Statistical Pattern Recognition

Homework 3

Shervin Halat

98131018

1.

1.

a.

$\hat{\theta} : \arg\max (P(D|\theta))$ samples are independent $(x_1, \dots x_n)$

$$\rightsquigarrow \hat{\theta} = \arg\max_{\theta} \left[\prod P(x_k|\theta) \right] = \arg\max_{\theta} \left[\ln\left(\prod P(x_k|\theta) \right) \right]$$

$$= \arg\max_{\theta} \left[\sum_{k=1}^{n} \ln P(x_k|\theta) \right]$$

$$\Longrightarrow \nabla_{\theta} = 0 \rightsquigarrow \nabla_{\theta} f(x) = 0 \Longrightarrow \hat{\theta} : \sum_{1}^{n} \frac{\partial \left(\frac{x^3 e^{-x/\theta}}{6\theta^4} \right)}{\partial \theta}$$

$$= \sum_{1}^{n} \frac{x^3 e^{-x/\theta} \times 6\theta^4 \times (x\theta^{-2}) - 24\theta^3 (x^3 e^{-x/\theta})}{(6\theta^4)^2} = 0$$

$$\Longrightarrow \sum_{1}^{n} (6x^4\theta^2 e^{-x/\theta} - 24\theta^3 x^3 e^{-x/\theta} = 0$$

assuming $\theta, x \neq 0 \rightsquigarrow \sum_{1}^{n} (6x - 24\theta) = \sum_{1}^{n} x - 4\theta = 0$

$$\Longrightarrow n \times 4\theta = \sum_{1}^{n} x \rightsquigarrow \boxed{\theta = \frac{\sum_{1}^{n} x}{4n}}$$

b. $\frac{(6.8) + 7.2 + 4.7 + 7.9 + 9.5 + 6.1}{6} \times \frac{1}{4} = \boxed{7{,}0\bar{3}}$

c.

R: number of Reds observed in $\underline{N}$ times

$N = 5 \quad R = 4 \qquad f_5(R=4|\theta)$: independent

Blue balls $= \{0, 1, 2, 3, 4\}$

Red Ball: 1

$$\begin{cases} 1 : \text{Red} \\ \theta : \text{Blue} \end{cases} \longrightarrow \boxed{f_5(R=4|\theta) = \left(\frac{1}{1+\theta}\right)^4 \times \left(\frac{\theta}{1+\theta}\right) \times 5}$$

d.

MLE : $\quad \hat{\theta} : \arg\max_{\theta} \left(P(D|\theta) \right) = \arg\max_{\theta} \left(f_5(R=4|\theta) \right)$

$\rightarrow \hat{\theta} = \arg\max_{\theta} \left(\sum_{1}^{n} \ln P(x_k|\theta) \right)$

$\rightarrow \left(\left(\frac{1}{1+\theta}\right)^4 \times \frac{\theta}{1+\theta} \right)' = 0 \rightarrow \ldots \rightarrow (\theta+1)^{-5} - 5\theta(\theta+1)^{-6} = 0$

$\rightarrow \boxed{\theta = 1/4} \rightarrow$ closest Discrete values to $1/4$ are $\underline{0}$ and $\underline{1}$ but $\underline{0}$ logically is not acceptable

hence, $\boxed{\theta = 1}$ ✓

e.

$\mathcal{E}(\Theta): 0.2$ (Posterior distribution for each θ in Θ)

$$P(\theta|D): P(\theta) \times P(D|\theta) = \left(\frac{1}{1+\theta}\right)^4 \left(\frac{\theta}{1+\theta}\right) \times 5 \times \underbrace{P(\theta)}_{0.2}$$

$$= \boxed{\left(\frac{1}{1+\theta}\right)^4 \times \left(\frac{\theta}{1+\theta}\right) = P(\theta|D)}$$

for each θ_i : $P(\theta_i): 0.2$

$\rightarrow$ for each θ_i : $P(\theta_i|D) = \left(\frac{1}{1+\theta}\right)^4 \left(\frac{\theta}{1+\theta}\right) \times \frac{1}{5} \times 5$

$$= \boxed{\left(\frac{1}{1+\theta}\right)^4 \times \left(\frac{\theta}{1+\theta}\right) = P(\theta_i|D)}$$

f.

$f(x|\lambda) = \lambda^2 x e^{-\lambda x}$ $\sim x > 0$

$\lambda > 0$ and $1 \leq i \leq n$

$T_1: 3$

$T_2: 5$

$T_3: 1$

$T_4: 4$

$T_5: 7$

$$f_5(x|\lambda) = (\lambda^2 \times 3 e^{-\lambda 3}) \times (\lambda^2 \times 5 \times e^{-\lambda 5})$$
$$\times (\lambda^2 \times 1 \times e^{-\lambda}) \times (\lambda^2 \times 4 \times e^{-\lambda \Sigma})$$
$$\times (\lambda^2 \times 7 \times e^{-7\lambda})$$
$$= \boxed{\lambda^{10} \times 420 \times e^{-20\lambda} = f_5(x|\lambda)}$$

g. with the help of taking 'Ln' from $\prod f(x|\lambda)$ and considering equation in part 'f' : $(\lambda^2 x e^{-\lambda x})$

$$\frac{\partial\left(\sum_{n=1}^{5} \ln(\lambda^2 x e^{-\lambda x})\right)}{\delta \lambda} = 0 \longrightarrow \frac{\delta\left(\sum_{1}^{5} (\ln \lambda^2 + \ln x - \lambda x)\right)}{\delta \lambda} = 0$$

$$\longrightarrow \left(5 \times \frac{2}{\lambda}\right) - (3+1+5+4+7) = 0 \longrightarrow \boxed{\lambda = 0.5}$$

2.

2.

a.

According to question's assumptions

$$f(k;\lambda) = \Pr(X=k) = \frac{\lambda^{k} e^{-\lambda}}{k!}$$

$$\mathrm{PDF}(Y_1 \dots Y_s \mid \lambda) = \frac{\prod_{k=1}^{S} (n_k x_k \lambda)^{y_k} \times e^{-\lambda \left(\sum_{k=1}^{S} n_k x_k \right)}}{\prod_{k=1}^{S} (y_k)!}$$

b.

Taking 'Ln' then derivation from equation above and set it equal to zero, the estimation of the parameter 'λ' will be obtained:

$$\sum_{k=1}^{s} y_k \frac{x_k n_k}{x_k n_k \lambda} - \sum_{k=1}^{s} n_k x_k = 0 \Longrightarrow$$

$$\sum_{k=1}^{S} y_k/\lambda = \sum_{k=1}^{S} x_k n_k = 0 \Longrightarrow \boxed{\hat{\lambda} = \frac{\sum_{k=1}^{S} y_k}{\sum_{k=1}^{S} n_k x_k}}$$

c.

In order to show that MLE is unbiased we have to show that $\underline{E[\hat{\lambda}] = \lambda}$

we have: $E\left[\frac{\sum_{k=1}^{S} y_k}{\sum_{k=1}^{S} n_k x_k} \right] = \frac{\sum_{k=1}^{S} E[y_k]}{\sum_{k=1}^{S} E[n_n x_k]}$

$$\underline{\underline{E[y_k] = \lambda_i}} \quad \frac{S \times \lambda_i}{S \times n_i x_i} = \frac{\lambda_i}{n_i x_i} \xrightarrow{\lambda_i = \lambda x_i n_i} \frac{\lambda_i}{x_i n_i} = \lambda \,//$$

Therefore the MLE is unbiased

d. $\text{Var}[\hat{\lambda}] = E[(\hat{\lambda} - \lambda)^2] = E[(\hat{\lambda})^2 + (\lambda)^2 - (2\lambda\hat{\lambda})]$

$$= \lambda^2 - 2\lambda \underbrace{E[\hat{\lambda}]}_{\lambda} + E[(\hat{\lambda})^2] = \lambda^2 - 2\lambda^2 + E[(\hat{\lambda})^2]$$

(from part 'c')

$$= E[(\hat{\lambda})^2] - \lambda^2 = \text{Var}[\hat{\lambda}]$$

e.

we know that: $MSE = \text{Var}[\hat{\lambda}] + \underbrace{(E[\hat{\theta}] - \theta)^2}_{Bias = 0}$

$$Bias = 0 \Rightarrow \boxed{MSE = \text{Var}[\hat{\theta}]}$$

f.

