CS-215 Assignment-3 Report

22B0914 - Utkarsh Pant
22B0994 - Jaladhi Joshi
22B1005 - Anirudh Garg

Contents

Question 1

PART

I

(a)

$P(X_1)$ is the additional number of times you have to pick a book such that you move from having picked books of i - 1 distinct colors to i distinct colors. Hence $P(X_1) = 1$.
When the $i-1$ colours have been selected, there exists $n-i-1$ colours that have not been selected. Hence, probabilty of selecting a new colour is $\frac{n-i+1}{n}$.

(b)

From the above calculated probabity, we can calculate $P(X_i = k) = (1-\frac{n-i+1}{n})^{k-1}\frac{n-i+1}{n}$ which is basically the probability of not getting a new colour $k-1$ times and then getting a new colour. Hence the parameter p is $\frac{n-i+1}{n}$.

(c)

Let Z we a R.V. with geometric distribution.

$$\begin{aligned}E[Z] &= \sum_{k=1}^{\infty}(1-p)^{k-1}p \cdot k \\ &= p\sum_{k=1}^{\infty}k(1-p)^{k-1}\end{aligned}$$

Note that $\sum_{k=1}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}$,

$$E[Z] = \frac{p}{(1-(1-p))^2} = \frac{1}{p}$$

Note that

$$\sum_{k=1}^{\infty} x^k = x + x^2 + \ldots = \frac{1}{1-x}$$

Differentiating both sides twice we get,

$$\sum_{k=1}^{\infty} k(k-1)x^{k-2} = \frac{2}{(1-x)^3}$$

Putting this in the equation, we get

$$\begin{aligned}
E[Z^2] &= p(1-p)\cdot\frac{2}{p^3}+\frac{1}{p}\\
&= \frac{2p-2p^2+p^2}{p^3}\\
&= \frac{2p-p^2}{p^3}
\end{aligned}$$

$$\begin{aligned}
Var(Z) = E[Z^2] - (E[Z])^2\\
&= \frac{2p-p^2}{p^3}-\frac{1}{p^2}\\
&= \frac{p-p^2}{p^3}\\
&= \frac{1-p}{p^2}
\end{aligned}$$

(d)

$$\begin{aligned}
E[X^{(n)}] &= E[X_1]+\ldots+E[X_n]\\
&= \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n}\frac{1}{p}\\
&= \sum_{i=1}^{n}\frac{n}{n-i+1} = \sum_{i=1}^{n}\frac{n}{i}
\end{aligned}$$

(e)

$X = X_1 + X_2 + \ldots + X_n$
Note that X_i and X_j are independent if $i \neq j$ since $P(X_i)$ is not dependent on $P(X_j)$. So,

$$\begin{aligned}
Var(X^{(n)}) &= Var(X_1)+\ldots+Var(X_n)\\
&= \sum_{i=1}^{n}\frac{1-\frac{n-i+1}{n}}{(\frac{n-i+1}{n})^2}\\
&= \sum_{i=1}^{n}\frac{(i-1)n}{(n-i+1)^2} = \sum_{i=1}^{n}\frac{(-n+i-1)n+n^2}{(n-i+1)^2}\\
&= \sum_{i=1}^{n}\left(\frac{n^2}{(n-i+1)^2}-\frac{n}{(n-i+1)}\right) \qquad = \sum_{i=1}^{n}\frac{n^2}{i^2}-\frac{n}{i}
\end{aligned}$$

Now,
$\sum_{i=1}^{n}\frac{1}{i^2} < \frac{1}{1^2}+\frac{1}{2^2}\ldots = \frac{\pi^2}{6}$
Also $\sum_{i=1}^{n}\frac{1}{i} > n\cdot\frac{1}{n} = 1$

or, $-\sum_{i=1}^{n}\frac{1}{i} < -n\cdot\frac{1}{n} = -1$
So the inequality becomes

$$\begin{aligned} Var(X^{(n)}) &= \sum_{i=1}^{n}\frac{n^2}{i^2} - \frac{n}{i} \\ &< n^2\frac{\pi^2}{6} - n \\ Var(X^{(n)}) &< \frac{(n\pi)^2}{6} - n \end{aligned}$$

(f)

$$\begin{aligned} E[X^{(n)}] &= E[X_1] + \ldots + E[X_n] \\ &= \sum_{i=1}^{n} E[X_i] \\ &= \sum_{i=1}^{n}\frac{n}{n-i+1} = \sum_{i=1}^{n}\frac{n}{i} \end{aligned}$$

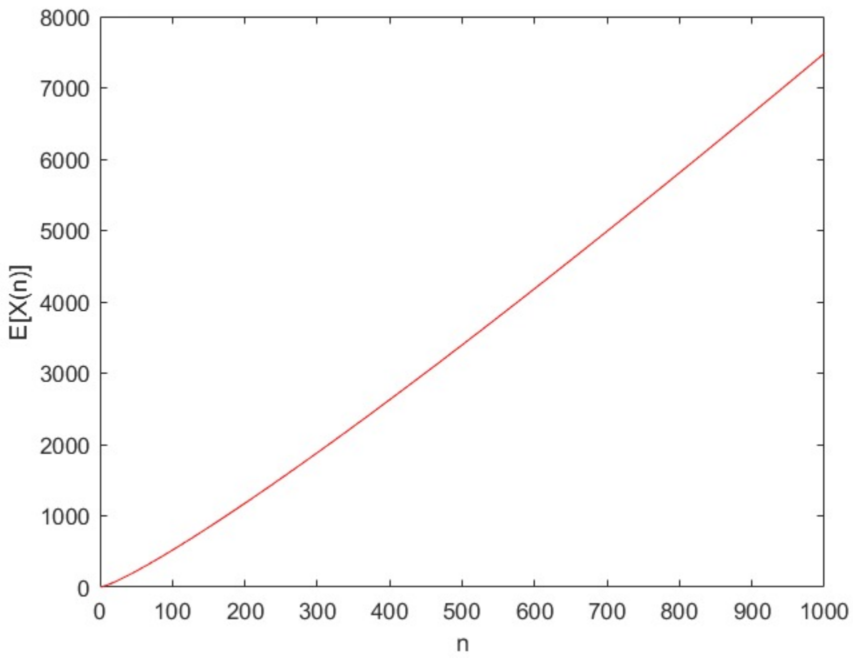

Figure 1. Graph of $E[x^{(n)}]$ vs n

To find the $\Theta(f(n))$, we need to show that both lower and upper bounds of $E[X^{(n)}]$ is $f(n)$.

