UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING - ITEE

STATISTICAL SIGNAL PROCESSING 1

MATLAB TASK #2

Group: 14

Students:
- AZZAZ Aissa (Number: 2207335)
- BOULFRAD Mourad (Number: 2207592)

Due date: October 3rd 2022

1. Estimation using linear model

The observation consists of two samples $X = [x[0], x[1]]^T$ of DC level A in the presence of Gaussian noise $W = [w[0], w[1]]^T$ that is zero-mean having a covariance matrix C.

$$C = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

where $-1 \leq \rho \leq 1$ is the correlation coefficient between $w[0]$ and $w[1]$.

1.1 Finding the efficient estimator for A using the linear model

Using the linear model, and efficient estimator of A could be found. Now, the sample could be represented as a vector X as follows

$$X = \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} = \begin{bmatrix} 1.A + w[0] \\ 1.A + w[1] \end{bmatrix} = \begin{bmatrix} 1.A \\ 1.A \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} A \\ A \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} A + \begin{bmatrix} w[0] \\ w[1] \end{bmatrix} = HA + W$$

where H is the transformation matrix defined as

$$H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Thus, The MUVE of the problem could be expressed as

$$\begin{cases} \hat{A}_{MVUE} = \left(H^{-1}C^{-1}H\right)^{-1} H^T C^{-1} X \\ C_{\hat{A}_{MVUE}} = \left(H^{-1}C^{-1}H\right)^{-1} \end{cases}$$

1.2 Implementing the estimators

The code below implements a Monte-Carlo simulation program for the problem using different values of ρ. For $\hat{A}_{MVUE}$ to exist, the matrix C should be invertible (easy to see from the previous expression) and it also should be positive definite. For the given values of $\rho = \{-1, 0, 0.5, 1\}$, the values $\rho = -1$ and $\rho = -1$ make C a non-invertible matrix. Thus, other estimators should be used for these special values. Estimator $\hat{B} = \frac{x[0]+x[1]}{2}$ representing the samples mean would be used for these values.

The code generates a random noise with the specified covariance matrix each time. Then, the code check which estimator to use $\hat{A}_{MVUE}$ or $\hat{B}$ depending on the existent of the inverse of the covariance matrix C, after that it calculates the statistics of the estimator and saves the results.

The estimator is unbiased as can be seen by the simulated mean values presented in Table 2. The variances of the estimator fro different values of ρ as presented in Table 1 and depending on the value of ρ the estimator could be efficient to some extent (it depends on the threshold we set for efficiency). These results are depicted in the plot of Figrue 1. One remarkable note in Figrue 1, for $\rho = -1$, the estimator has a zero variance, and it reduces to pulse $\delta(A)$. Reflecting back to the expression of the estimator in this case $\hat{B} = \frac{x[0]+x[1]}{2}$, we clearly see that this result is expected as summing two fully uncorrelated noise elements $(\rho = -1)$ will cancel them, and thus we will end up with only the signal of interest without variations.

```
close all ; clear ; clc ;
rng(0) ; % reset the random number generator (for reproducibility)

A = 2 ;
sigma_squared = 0.5 ;
MC = 100000 ; % number of Monte Carlo loops

rho = [-1, 0, 0.5, 1] ; % the values of rho to test for

N = 2 ; % Number of samples in the data
n = [0:N-1] ; % row vector to represent the index of each sample
H =  ones(N,1) ; % transformation matrix as derived in part 1.a
```

```
% intializing the results vectors
estimate_simulated = NaN*ones(size(rho)) ;
var_estimate_simulated = NaN*ones(size(rho)) ;
var_estimate_theory = NaN*ones(size(rho)) ;
estimate = cell(size(rho)) ;

for ii = 1:length(rho)
    %     all_C{ii} = sigma_squared.*[ 1 rho(ii) ; rho(ii) 1 ] ; % used to store C
    %     all_inv_C{ii} = inv(all_C{ii}) ; % used to store inverse of C

    C = sigma_squared.*[ 1 rho(ii) ; rho(ii) 1 ] ; % generate the covariance matrix

    noise = zeros(MC,N) + randn(MC,N)*chol(C) ; % N cols of GN zero mean and C cov mat repeated MC rows
    %     cov_noise{ii} = cov(noise) ; % to check the cov mat of the noise (yes, it matches C)

    signal = H*A ; % H*A ==> linear model of the signal of interest
    X = signal + noise' ; % captured samples with MC times in columns and has N=2 samples in rows

    if det(C) ~= 0
        estimate{ii} = inv(H.'*inv(C)*H)*H.'*inv(C)*X ;  %  it gives 1*MC estimated values
        estimate_simulated(ii) = mean(estimate{ii}) ; % mean of the estimate along MC loops
        var_estimate_simulated(ii) = var(estimate{ii}) ;  % variance of the estimate along MC loops
        var_estimate_theory(ii) = inv(H.'*inv(C)*H) ; % from part 1.a
    else
        estimate{ii} = mean(X) ;  %  (mean along rows == along samples) it gives 1*MC estimated values
        estimate_simulated(ii) = mean(estimate{ii}) ; % mean of the estimate along MC loops
        var_estimate_simulated(ii) = var(estimate{ii}) ;  % variance of the estimate along MC loops
        var_estimate_theory(ii) = nan ;
    end
end
results_est_mean = [estimate_simulated]
```

```
    results_est_mean = 1x4
        2.0000    1.9998    2.0017    2.0031
```

```
results_est_var = [var_estimate_simulated ; var_estimate_theory]
```

```
    results_est_var = 2x4
        0.0000    0.2505    0.3738    0.5001
            NaN    0.2500    0.3750       NaN
```

```
% Plotting thing
f1 = figure ;
for ii = 1:length(rho)
    histogram(estimate{ii},'Normalization','probability','DisplayStyle',"stairs",'LineWidth',1.2) ;
    hold on
    myLegend{ii} = strcat('$\rho =$', num2str(rho(ii))) ;
end
legend(myLegend,'interpreter','latex','Location','best') ;
grid minor
xlabel('$\hat{A}$','interpreter','latex') ;
ylabel('PDF','interpreter','latex') ;
```