According to the equation obtained in Part 'b':

$$\frac{1+3+15}{0.3\times 10+0.2\times 25+0.5\times 100} = \frac{19}{58}$$

$$\Rightarrow \boxed{\lambda = 1/3}$$

g.

From the Bayes theorem we have:

$$P(\mathbf{\Theta}|data) = \frac{P(data|\mathbf{\Theta}) \times P(\mathbf{\Theta})}{P(data)}$$

We now want to calculate the left hand side of the equation above (for parameter λ) which is called the posterior probability.

P(data| λ) is something we've come across before in part 'a' and now we have the prior probability of λ.

As we haven't any information on P(data), we disregard this parameter which plays a normalization constant. Hence, the equation below will be used:

$$P(\mathbf{\Theta}|data) \propto P(data|\mathbf{\Theta}) \times P(\mathbf{\Theta})$$

g.

$$P(\Lambda \mid \text{data}) \overset{\text{Bayes}}{=\!=} P(\Lambda) \times P(\text{data} \mid \Lambda)$$

$$= \frac{\beta^{\alpha} \lambda^{\alpha-1} e^{-\beta\lambda}}{(\alpha-1)!} \times \frac{\prod_{k=1}^{S} (n_k x_k \lambda)^{y_k} \times e^{-\lambda\left(\sum_{k=1}^{S} n_k x_k\right)}}{\prod_{k=1}^{S} (y_k!)}$$

3.

a.

Considering logic of Central Limit Theorem (which states that given any kind of distribution with a mean **μ** and variance $\boldsymbol{\sigma^2}$, the sampling distribution of the mean approaches a normal distribution with a mean of **μ** and variance of $\boldsymbol{\sigma^2/N}$ as N, the sample size, increases), also considering that for a series of numbers from 1 to N, the mean is computed by (N+1)/2, the following parts were answered by equation below:

(N+1)/2 = mean (observed(X(i))

$$\boldsymbol{(N+1)/2 = \frac{\sum_{i=1}^{n} X(i)}{n}}$$

x(1) = 50 → (N+1)/2 = mean(x(1)) → (N+1)/2 = 50 → N = 99

a1.

x(2) = 30

(N+1)/2 = mean(x(1) , x(2)) → N = 79

a2.

x(3) = 60

(N+1)/2 = mean(x(1) , x(2) , x(3)) → N = 92

a3.

x(4) = 70

(N+1)/2 = mean(x(1) , x(2) , x(3), x(4)) → N = 104 a4.

x(5) = 5

(N+1)/2 = mean(x(1) , x(2) , x(3), x(4), x(5)) → N = 85

b.

Obviously, the distribution of X is uniform (since capturing each one of the tanks are equally likely) with parameter of N which determine the interval of [1,N] of the distribution.

Hence, **X ~ U (1,N)** (N is total number of tanks)

For uniform distribution we have:

f(x|N) = 1/N (for 0 ≤ x ≤ N)

considering (x(1) to x(n) be independent) and letting x(1)≤x(2)≤⋯≤x(n)x(1)≤x(2)≤⋯≤x(n) be the order statistics. Then it is easy to see that the likelihood function is as follows:

f(X|N) = $\prod_{i=1}^{n} f(x(i)\,|N)$ = $\prod_{i=1}^{n} 1/\mathrm{N}$ = N^{-n}

Hence the steps of calculating maximum likelihood estimator for N is as follows:

Step1:

We know that N should be at least greater than maximum of {x(1), x(2), …, x(n)}:

maximum{x(1), x(2), …, x(n)} ≤ N

Step2:

Also, the derivation of N^{-n} over N should become as close as possible to zero to maximize *f(X|N) or* N^{-n}.

The derivation is as follows:

$$\frac{d\ f(X|N)}{d\ N} = (-n) * N^{-n-1}$$

As we can see, the derivation is negative. Therefore, N should have the least possible value to get to the closer to zero value.

Considering both steps 1 and 2 leads us to:

Estimation: N' = maximum{x(1), x(2), …, x(n)}

c.

Logically, the obtained MLE for N is not acceptable and for sure it is a biased estimator (since, logically, Parameter N should be something greater than the maximum!). In the following the bias and variance of the estimator is calculated:

Bias:

To compute the bias, we will first compute the CDF of N' then we compute PMF of N' and then The bias (E[N'] – N).

The CDF would become: **P(Y≤x)**

$P(Y \leq x) = P(\max(X1,X2,\cdots,Xn) \leq x)$

$= P(X1 \leq x, X2 \leq x, \cdots, Xn \leq x) = \prod_{i=1}^{n} P(Xi \leq x)$

$= \prod_{i=1}^{n} \frac{x}{\mathrm{N}}$

$$\mathbf{= (\frac{x}{N})^n = CDF \rightarrow PDF = n_x(x^{n-1}/N^n)}$$

Therefore, the E[max(x1…xn)] would become:

E[max(x1…xn)]= $\frac{n(N+1)}{\mathbf{n+1}}$

Hence, the Bias becomes:

$$\textbf{\textit{Bias:}}\ \frac{n-N}{n+1}$$

So the MLE is biased!

Variance:

Variance = E[(N'-N)^2] = E[N'^{2}+N^2-2N'xN] =

= N^2 – 2xNx(n(N+1)/n+1) + E[N'^{2}]

Therefore, the variance becomes:

$$\textbf{\textit{Variance:}}\ N^2\frac{N}{(N+1)^2(N+2)}$$

d.

In order to get rid of the bias of the MLE, the estimation should be changed in a way that the calculated bias becomes equal to zero.

Therefore, it is enough to calculate the estimation by the equation below:

$$\textbf{Unbiased Estimation: N' =}\ \frac{n+1}{n} * \max\{X1, \ldots, Xn\} - 1$$

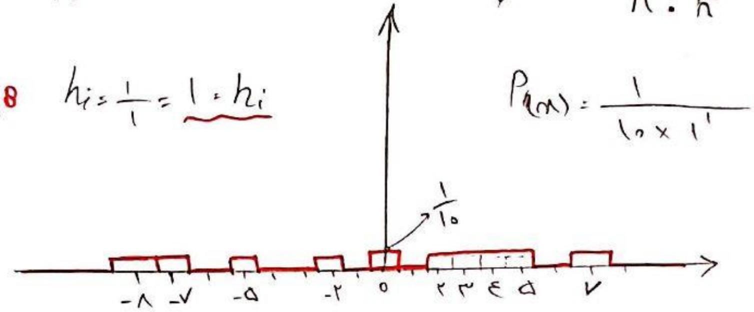
As it was mentioned before, the logic behind the unbiased estimation is that it is more probable that N is more than the maximum of observed samples.

4.

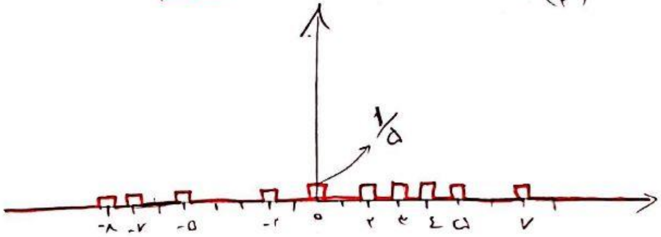
a.