$$E[X^{(n)}] = n\sum_{i=1}^{n}\frac{1}{i} \leq n(1 + log(n))$$
$$\leq c_1 \cdot nlog(n) \quad \text{for some } c_1$$

Also,

$$E[X^{(n)}] = n\sum_{i=1}^{n}\frac{1}{i} \geq nlog(n+1)$$
$$\geq c_2 \cdot nlog(n) \quad \text{for some } c_2$$

Hence for some c_1, c_2 we can show that

$$c_2 \cdot nlog(n) \leq E[X^{(n)}] \leq c_1 \cdot nlog(n)$$

Hence, $E[X_n] \in \Theta(nlog(n))$

Question 2

PART

II

(a)

Starting with the student's method:

1. $X \sim F_x$ (X is a continuous random variable with CDF F_x).
2. Generate n independent random samples $u_i \sim U(0,1)$ (uniform $[0,1]$ distribution).
3. Calculate $v_i = F_x^{-1}(u_i)$ for each u_i.

   Now, to demonstrate that $\{v_i\}_{i=1}^n$ follows the distribution F:

$$P\left(F_x^{-1}(\{u\}) \leq x\right) = P\left(\{u\} \leq F_x(x)\right) = F_x(x)$$
$$\therefore P\left(F_x^{-1}(\{u\}) \leq x\right) = F_x(x)$$
$$\therefore F_x^{-1}(u_i) = v_i \text{ is a random variable with } F_x \text{ distribution.}$$

   So, the values $\{v_i\}_{i=1}^n$ indeed follow the distribution F_x.

(b)

Let F be a cdf of X and Y be a random variable such that, $Y = F(X)$. Then,

$$F(x) = P(X \leq x)$$

Since $F(x)$ is an increasing function,

$$\begin{aligned} F(x) &= P(F(X) \leq F(x)) \\ &= P(Y \leq F(x)) \textbf{ or} \\ y &= P(Y \leq y) \end{aligned}$$

Hence every distribution derived from a the cdf of a r.v. is a uniform distribution. So let $U_i = F(Y_i) \forall\ i$.

$$\begin{aligned} E &= \max_{0 \leq y \leq 1} \left| \frac{\sum_{i=1}^n \mathbf{1}(U_i \leq y)}{n} - y \right| \\ &= \max_{0 \leq y \leq 1} \left| \frac{\sum_{i=1}^n \mathbf{1}(F(Y_i) \leq y)}{n} - y \right| \end{aligned}$$

Since $0 \leq y \leq 1$ we can also write, $y = F(x)$ for some x. Or,

$$\begin{aligned}
E &= \max_{0 \leq F(x) \leq 1} \left| \frac{\sum_{i=1}^{n} \mathbf{1}(F(Y_i) \leq F(x))}{n} - F(x) \right| \\
&= \max_{-\infty \leq x \leq \infty} \left| \frac{\sum_{i=1}^{n} \mathbf{1}(Y_i \leq x)}{n} - F(x) \right| \\
&= \max_{x} |F_e(x) - F(x)|
\end{aligned}$$

Hence E and D will have same cdf. This implies $P(E > d) = P(D > d)$.

The equality $P(E \geq d) = P(D \geq d)$ allows us to simultaneously assess the uniformity of empirical data and perform consistency checks. When E is large and has a significant difference from D, it signals that the empirical data deviates notably from a uniform distribution and is inconsistent with the chosen theoretical model. This insight is essential for quality control, ensuring that data conforms to expected patterns, and making informed decisions based on data reliability. When E is small, indicating that the empirical data closely resembles a uniform distribution, and D suggests a small difference from the theoretical model, it confirms that the empirical data adheres well to the expected uniformity. This is vital for quality control, ensuring data consistency, and making informed decisions based on reliable data.

Question 3

PART

III

(a)

Least squares plane fitting

Consider the model:

$$z_i = ax_i + by_i + c + \varepsilon_i$$

where $\varepsilon_i \sim N(0, \sigma)$.

We know: a, b, c, $\{x_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$ (accurately), while the $\{z_i\}_{i=1}^n$ estimates are inaccurate.

Now, we can express z_i as a normal distribution:

$$z_i \sim N\left(ax_i + by_i + c, \sigma\right)$$

The probability density function for y_i given x_i, a, b, and c is:

$$p\left(y_i \mid x_i, a, b, c\right) = \frac{e^{\frac{-(z_i-(ax_i+by_i+c))^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}$$

The log-likelihood for $\{y_i\}$ given $\{x_i\}$, a, b, and c is:

$$\log p\left(\{y_i\} \mid \{x_i\}, a, b, c\right) = -\sum_{i=1}^{n}\left[\frac{z_i - (ax_i + by_i + c)}{\sqrt{2}\sigma}\right]^2 - n\log\sqrt{2\pi} - n\log\sigma$$

Taking the partial derivative with respect to a and setting it to zero:

$$\frac{\partial \log p}{\partial a} = \sum_{i=1}^{n} \frac{(z_i - ax_i - by_i - c)x_i}{\sigma^2} = 0$$

This gives us:

$$\sum_{i=1}^{n} x_i z_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i y_i + c\sum_{i=1}^{n} x_i$$

