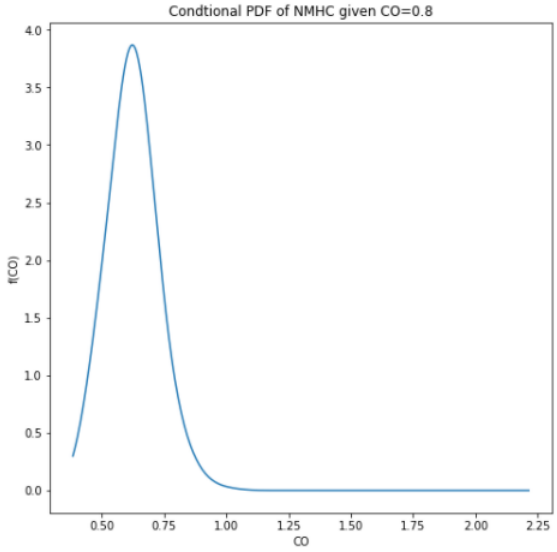


(d) Using the pdf estimated via KDE using the Gaussian kernel in part (b), plot the conditional pdf of NMHC given CO = 0.8. Describe your approach.

The following code was used to produce the condition pdf of NMHC when CO = 0.8. This was achieved by finding all of the likelihood values along the line CO = 0.8 and normalizing by the total area enclosed under the curve (its best to imagine the plot as 3D, with the color corresponding to the height at different regions).

```
xx = np.empty(300)
xx.fill(0.8)
yy = yy[0]
xy_sample = np.vstack([yy.ravel(), xx.ravel()]).T
zz = np.exp(kde_skl.score_samples(xy_sample))
con =  zz/(sum(zz)*(yy[1]-yy[0]))
```

```
plt.figure(figsize=(8,8))
plt.plot(yy,con)
plt.title('Condtional PDF of NMHC given CO=0.8')
plt.xlabel('CO')
plt.ylabel('f(CO)')
```


Condtional PDF of NMHC given CO=0.8
f(CO)
CO
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25