$h_i = \frac{1}{\sqrt{i}}$ $i = 1, 4, 11$ $P_\phi(x) = \frac{1}{n \cdot h^d} \sum_{i=1}^{n} \phi\left(\frac{X - X_i}{h}\right)$

$\underline{i=1}$ 8 $h_i = \frac{1}{1} = \underline{1 = h_i}$ $P_{(x)} = \frac{1}{۱۰ \times 1^1} \times 1 = \boxed{\frac{1}{۱۰}}$



$\underline{i=4}$ 8 $h_i = \frac{1}{\sqrt{4}} = \underline{\frac{1}{2}} = h_i$ $P_{(x)} = \frac{1}{۱۰ \times (\frac{1}{۲})^1} \times 1 = \boxed{\frac{1}{۵}}$



$\underline{i=11}$ 8 $h_i = \frac{1}{\sqrt{11}} = \frac{1}{۳٫۳}$ $P_{(x)} = \frac{1}{۱۰ \times \frac{1}{۳٫۳}} = \boxed{\frac{1}{۳}}$

"نمودار i=11 نیز همانند i=4 است فقط $P_{(x)} = \frac{1}{۳}$ برابر هر نقطه نمونه "

b.

$h_i = \frac{h}{\sqrt{i}}$

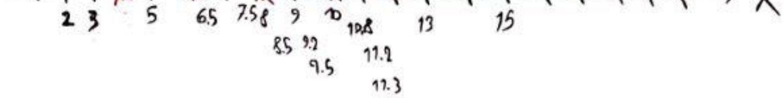
Considering 'i' being Constant, for very little values of 'h' The parzen windows would be Small enough that they have no Shared points and as a result, the overall shape of the estimated PDFs would be the Same with many Spikes and bumpy surfaces; for again Small enough values of h. As h is increased, the roughness of the curves decreases and become more smooth with Less spikes to a value of 'h' at which the whole PDF becomes a straight line with constant value and

C.

$i = 9, 16, 25$ / $k = \sqrt{i}$ / $P_x = \frac{k}{n \cdot V}$ / $n = 16$

$i = 9\,8$ $k = 3$

X	k-Neighbors	V	P(X)
4	2, 3, 5	4	3/64
6	5, 6.5, 7.5	3	1/16
8	7.5, 8, 8.5	1	3/16
10	9.2, 9.5, 10.8	1.6	3/25.6
12	11.2, 11.3, 13	2	3/32



i: 16 :   k: 4

X	k-neighbors	V	P(x)
4	10.8, 11.2, 11.3, 13	2.4	1/9.6
6	5, 6.5, 7.5, 8	4	1/16
8	7.5, 8, 8.5, 9	2	1/8
10	9.2, 9.5, 10, 10.8	1.6	1/6.4
12	10.8, 11.2, 11.3, 13	2.4	1/9.6

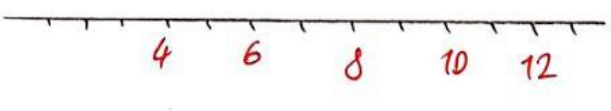
i: 25 :   k: 5

X	k-Neighbors	V	P(x)
4	2 3, 5, 6.5, 7.5	5	1/16
6	5, 6.5, 7.5, 8, 8.5	5	1/16
8	7.5, 8, 8.5, 9, 9.2	2.4	5/38.4
10	9, 9.2, 9.5, 10, 10.8	2	5/32
12	10, 10.8, 11.2, 11.3, 13	4	5/64

d.

sketch of estimated densities for i=9, 16, 25
with different colors:

k=3 : ——
k=5 : ——
K=4 : ——



e.

To solve this problem, the PDF of each sample should be computed and then the sum of all these PDFs over each point on X axis should be computed to estimate the density.

The results for points of $y = 4, 9, 14$ are calculated: (Parameter 'h' is estimated 4 since $\sqrt{16} = \sqrt{N} = h = 4$)

$$P_\phi(x) = \frac{1}{16} \times \frac{1}{4} \times \sum_{i=1}^{n} \phi\left(\frac{x - x_i}{4}\right)$$

$$\phi\left(\frac{x - x_i}{4}\right) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x - x_i}{4}\right)^2 / 2}$$

The results were calculated using Python. (Related Script is P4.e)

$y = 4 \longrightarrow P_\phi(4) = 0.0479$

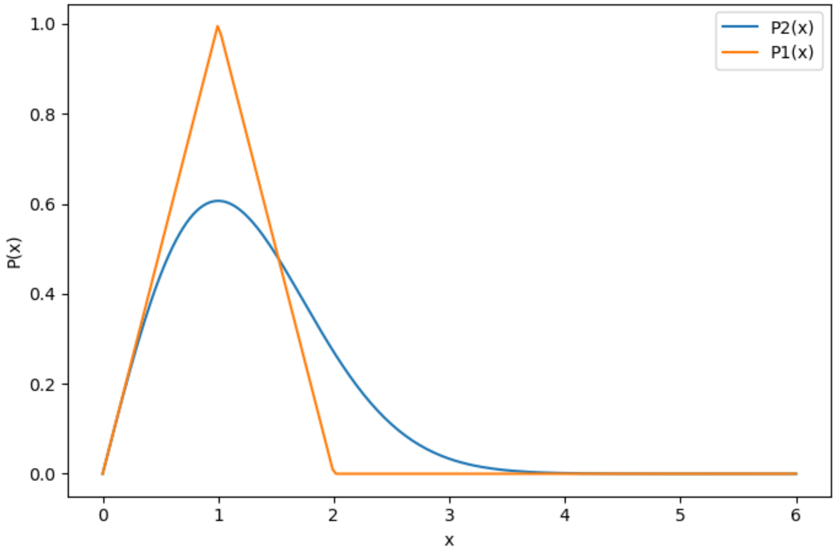
$y = 9 \longrightarrow P_\phi(9) = 0.0587$

$y = 16 \longrightarrow P_\phi(16) = 0.0327$

5.

a.

Sketch of PDFs: (related script is **p5.a**)



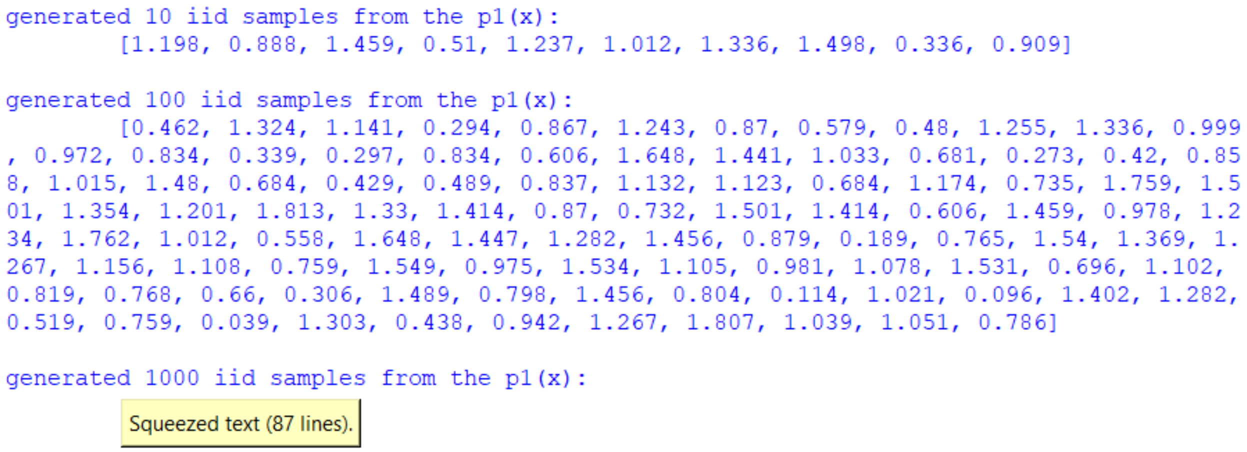
PDFs

b.