Similarly, taking the partial derivative with respect to b and setting it to zero:

$$\frac{\partial \log p}{\partial b} = \sum_{i=1}^{n} \frac{(z_i - ax_i - by_i - c)y_i}{\sigma^2} = 0$$

This gives us:

$$\sum_{i=1}^{n} y_i z_i = a \sum_{i=1}^{n} x_i y_i + b \sum_{i=1}^{n} y_i^2 + c \sum_{i=1}^{n} y_i$$

Lastly, taking the partial derivative with respect to c and setting it to zero:

$$\sum_{i=1}^{n} \frac{(z_i - a x_i - b y_i - c)}{\sigma^2} = 0$$

This gives us:

$$\sum_{i=1}^{n} z_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} y_i + c(n)$$

Then, we can rewrite the equations in matrix form as:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

(b)

Consider the model:

$$z_i = a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6 + \varepsilon_i$$

where $\varepsilon_i \sim N(0, \sigma^2)$.

The log-likelihood function to be maximized is:

$$\begin{aligned}\log L = &-\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))^2}{2\sigma^2} \\ &- \frac{N}{2}\log\left(2\pi\sigma^2\right)\end{aligned}$$

Taking partial derivatives with respect to the parameters and setting them to zero:

For a_1:

$$\frac{\partial \log L}{\partial a_1} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))x_i^2}{\sigma^2} = 0$$

For a_2:

$$\frac{\partial \log L}{\partial a_2} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))y_i^2}{\sigma^2} = 0$$

For a_3:

$$\frac{\partial \log L}{\partial a_3} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))x_iy_i}{\sigma^2} = 0$$

For a_4:

$$\frac{\partial \log L}{\partial a_4} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))x_i}{\sigma^2} = 0$$

For a_5:

$$\frac{\partial \log L}{\partial a_5} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))y_i}{\sigma^2} = 0$$

For a_6:

$$\frac{\partial \log L}{\partial a_6} = -\sum_{i=1}^{N} \frac{(z_i - (a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6))}{\sigma^2} = 0$$

We get:

$$\sum_{i=1}^{n} x_i^2(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} x_i^2 z_i$$
$$\sum_{i=1}^{n} y_i^2(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} y_i^2 z_i$$
$$\sum_{i=1}^{n} x_iy_i(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} x_iy_iz_i$$
$$\sum_{i=1}^{n} x_i(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} X_iz_i$$
$$\sum_{i=1}^{n} y_i(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} y_iz_i$$
$$\sum_{i=1}^{n}(a_1x_i^2 + a_2y_i^2 + a_3x_iy_i + a_4x_i + a_5y_i + a_6) = \sum_{i=1}^{n} z_i$$

Matrix form:

$$\begin{bmatrix} \sum x_i^4 & \sum x_i^2y_i^2 & \sum x_i^3y_i & \sum x_i^3 & \sum x_i^2y_i & \sum x_i^2 \\ \sum x_i^2y_i^2 & \sum y_i^4 & \sum x_iy_i^3 & \sum x_iy_i^2 & \sum x_iy_i^2 & \sum x_iy_i \\ \sum x_i^3y_i & \sum x_iy_i^3 & \sum x_i^2y_i^2 & \sum x_i^2y_i & \sum x_i^2y_i & \sum x_i^2 \\ \sum x_i^3 & \sum x_iy_i^2 & \sum x_i^2y_i & \sum x_i^2 & \sum x_i^2y_i & \sum x_i \\ \sum x_i^2y_i & \sum x_iy_i^2 & \sum x_iy_i^2 & \sum x_i^2y_i & \sum y_i^2 & \sum x_iy_i \\ \sum x_i^2 & \sum x_iy_i & \sum x_i^2 & \sum x_i & \sum x_iy_i & n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum x_i^2z_i \\ \sum y_i^2z_i \\ \sum x_iy_iz_i \\ \sum x_iz_i \\ \sum y_iz_i \\ \sum z_i \end{bmatrix}$$

(c)

The analysis of the dataset yields the following results:

- **Predicted Plane Equation**:

$$10.002208x + 19.998022y + 29.951579z = 0 \tag{0.1}$$

- **Predicted Noise Variance**:

$$23.091606 \tag{0.2}$$

These results indicate the equation of the predicted plane and the associated noise variance. The plane equation represents the best-fit plane to the data, while the noise variance quantifies the level of variability in the data with respect to the fitted plane.

Here is the MATLAB code used for the analysis:

```
% Read the data from 'XYZ.txt'
data = dlmread('XYZ.txt');

% Extract X, Y, and Z coordinates from the data
X = data(:, 1);
Y = data(:, 2);
Z = data(:, 3);

% Calculate the required summations
n = length(X);
sum_xi_sq = sum(X.^2);
sum_xi_yi = sum(X.*Y);
sum_xi = sum(X);
sum_yi_sq = sum(Y.^2);
sum_yi = sum(Y);
sum_xi_zi = sum(X.*Z);
sum_yi_zi = sum(Y.*Z);
sum_zi = sum(Z);

% Create coefficient matrix A
A = [sum_xi_sq, sum_xi_yi, sum_xi;
     sum_xi_yi, sum_yi_sq, sum_yi;
     sum_xi, sum_yi, n];

% Create the right-hand side vector B
B = [sum_xi_zi;
     sum_yi_zi;
     sum_zi];

```

```
% Solve the linear system to find the coefficients of the plane equation
coefficients = A \ B;

% Extract coefficients for the plane equation
a = coefficients(1);
b = coefficients(2);
c = coefficients(3);

% Calculate the predicted Z values using the plane equation
predicted_Z = a * X + b * Y + c;

% Calculate the residuals
residuals = Z - predicted_Z;

% Calculate the predicted noise variance
predicted_noise_variance = sum(residuals.^2) / (n - 3);

% Display the coefficients of the plane equation
fprintf('Predicted Plane Equation: %.6fx + %.6fy + %.6fz = 0\n', a, b, c);

% Display the predicted noise variance
fprintf('Predicted Noise Variance: %.6f\n', predicted_noise_variance);
```