Table 1: Simulated and theoretical variances of the estimator

ρ	-1	0	0.5	1
Simulated variance	0.0000	0.2505	0.3738	0.5001
Theoretical variance	NaN	0.2500	0.3750	NaN

Table 2: Simulated and theoretical means of the estimator

ρ	-1	0	0.5	1
Simulated mean	2.0000	1.9998	2.0017	2.0031
Theoretical mean = A_{true}	2			

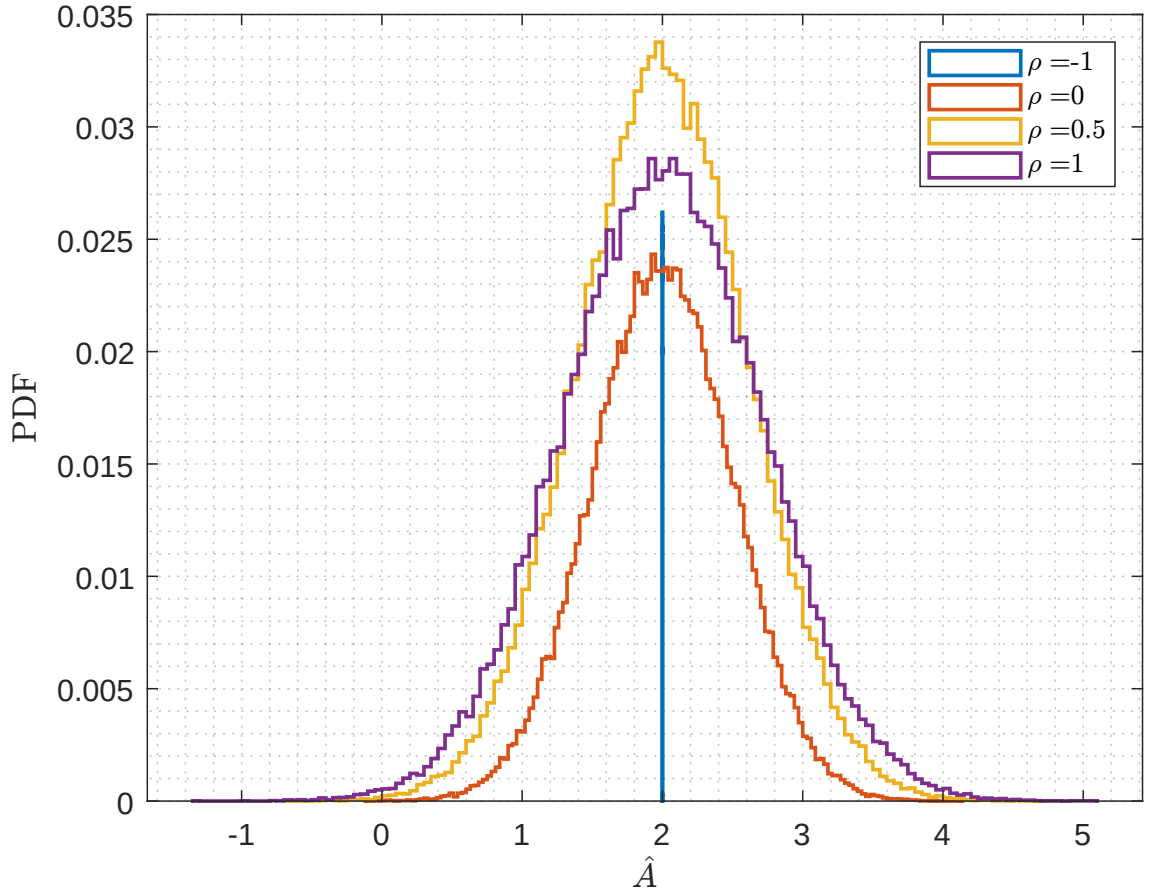

Figure 1: Theoretical PDFs of the estimator of A for different values of ρ

2. MLE for uniform distribution

2.1 Finding the maximum likelihood estimator of θ

The data represents an N IID elements uniformly distributed on the range $[0, \theta]$ (i.e $x[n] \sim U[0, \theta]$). In the code below, the first lines are used to generate X (the random number generator has been rested for reproducibility).

The pdf of $x[n]$ is

$$p(x[n];\theta) = \begin{cases} \frac{1}{\theta} & \text{for } 0 \leq x[n] \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

and since the elements are I.I.D, the pdf (likelihood function) of X is

$$LF(X) = p(X;\theta) = \prod_{n=1}^{N} p(x[n];\theta) = \begin{cases} \left(\frac{1}{\theta}\right)^N = \frac{1}{\theta^N} & \text{for } 0 \leq x[n] \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

The log likelihood function of X is

$$LLF(X) = \ln\{p(X;\theta)\} = \ln\left\{\frac{1}{\theta^N}\right\} = -N\ln\{\theta\}$$

The derivative of the log likelihood function of X is

$$\frac{d}{d\theta}\{LLF(X)\} = \frac{d}{d\theta}\{\ln\{p(X;\theta)\}\} = \frac{d}{d\theta}\{-N\ln\{\theta\}\} = -\frac{N}{\theta}$$

The derivative of the log likelihood function is minimized when the θ is maximized and $theta$ is estimated on the basis of the observation, hence it can only take values from the observations $x[n]$. Therefore, the MLE of θ is