Generated iid samples for given values of N are as follows (numbers are rounded with 3 decimal points!):

(related script is **p5.b**)

Results for P1(x):

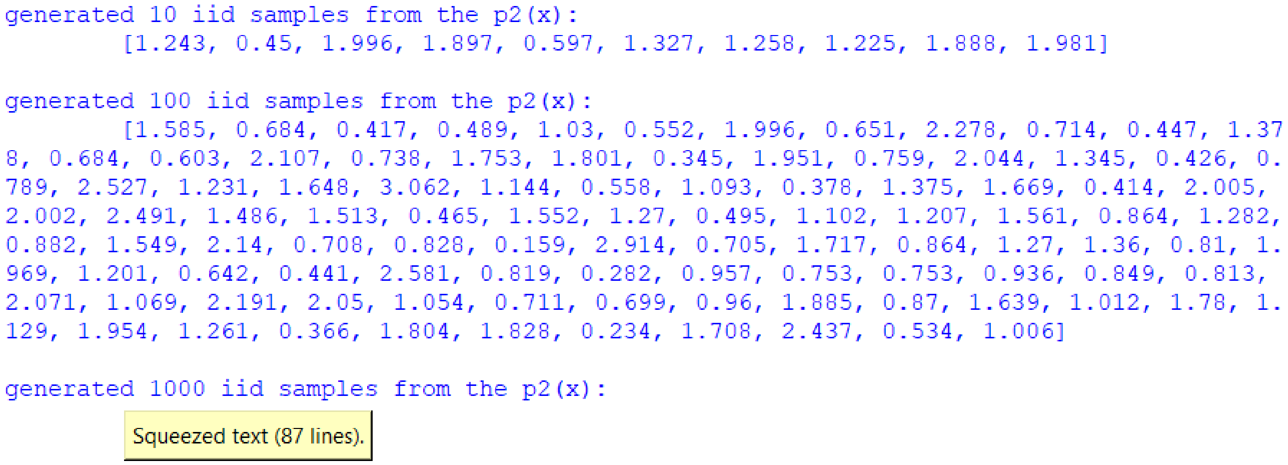


```
generated 10 iid samples from the p1(x):
        [1.198, 0.888, 1.459, 0.51, 1.237, 1.012, 1.336, 1.498, 0.336, 0.909]

generated 100 iid samples from the p1(x):
        [0.462, 1.324, 1.141, 0.294, 0.867, 1.243, 0.87, 0.579, 0.48, 1.255, 1.336, 0.999
, 0.972, 0.834, 0.339, 0.297, 0.834, 0.606, 1.648, 1.441, 1.033, 0.681, 0.273, 0.42, 0.85
8, 1.015, 1.48, 0.684, 0.429, 0.489, 0.837, 1.132, 1.123, 0.684, 1.174, 0.735, 1.759, 1.5
01, 1.354, 1.201, 1.813, 1.33, 1.414, 0.87, 0.732, 1.501, 1.414, 0.606, 1.459, 0.978, 1.2
34, 1.762, 1.012, 0.558, 1.648, 1.447, 1.282, 1.456, 0.879, 0.189, 0.765, 1.54, 1.369, 1.
267, 1.156, 1.108, 0.759, 1.549, 0.975, 1.534, 1.105, 0.981, 1.078, 1.531, 0.696, 1.102,
0.819, 0.768, 0.66, 0.306, 1.489, 0.798, 1.456, 0.804, 0.114, 1.021, 0.096, 1.402, 1.282,
0.519, 0.759, 0.039, 1.303, 0.438, 0.942, 1.267, 1.807, 1.039, 1.051, 0.786]

generated 1000 iid samples from the p1(x):
        Squeezed text (87 lines).
```

Results for P2(x):



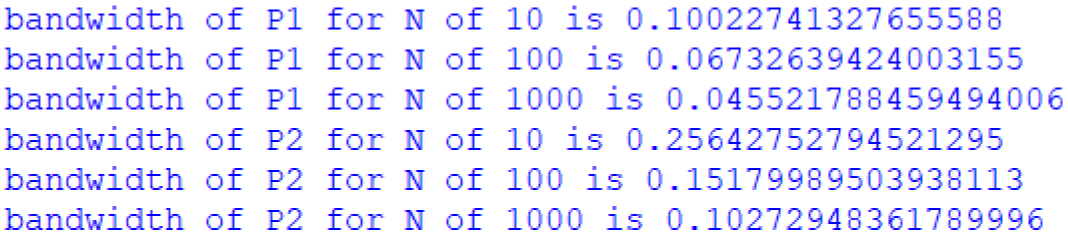
```
generated 10 iid samples from the p2(x):
        [1.243, 0.45, 1.996, 1.897, 0.597, 1.327, 1.258, 1.225, 1.888, 1.981]

generated 100 iid samples from the p2(x):
        [1.585, 0.684, 0.417, 0.489, 1.03, 0.552, 1.996, 0.651, 2.278, 0.714, 0.447, 1.37
8, 0.684, 0.603, 2.107, 0.738, 1.753, 1.801, 0.345, 1.951, 0.759, 2.044, 1.345, 0.426, 0.
789, 2.527, 1.231, 1.648, 3.062, 1.144, 0.558, 1.093, 0.378, 1.375, 1.669, 0.414, 2.005,
2.002, 2.491, 1.486, 1.513, 0.465, 1.552, 1.27, 0.495, 1.102, 1.207, 1.561, 0.864, 1.282,
0.882, 1.549, 2.14, 0.708, 0.828, 0.159, 2.914, 0.705, 1.717, 0.864, 1.27, 1.36, 0.81, 1.
969, 1.201, 0.642, 0.441, 2.581, 0.819, 0.282, 0.957, 0.753, 0.753, 0.936, 0.849, 0.813,
2.071, 1.069, 2.191, 2.05, 1.054, 0.711, 0.699, 0.96, 1.885, 0.87, 1.639, 1.012, 1.78, 1.
129, 1.954, 1.261, 0.366, 1.804, 1.828, 0.234, 1.708, 2.437, 0.534, 1.006]

generated 1000 iid samples from the p2(x):
        Squeezed text (87 lines).
```

c.

the optimal value of bandwidth for each N of each of the distributions are as follows: (related script is **p5.c**)

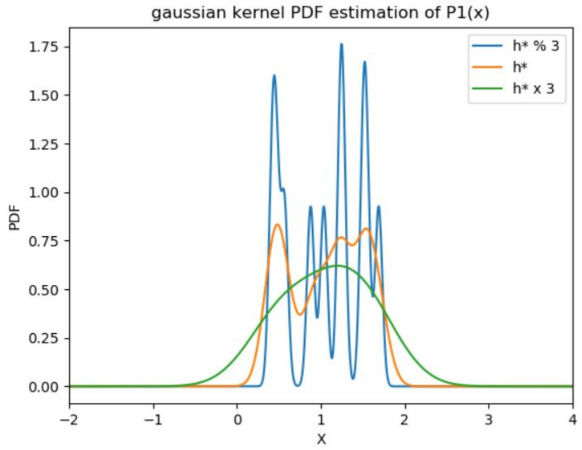


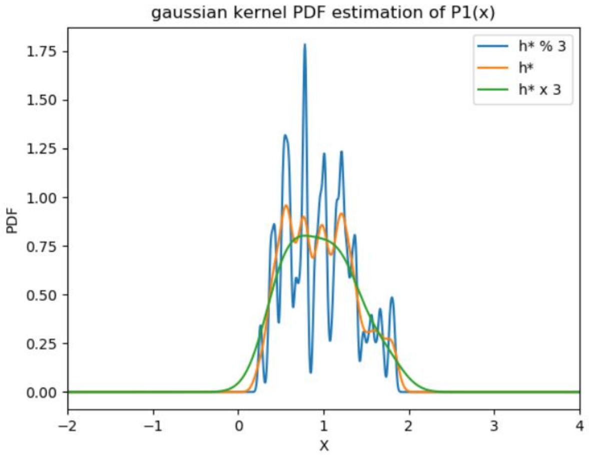
```
bandwidth of P1 for N of 10 is 0.10022741327655588
bandwidth of P1 for N of 100 is 0.06732639424003155
bandwidth of P1 for N of 1000 is 0.045521788459494006
bandwidth of P2 for N of 10 is 0.25642752794521295
bandwidth of P2 for N of 100 is 0.15179989503938113
bandwidth of P2 for N of 1000 is 0.10272948361789996
```

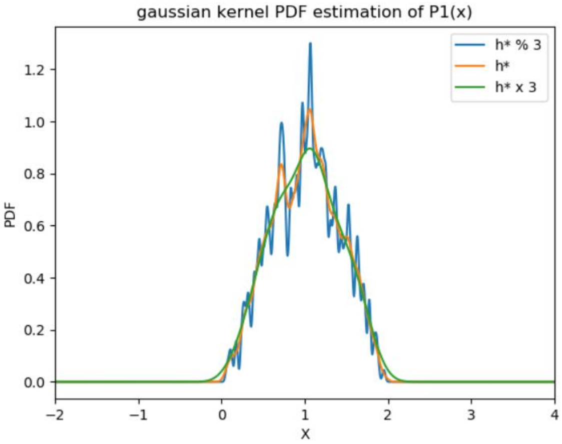
d.