Listing 1. MATLAB Code for Analysis

PART

IV

Question 4

(a)

T and V are generated using randperm function and the distribution is generated using the normrnd function of matlab.

```
clear;
n = 1000;
mu = 0;
sigma = sqrt(16);
data = normrnd(mu, sigma, n, 1);
T_size = 750;
rand_indices = randperm(n);
T_indices = rand_indices(1:T_size);
V_indices = rand_indices(T_size+1:end);
T = data(T_indices);
V = data(V_indices);
```

Listing 2. MATLAB Code

(b)

For the training set T and bandwidth parameter σ, the pdf estimate

$$\hat{p}_n(x;\sigma) = \frac{1}{750\sigma\sqrt{2\pi}} \sum_{i=1}^{750} \exp\left(-\frac{(x-T_i)^2}{2\sigma^2}\right)$$

In this expression:

- x is the value at which you want to estimate the PDF.
- T_i represents each of the values in the training set T.

Based on this,the expression for the Joint Likelihood of the samples in V is

$$L(V|T,\sigma) = \prod_{j=1}^{250} \left(\sum_{i=1}^{750} \frac{1}{750\sigma\sqrt{2\pi}} \exp\left(-\frac{(V_j-T_i)^2}{2\sigma^2}\right)\right)$$

(c)

The provided MATLAB code calculates the log likelihood (LL) of observing the validation samples (V) based on a PDF estimate built from the training set (T) for various bandwidth parameters (σ) ranging from very small to large values. It iteratively computes the log likelihood for each σ value, considering the contribution of each validation sample to the overall likelihood. The resulting plot of LL versus $\log \sigma$ helps visualize how the bandwidth parameter affects the likelihood of observing the validation data under the estimated PDF.

```
clear;
n = 1000;
mu = 0;
sigma = sqrt(16);
data = normrnd(mu, sigma, n, 1);
T_size = 750;
rand_indices = randperm(n);
T_indices = rand_indices(1:T_size);
V_indices = rand_indices(T_size+1:end);
T = data(T_indices);
V = data(V_indices);
sigmas = [0.001, 0.1, 0.2, 0.9, 1, 2, 3, 5, 10, 20, 100];
LL = zeros(size(sigmas));
for s = 1:length(sigmas)
    sigma = sigmas(s);
    log_likelihood = 0;
     for j = 1:length(V)
        x = V(j);
        likelihood = 0;
         for i = 1:length(T)
                Ti = T(i);
                likelihood = likelihood + (1 / (750 * sigma *
                   sqrt(2 * pi))) ...
                    * exp(-(x - Ti)^2 / (2 * sigma^2));
         end
         log_likelihood = log_likelihood + log(likelihood);
     end
     LL(s) = log_likelihood;
end
figure;
plot(log(sigmas), LL, '-o');
xlabel('log(\sigma)');
ylabel('Log Likelihood (LL)');
title('Log Likelihood vs. log(\sigma)');
grid on;
[best_LL, best_sigma_idx] = max(LL);
best_sigma = sigmas(best_sigma_idx);
fprintf('Best sigma value: %.3f\n', best_sigma);
fprintf('Corresponding LL value: %.3f\n', best_LL);
```

Listing 3. MATLAB Code

σ=1 yeilded the best LL value

The code for comparing estimated and true density is:

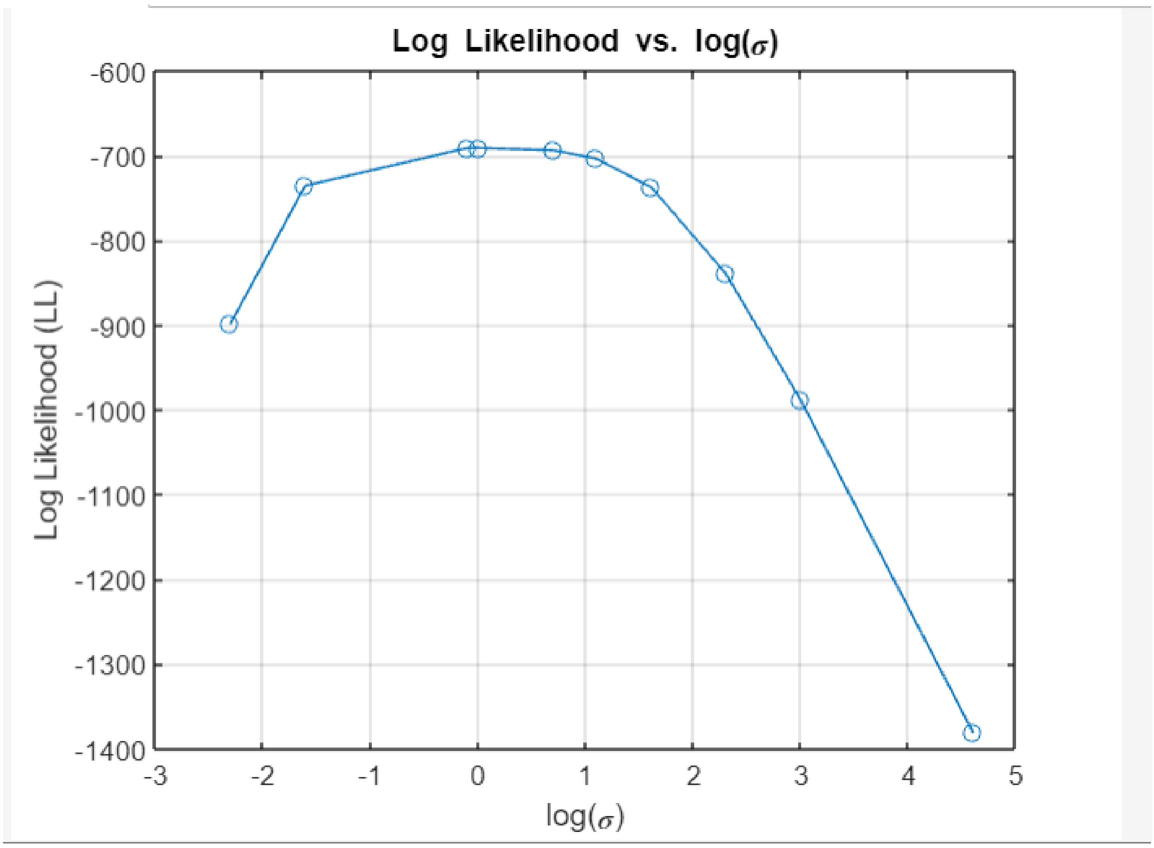

Figure 2. Plot of Log Likelihood vs. $\log \sigma$

```
clear;
n = 1000;
mu = 0;
sigma = sqrt(16);
data = normrnd(mu, sigma, n, 1);
T_size = 750;
rand_indices = randperm(n);
T_indices = rand_indices(1:T_size);
V_indices = rand_indices(T_size+1:end);
T = data(T_indices);
V = data(V_indices);
sigmas = [0.001, 0.1, 0.2, 0.9, 1, 2, 3, 5, 10, 20, 100];
LL = zeros(size(sigmas));
for s = 1:length(sigmas)
    sigma = sigmas(s);
    log_likelihood = 0;
     for j = 1:length(V)
        x = V(j);
        likelihood = 0;
         for i = 1:length(T)
                Ti = T(i);
                likelihood = likelihood + (1 / (750 * sigma *
```

```
                    sqrt(2 * pi))) ...
                     * exp(-(x - Ti)^2 / (2 * sigma^2));
          end
          log_likelihood = log_likelihood + log(likelihood);
      end
      LL(s) = log_likelihood;
end
figure;
plot(log(sigmas), LL, '-o');
xlabel('log(\sigma)');
ylabel('Log Likelihood (LL)');
title('Log Likelihood vs. log(\sigma)');
grid on;
[best_LL, best_sigma_idx] = max(LL);
best_sigma = sigmas(best_sigma_idx);
fprintf('Best sigma value: %.3f\n', best_sigma);
fprintf('Corresponding LL value: %.3f\n', best_LL);
x_values = -8:0.1:8;

% Calculate the estimated PDF using the best sigma
pdf_estimate = zeros(size(x_values));
for j = 1:length(x_values)
    x = x_values(j);
    pdf_value = 0;
    for i = 1:length(T)
        Ti = T(i);
        pdf_value = pdf_value + (1 / (T_size * best_sigma *
            sqrt(2 * pi))) ...
             * exp(-(x - Ti)^2 / (2 * best_sigma^2));
    end
    pdf_estimate(j) = pdf_value;
end

% Calculate the true density (N(0, 4)) for the same x values
true_density = normpdf(x_values, 0, 4); % Mean = 0, Standard
    Deviation = 4

% Plot the estimated PDF and the true density
figure;
plot(x_values, pdf_estimate, 'b', 'LineWidth', 2);
hold on;
plot(x_values, true_density, 'r', 'LineWidth', 2);
xlabel('x');
ylabel('Probability Density');
title('Estimated PDF vs. True Density');
legend('Estimated PDF', 'True Density');
grid on;
hold off;
```

Listing 4. MATLAB Code

The plot comparing both is:

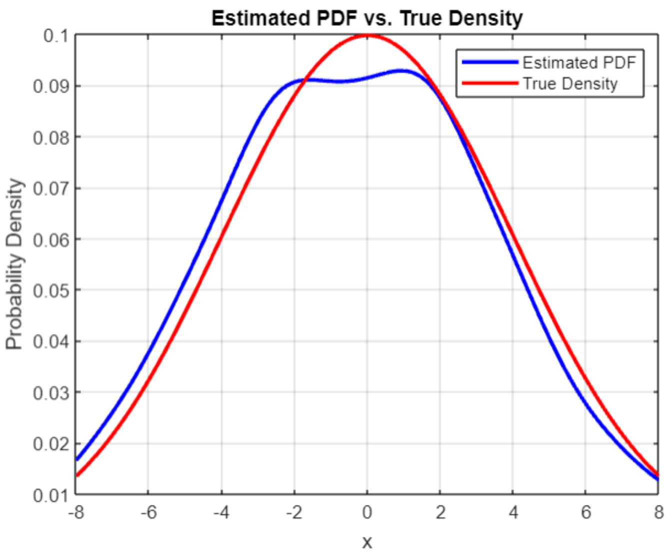

Figure 3. Estimated PDF vs. True Density

(d)