In the following, estimated PDF plots (three plots) for each N random samples (10, 100 and 1000 respectively) were created by each bandwidths (h/3, h and h*3) for each PDFs of P1(x) and P2(x). (related script is **p5.d**)

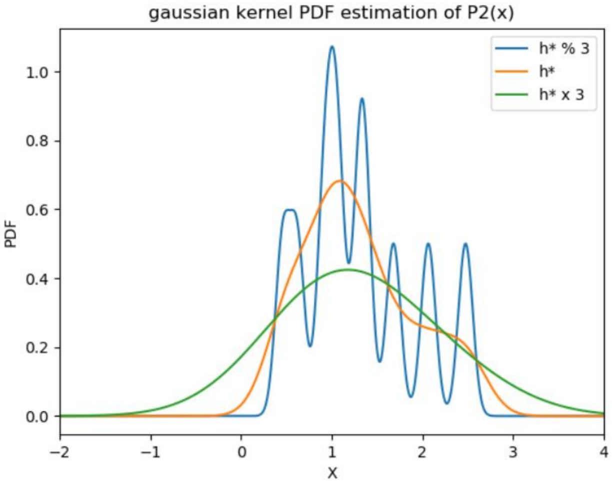
The first three ones are for N of 10, 100 and 1000 respectively and P1(x):

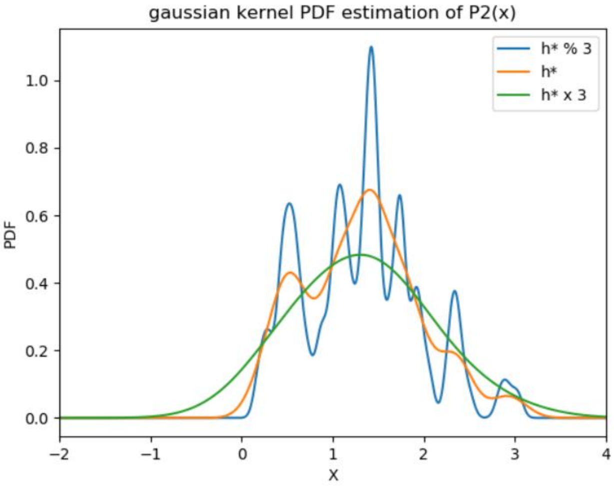


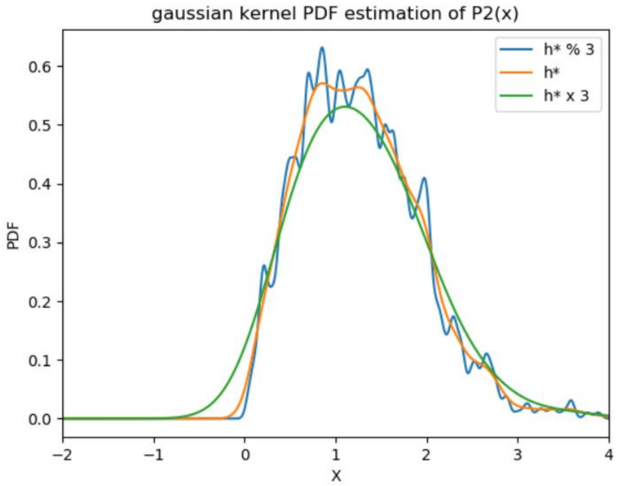




The second three ones are for N of 10, 100 and 1000 respectively and P2(x):



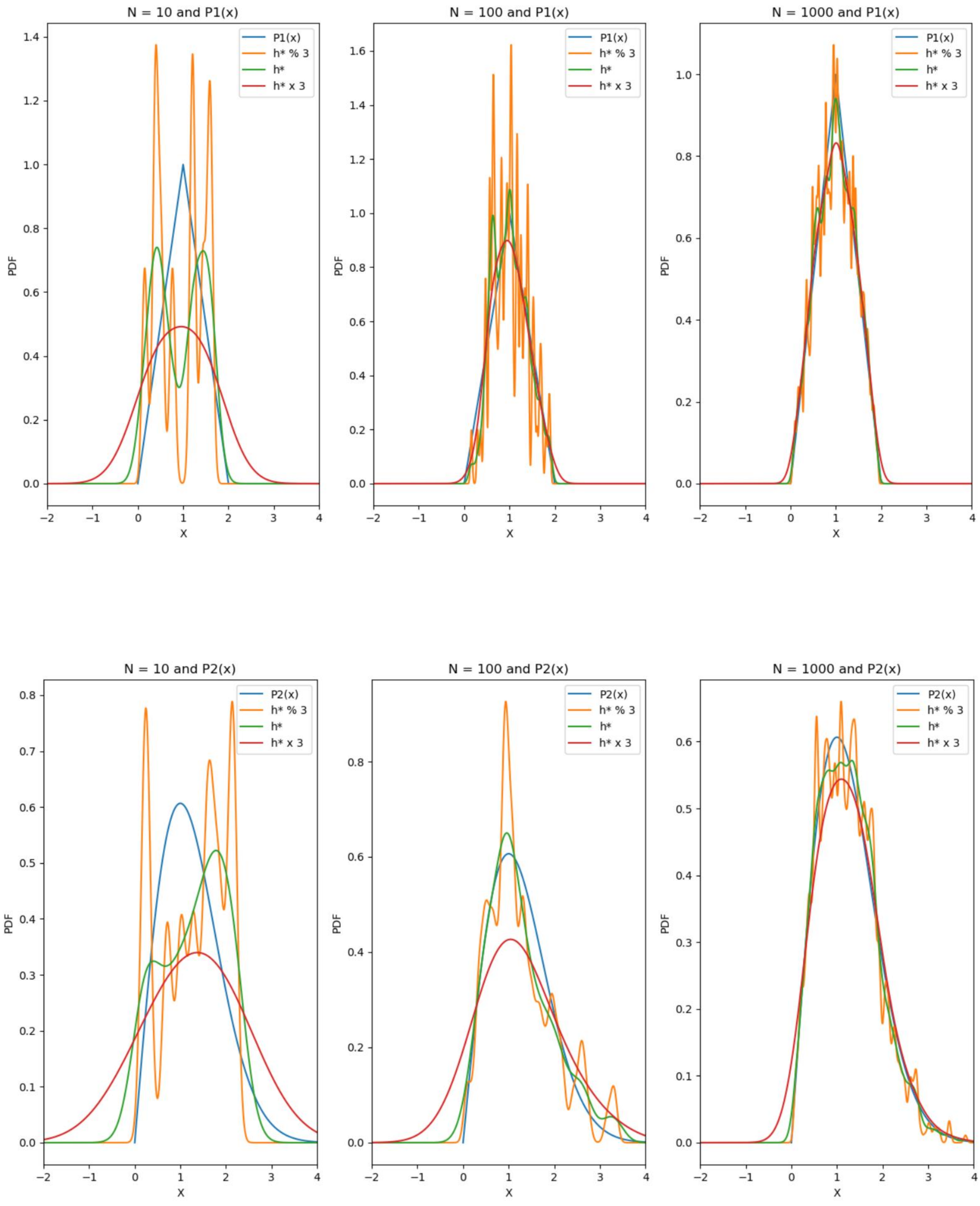




e.

(related script is **p5.e)**

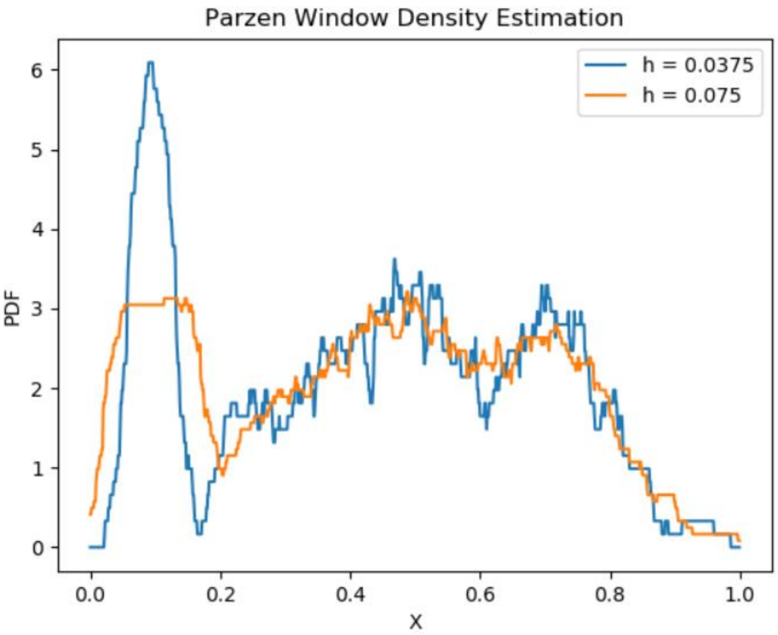
As it can be figured out from the plots below, **considering the effect of parameter 'h'**, it can be seen that as we increase the value of 'h' the plots become smoother for both PDFs which doesn't necessarily mean we have obtained better estimations but also other parameters' values should be considered. **Considering the effect of parameter 'N'**, regardless of other parameters, as we increase the value of 'N', the accuracy of the plots gets higher and they become closer to the groundtruth densities of both PDFs.


N = 10 and P1(x)
N = 100 and P1(x)
N = 1000 and P1(x)
N = 10 and P2(x)
N = 100 and P2(x)
N = 1000 and P2(x)
P1(x)
P2(x)
h* % 3
h*
h* x 3
PDF
X
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0.1
0.3
0.5
0.7
−2
−1
0
1
2
3
4

6.

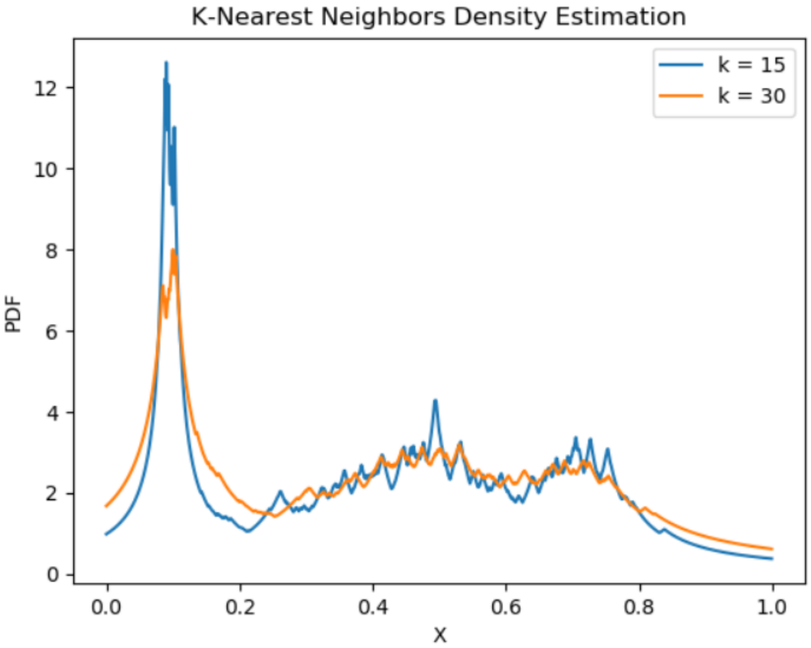
a.

(related script is **p6.a)**



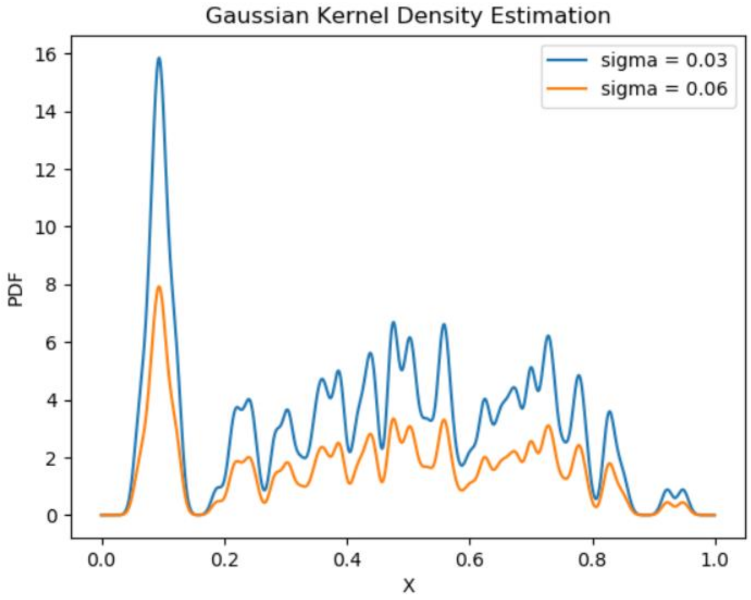
b.

(related script is **p6.b**)



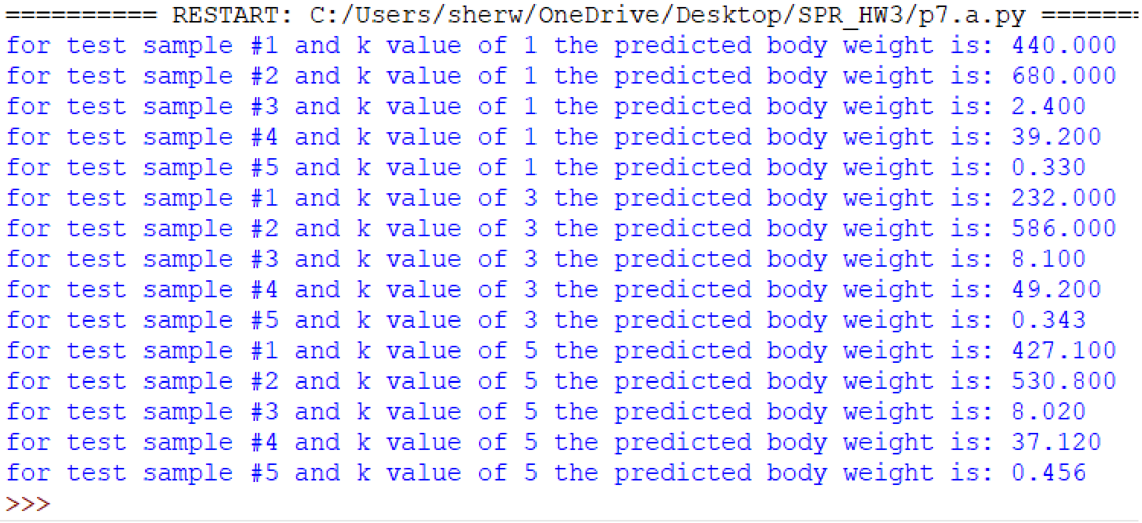
c.