For D calculations the code is as follows:

```
clear;
n = 1000;
mu = 0;
sigma = sqrt(16);
data = normrnd(mu, sigma, n, 1);
T_size = 750;
rand_indices = randperm(n);
T_indices = rand_indices(1:T_size);
V_indices = rand_indices(T_size+1:end);
T = data(T_indices);
V = data(V_indices);
sigmas = [0.001, 0.1, 0.2, 0.9, 1, 2, 3, 5, 10, 20, 100];
LL = zeros(size(sigmas));
for s = 1:length(sigmas)
    sigma = sigmas(s);
    log_likelihood = 0;
     for j = 1:length(V)
        x = V(j);
        likelihood = 0;
         for i = 1:length(T)
                Ti = T(i);
```

```
                likelihood = likelihood + (1 / (750 * sigma * sqrt(2 * pi))) ...
                    * exp(-(x - Ti)^2 / (2 * sigma^2));
        end
        log_likelihood = log_likelihood + log(likelihood);
    end
    LL(s) = log_likelihood;
end
[best_LL, best_sigma_idx] = max(LL);
best_sigma = sigmas(best_sigma_idx);

D_values = zeros(size(sigmas));
true_mean = 0;
true_stddev = 4;
p_x = @(x) (1 / (true_stddev * sqrt(2 * pi))) * exp(-(x - true_mean).^2 / (2 * true_stddev^2));

for s = 1:length(sigmas)
    sigma = sigmas(s);
    D = 0;

    for j = 1:length(V)
        xi = V(j);
        D = D + (p_x(xi) - calculate_pdf(xi, T, T_size, sigma))^2;
    end

    D_values(s) = D;
end
[best_D, best_sigma_idx_D] = min(D_values);
best_sigma_D = sigmas(best_sigma_idx_D);

fprintf('Best sigma value based on D: %.3f\n', best_sigma_D);
fprintf('D value for the    parameter which yielded the best LL (  =%.3f): %.3f\n', best_sigma_D, D_values(best_sigma_idx_D));

% Plot D versus log
figure;
plot(log(sigmas), D_values, 'bo-');
xlabel('log(\sigma)');
ylabel('D');
title('D vs. log(\sigma)');
grid on;

function pdf = calculate_pdf(x, T, T_size, sigma)
    pdf_value = 0;
    for i = 1:length(T)
        Ti = T(i);
```

```
        pdf_value = pdf_value + (1 / (T_size * sigma * sqrt(2 * pi))) ...
            * exp(-(x - Ti)^2 / (2 * sigma^2));
    end
    pdf = pdf_value;
end
```

Listing 5. MATLAB Code

Best sigma value based on D: 1.000

D value for the σ parameter which yielded the best LL: 0.005

For the best sigma that gives an appropriate value of D , we compare the true and

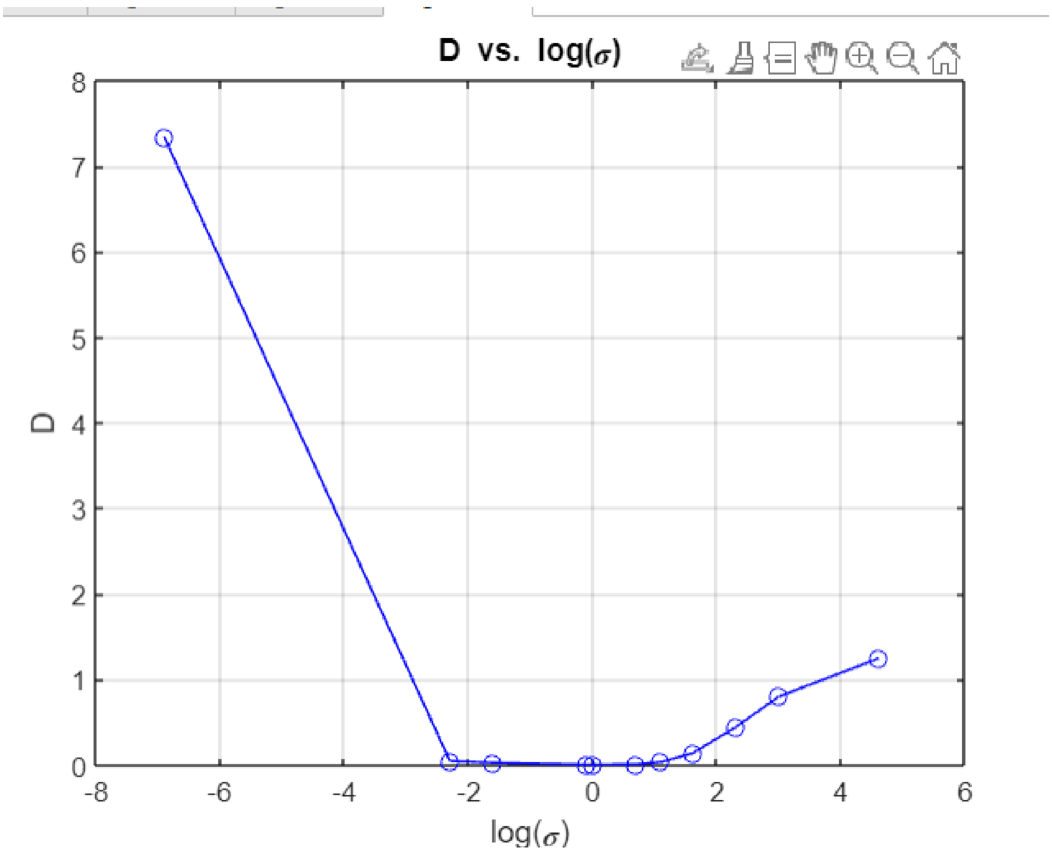

Figure 4. D vs. log(σ) plot

estimated values of density as follows:

```
rng(123);
clear;
n = 1000;
mu = 0;
sigma = sqrt(16);
data = normrnd(mu, sigma, n, 1);
T_size = 750;
rand_indices = randperm(n);
T_indices = rand_indices(1:T_size);
```

```
V_indices = rand_indices(T_size+1:end);
T = data(T_indices);
V = data(V_indices);
sigmas = [0.001, 0.1, 0.2, 0.9, 1, 2, 3, 5, 10, 20, 100];
LL = zeros(size(sigmas));
for s = 1:length(sigmas)
    sigma = sigmas(s);
    log_likelihood = 0;
     for j = 1:length(V)
        x = V(j);
        likelihood = 0;
         for i = 1:length(T)
                Ti = T(i);
                likelihood = likelihood + (1 / (750 * sigma * sqrt(2 * pi))) ...
                    * exp(-(x - Ti)^2 / (2 * sigma^2));
         end
         log_likelihood = log_likelihood + log(likelihood);
     end
     LL(s) = log_likelihood;
end
[best_LL, best_sigma_idx] = max(LL);
best_sigma = sigmas(best_sigma_idx);

D_values = zeros(size(sigmas));
true_mean = 0;
true_stddev = 4;
p_x = @(x) (1 / (true_stddev * sqrt(2 * pi))) * exp(-(x - true_mean).^2 / (2 * true_stddev^2));

for s = 1:length(sigmas)
    sigma = sigmas(s);
    D = 0;

    for j = 1:length(V)
        xi = V(j);
        D = D + (p_x(xi) - calculate_pdf(xi, T, T_size, sigma))^2;
    end

    D_values(s) = D;
end
[best_D, best_sigma_idx_D] = min(D_values);
best_sigma_D = sigmas(best_sigma_idx_D);

fprintf('Best sigma value based on D: %.3f\n', best_sigma_D);
fprintf('D value for the    parameter which yielded the best LL (  =%.3f): %.3f\n', best_sigma_D, D_values(best_sigma_idx_D));
```

```
% Plot D versus log
figure;
plot(log(sigmas), D_values, 'bo-');
xlabel('log(\sigma)');
ylabel('D');
title('D vs. log(\sigma)');
grid on;
x_values = -8:0.1:8;
pdf_estimate = zeros(size(x_values));
for j = 1:length(x_values)
    x = x_values(j);
    pdf_value = 0;
    for i = 1:length(T)
        Ti = T(i);
        pdf_value = pdf_value + (1 / (T_size * best_sigma_D *
            sqrt(2 * pi))) ...
             * exp(-(x - Ti)^2 / (2 * best_sigma_D^2));
    end
    pdf_estimate(j) = pdf_value;
end

% Calculate the true density (N(0, 4)) for the same x values
true_density = normpdf(x_values, 0, 4); % Mean = 0, Standard
    Deviation = 4

% Plot the estimated PDF and the true density
figure;
plot(x_values, pdf_estimate, 'b', 'LineWidth', 2);
hold on;
plot(x_values, true_density, 'r', 'LineWidth', 2);
xlabel('x');
ylabel('Probability Density');
title('Estimated PDF vs. True Density');
legend('Estimated PDF', 'True Density');
grid on;
hold off;
function pdf = calculate_pdf(x, T, T_size, sigma)
    pdf_value = 0;
    for i = 1:length(T)
        Ti = T(i);
        pdf_value = pdf_value + (1 / (T_size * sigma * sqrt(2
            * pi))) ...
             * exp(-(x - Ti)^2 / (2 * sigma^2));
    end
    pdf = pdf_value;
end
```

The obtained plot is as follows:

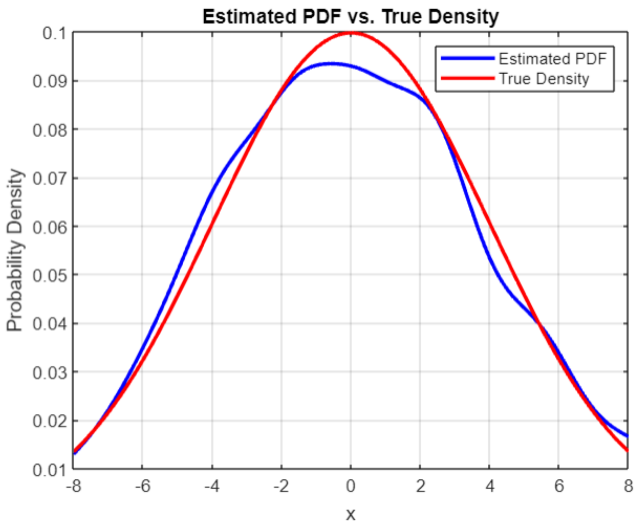

Figure 5. Comparison of Estimated and True Densities for σ = BestSigma

(e)

What Happens:

In the context of the cross-validation procedure with equally sized training and validation sets:

1. **Data Usage:** When T and V are equal, the entire dataset is used for both training and validation simultaneously. There's no clear distinction between training and validation data.

2. **Parameter Estimation:** Normally, the parameter σ (bandwidth) is estimated using T and then evaluated on V. However, with equal T and V, σ is chosen based on the same data it was estimated from.

3. **Performance Assessment:** Cross-validation aims to provide an unbiased assessment of model performance on unseen data. But in this scenario, performance metrics, such as likelihood or density estimation accuracy, will be overly optimistic. The model essentially memorizes the training data.

4. **Sigma and Best Sigma:** Sigma, the bandwidth parameter, may not generalize well to new data as it's chosen to fit the same data it was derived from. The "best" sigma, determined during cross-validation, may not be suitable for unseen data.

5. **D and PDF Estimation:** The D value, representing the squared difference between true and estimated densities, will be artificially low since the model has seen and memorized the data. Similarly, PDF estimation may show unrealistically high accuracy due to overfitting.

Why:

1. **Loss of Generalization Assessment:** In standard cross-validation, separate training and validation sets provide insight into a model's generalization ability. When T equals V, this assessment is lost since there's no distinction between training and validation data.

2. **Loss of Unbiased Performance Estimation:** Cross-validation aims to provide unbiased performance estimates on new data. However, when T and V coincide, this objectivity is compromised, and performance estimates may be overly optimistic due to data reuse.

3. **Overfitting Risk:** Equal T and V sets increase the risk of overfitting, where the model fits the training data too closely and captures noise rather than the underlying pattern. This can lead to poor generalization to new, unseen data.

4. **Sigma Choice:** The choice of sigma, based on training data, may not generalize well to new data, as it's tailored to fit the specific training set. This can lead to suboptimal parameter selection.

5. **Misleading Metrics:** Metrics like D and PDF estimation accuracy may provide misleadingly positive results since the model has essentially memorized the data it's tested on.