Considering introduced equation in the part P5.c for estimation of optimal bandwidth and the standard deviations which were given in the problem, the following plot was obtained: (related script is **p6.c**)



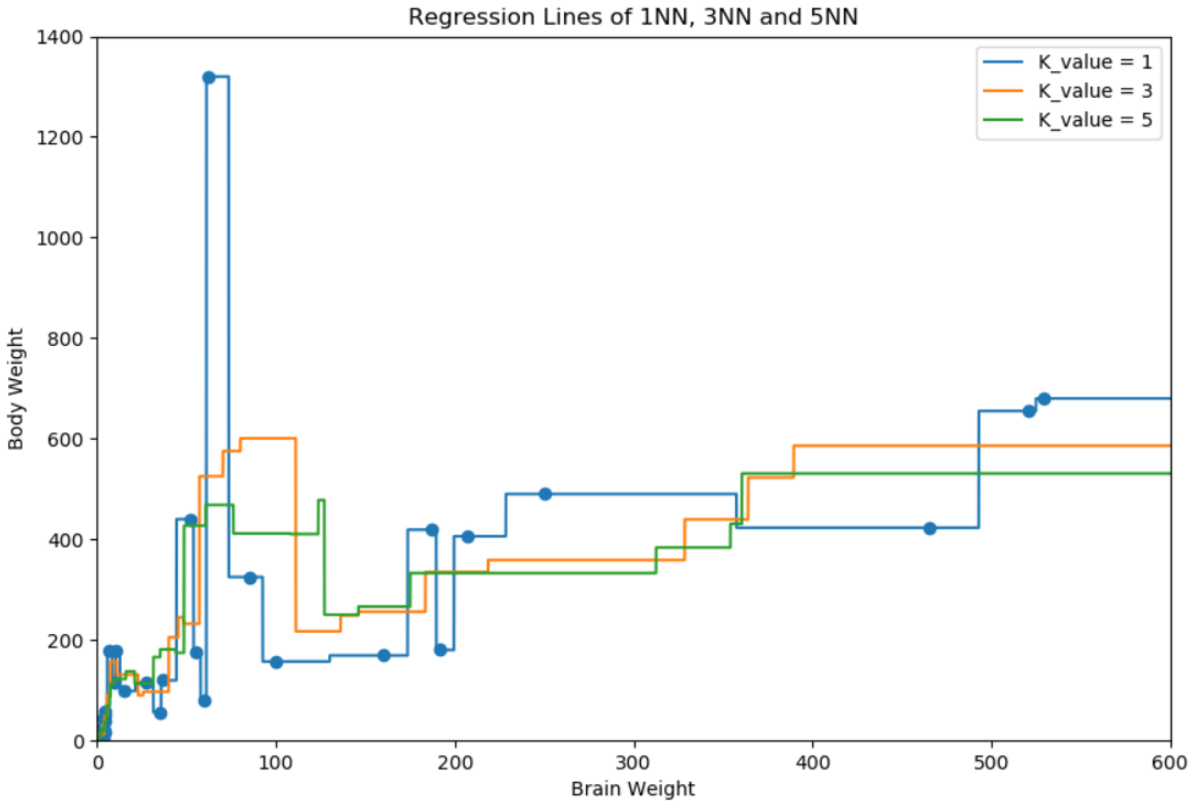
7.

a. (related script is **p7.a**)

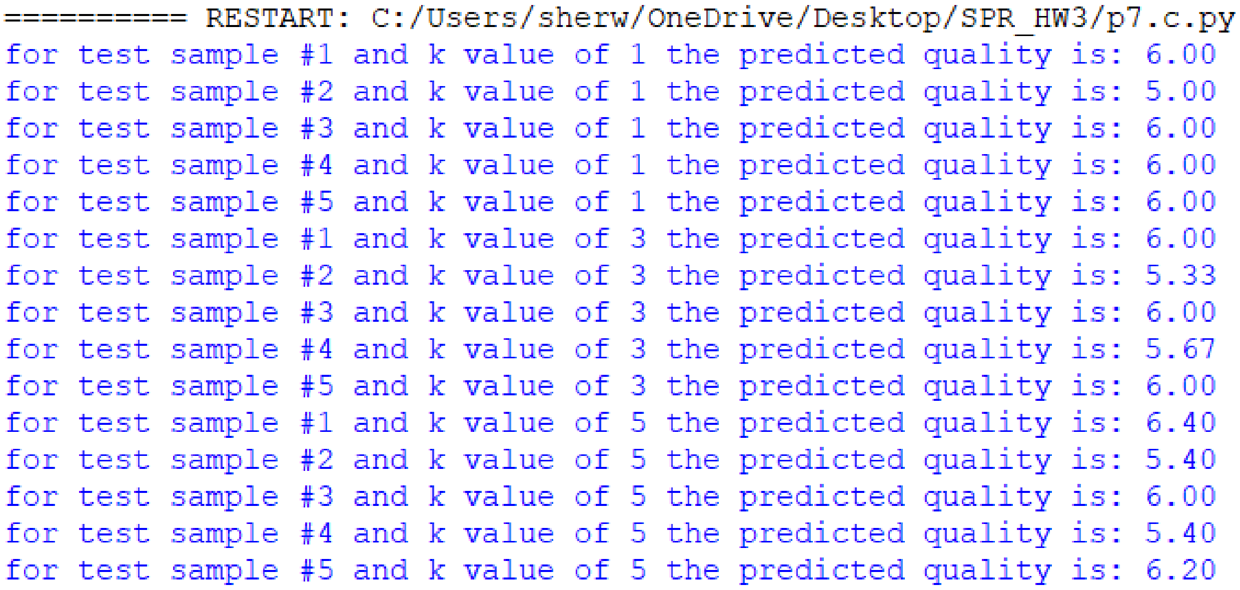


```
========== RESTART: C:/Users/sherw/OneDrive/Desktop/SPR_HW3/p7.a.py =======
for test sample #1 and k value of 1 the predicted body weight is: 440.000
for test sample #2 and k value of 1 the predicted body weight is: 680.000
for test sample #3 and k value of 1 the predicted body weight is: 2.400
for test sample #4 and k value of 1 the predicted body weight is: 39.200
for test sample #5 and k value of 1 the predicted body weight is: 0.330
for test sample #1 and k value of 3 the predicted body weight is: 232.000
for test sample #2 and k value of 3 the predicted body weight is: 586.000
for test sample #3 and k value of 3 the predicted body weight is: 8.100
for test sample #4 and k value of 3 the predicted body weight is: 49.200
for test sample #5 and k value of 3 the predicted body weight is: 0.343
for test sample #1 and k value of 5 the predicted body weight is: 427.100
for test sample #2 and k value of 5 the predicted body weight is: 530.800
for test sample #3 and k value of 5 the predicted body weight is: 8.020
for test sample #4 and k value of 5 the predicted body weight is: 37.120
for test sample #5 and k value of 5 the predicted body weight is: 0.456
>>>
```

b.   (related script is **p7.b**)



c.   (related script is **p7.c**)



```
========== RESTART: C:/Users/sherw/OneDrive/Desktop/SPR_HW3/p7.c.py
for test sample #1 and k value of 1 the predicted quality is: 6.00
for test sample #2 and k value of 1 the predicted quality is: 5.00
for test sample #3 and k value of 1 the predicted quality is: 6.00
for test sample #4 and k value of 1 the predicted quality is: 6.00
for test sample #5 and k value of 1 the predicted quality is: 6.00
for test sample #1 and k value of 3 the predicted quality is: 6.00
for test sample #2 and k value of 3 the predicted quality is: 5.33
for test sample #3 and k value of 3 the predicted quality is: 6.00
for test sample #4 and k value of 3 the predicted quality is: 5.67
for test sample #5 and k value of 3 the predicted quality is: 6.00
for test sample #1 and k value of 5 the predicted quality is: 6.40
for test sample #2 and k value of 5 the predicted quality is: 5.40
for test sample #3 and k value of 5 the predicted quality is: 6.00
for test sample #4 and k value of 5 the predicted quality is: 5.40
for test sample #5 and k value of 5 the predicted quality is: 6.20
```

8. ???