Question 5

PART

V

First we'll assume the given inequality and try to find the upper bound on $P(S_n - E[X_n] > t)$,
We can write the following.

$$P(S_n - E[X_n] > t) = P(e^{S_n - E[X_n]} > e^t) \quad \text{where } S_n = X_1 + \ldots + X_n$$

This is because $e^s x$ is an increasing function. Since the LHS of inequality is a positive valued, we can apply Markov's property.

$$\begin{aligned} P(e^{S_n - E[X_n]} > e^t) &< \frac{E[e^{s(S_n - E[X_n])}]}{e^t} \\ &< \frac{E[e^{s(X_1 + X_2 + \ldots + X_n - (E[X_1] + E[X_2] + \ldots + E[X_n]))}]}{e^{st}} \end{aligned}$$

Since $X_1, X_2 \ldots X_n$ are independent, it implies $e^{s(X_1 - E[X_1])}, e^{s(X_2 - E[X_2])} \ldots e^{s(X_n - E[X_n])}$ are independent too.
Also for independent R.V. we get that the expectation of product equals the product of exxpectaion of R.V.s

$$\begin{aligned} P(S_n - E[X_n] > t) &< \frac{E[e^{s(X_1 + X_2 + \ldots + X_n - (E[X_1] + E[X_2] + \ldots + E[X_n]))}]}{e^{st}} \\ &< \frac{E[e^{s(X_1 - E[X_1])}]E[e^{s(X_2 - E[X_2])}] \ldots E[e^{s(X_n - E[X_n])}]}{e^{st}} \end{aligned}$$

Using the given inequality..

$$\begin{aligned} P(S_n - E[S_n] > t) &< \frac{e^{s^2 \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{8}}}{e^{st}} \quad \text{or,} \\ P(S_n - E[S_n] > t) &< e^{s^2 \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{8} - st} \end{aligned}$$

Let $T = \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{8}$,
for $t \geq 0$
Inorder to tighten the bound, we need to find the minimise the power of the exponent. Or differentiating the power w.r.t. s we get,

$$\begin{aligned} 2sT - t &= 0 \\ s &= \frac{t}{2T} \end{aligned}$$

So the exponent becomes,

$$\begin{aligned}
P(S_n - E[S_n] > t) &< e^{\frac{t^2}{4T^2}T - \frac{t^2}{2T}} \\
&< e^{-\frac{t^2}{4T}} \\
P(S_n - E[S_n] > t) &< e^{-\frac{t^2}{4\sum_{i=1}^{n}\frac{(a_i-b_i)^2}{8}}}
\end{aligned}$$

$$\boxed{P(S_n - E[S_n] > t) < e^{-\frac{2t^2}{\sum_{i=1}^{n}(a_i-b_i)^2}}}$$

for $t < 0$
The exponent is increasing for $s > 0$. Inorder to tighten the bound we put $s = 0$. Hence the exponent becomes 1.

$$\boxed{P(S_n - E[S_n] > t) < 1}$$

(a)

Since e^{sx} is a convex function, we get $E(e^{sx}) \leq e^{L(s(b-a))}$. Also VLOG we have assumed $E[X] = 0$.

(b)

$$e^{sx} \leq \frac{(b-x)e^{sa}}{b-a} + \frac{(x-a)e^{sb}}{b-a}$$

Taking expectation both sides

$$\begin{aligned}
E[e^{sx}] &\leq E[\frac{(b-x)e^{sa}}{b-a} + \frac{(x-a)e^{sb}}{b-a}] \\
&\leq \frac{1}{b-a}E[(b-x)e^{sa} + (x-a)e^{sb}] \\
&\leq \frac{1}{b-a}((b-E[x])e^{sa} + (E[x]-a)e^{sb}) \\
&\leq \frac{(be^{sa} - ae^{sb})}{b-a} \\
&\leq e^{sa}\frac{(b-a+a-ae^{s(b-a)})}{b-a} \\
&\leq e^{sa}\left(1 + \frac{(a-ae^{s(b-a)})}{b-a}\right) \\
&\leq e^{log\left(e^{sa}\left(1+\frac{(a-ae^{s(b-a)})}{b-a}\right)\right)} \\
&\leq e^{sa+log\left(\left(1+\frac{(a-ae^{s(b-a)})}{b-a}\right)\right)} \\
&\leq e^{\frac{sa(b-a)}{(b-a)}+log\left(\left(1+\frac{(a-ae^{s(b-a)})}{b-a}\right)\right)} \\
E[e^{sx}] &\leq e^{L(s(b-a))}
\end{aligned}$$

Where $L(h) = \dfrac{ha}{b-a} + \log\Big(1 + (a - ae^h)/(b-a)\Big)$.

(c)

Differentiating $L(h)$ twice we get,

$$\begin{aligned}
L^{`}(h) &= \frac{a}{a-b} + \frac{ae^h}{ae^h - b} \\
L^{``}(h) &= \frac{abe^h}{(ae^h - b)^2} \\
L^{``}(h) &= -\frac{ab}{a^2e^h + b^2e^{-h} - 2ab}
\end{aligned}$$

The minimum value of denominator can be found using AM-GM;

$$2\sqrt{a^2b^2} - 2ab = -2ab - 2ab = -4ab$$

Hence, $L^{``}(h)$ will maximise when the denominator minimises.

$$L^{``}(h) \leq \frac{1}{4}$$

(d)

Integrating both sides we get from 0 to h, twice.

$$\begin{aligned}
L^{`}(h) &\leq \frac{h}{4} \\
L^{``}(h) &\leq \frac{h^2}{8}
\end{aligned}$$

putting h as $s(b-a)$, we get

$$\begin{aligned}
L(h) &\leq \frac{s^2(b-a)^2}{8} \\
&\text{or,} \\
E[e^{sx}] &\leq e^{L(s(b-a))} \leq e^{\frac{s^2(b-a)^2}{8}}
\end{aligned}$$