9.

a.

Yes, it is possible. Since, for MLE the derivative is taken with respect to the parameter to be estimated and not the random variable. Thus, despite the random variable which is discrete, the parameter can be continuous.

b.

In practice, prior information is often lacking, hard to put into pdf form, or (worst of all) incorrect. It can be easier to just implement MLE in practice. Function of MAP depends on there being actual correct information about the true state in the prior pdf. Therefore, if the priors are dubious or hard to formulate, it is better to be discarded and trust the data and use MLE. Although MLE may not give us a good enough answer, it's often cheaper, easier, and quicker to collect better (more informtive) data than trying to mess around with expressing prior information we don't really have.

c.

Due to the fact that in calculus, the derivative is a linear operator

[(f + g)’ = f’ + g’]

but not a multiplicative operator, what’s more, when we talk about the MLE of a sample, a product naturally arises because the joint distribution of independent observations $(x1,\ldots,xn)$ is given by the *product* of the marginal distributions of each observation:

$$\mathbf{f(x1,\ldots,xn|\theta)=\prod f(xi|\theta)}$$

Therefore, to find the maximum likelihood, it is usually easier to apply a **monotone transformation (logarithm is a monotonic function)** to the likelihood (thus preserving the location of relative extremum) that **converts multiplication to addition**--this is the logarithm function.

d.

Although KNN method is categorized as Lazy learner, in a sense it doesn’t learn any discriminative function from the training data, but actually it does have a step before inference phase in order to find the closest neighbors efficiently. This step which can be considered as Training Phase consists of arranging the data (sort of indexing process) in order to find the closest neighbors efficiently (as said before). Otherwise, it would have to compare each new case during inference with the whole dataset which leads KNN method become quite inefficient especially for large datasets.

e.

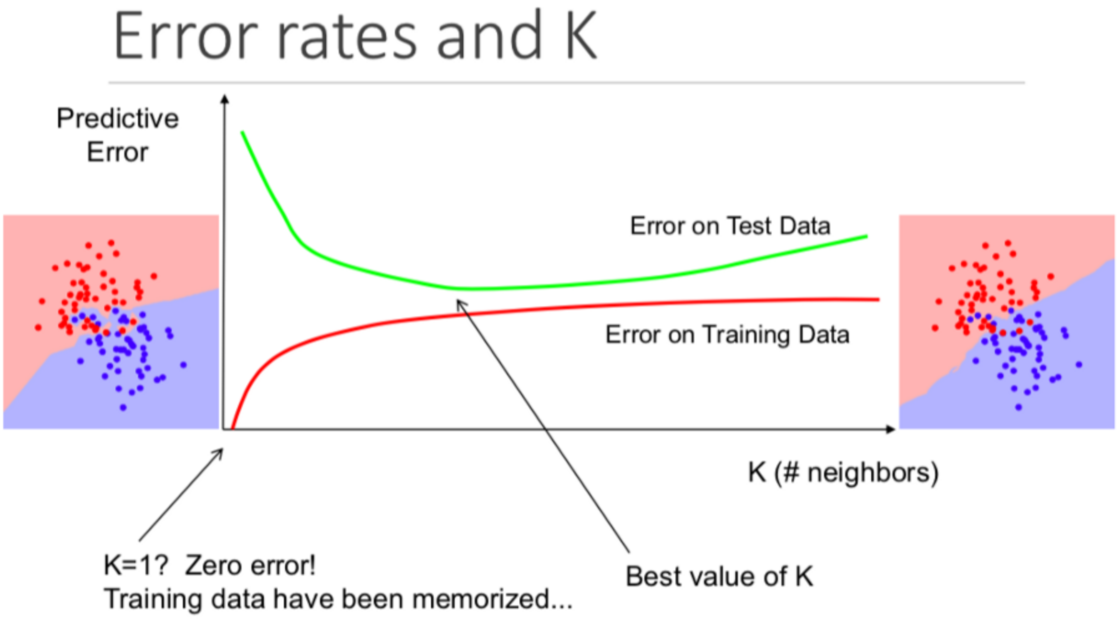
???

f.

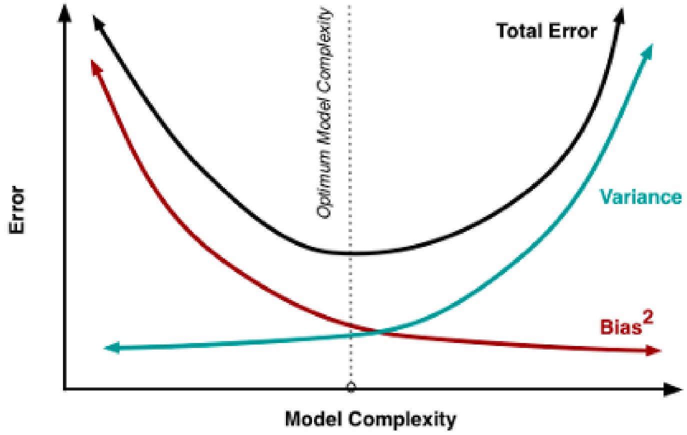
As we know KNN is a lazy learning algorithm. By considering an extreme case K = 1, the training data will perfectly predicted and hence the bias will become 0 when k = 1 but when it comes to new data (in test set) it has higher chance to be an error which leads to high variance. When we increase K the training error will increase (increase bias), but test error may decrease at the same time (decrease variance).

Therefore, for k = 1 we can say that the complexity of the KNN model reaches its peak and by increasing it the complexity decreases.

Now by splitting the data into training and validation set the following graph will be generated for different values of K.

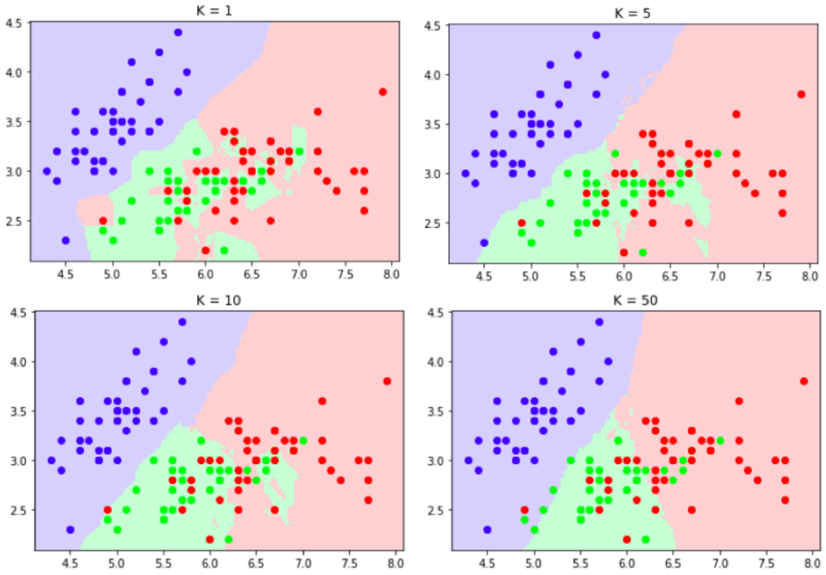


As it can be figured out from the figure above, as we increase value of K from 1, the Error of Training Data keeps constantly increasing but the Error of Test Data decreases to an optimum value of K (complexity) but afterward it starts to increase as K increases. In the following the effect of model complexity (K value in other words) in KNN algorithm on Total Error, Variance and Bias is shown:



As it can be figured out from the graph above there is an optimum value for K (or model complexity) in KNN algorithm.

Also it should be noted that, as the chosen K is closer to the optimum value, the boundaries of the classifier becomes more consistent and reasonable at the same time which can be seen in the following graphs.



Visualization of the results based on different K for KNN algorithm g.

???