$$\hat{\theta}_{MLE} = max\,(x[n]) = max\,(x[1], x[2], ..., x[N])$$

2.2 Implementing the estimators $\hat{\theta}_{MLE}$ and $\hat{A}$

The MLE $\hat{\theta}_{MLE}$ estimator represents the maximum value of the samples. The estimator $\hat{A}$ represents twice the samples mean. The code for the Monte-Carlo simulation of both estimators is shown bellow, where the *for* loop approach was replaced by a vectorized code.

```
close all ; clear ; clc ;
rng ; % reset the random number generator (so that we get the same results everytime)

MC = 100000 ; % number of Monte Carlo loops
theta = 1 ;
N = 100 ;

X = theta.*rand(N,MC) ; % N rows of uniform random elements repeated MC columns
% size(X)

theta_ML = max(X) ;  %  (from part 1.a) it gives MC*1 estimated values (max of X along the rows)
% size(theta_ML)

A_est_mean = 2*mean(X) ; % it gives MC*1 est_A values (2*sampleMean of X along the rows)
% size(A_est_mean)
```

2.3 Checking whether the estimators are biased

Using the code below, the bias of both estimator can be checked by using their mean ($E\{\hat{\theta}_{MLE}\}, E\{\hat{A}\}$) to obtain the absolute percentage error with respect to the true value of θ. In the code below, the error percentage was less than 1% and hence, both estimators are unbiased.

```
% bias
if abs(mean(theta_ML) - theta)/abs(theta) < 0.01
    disp('ML estimator is unbiased')
else
    disp('ML estimator is biased')
end
```

```
ML estimator is unbiased
```

```
if abs(mean(A_est_mean) - theta)/abs(theta) < 0.01
    disp('A_est_mean is unbiased')
else
    disp('A_est_mean is biased')
end
```

```
A_est_mean is unbiased
```

2.4 Finding the theoretical PDFs and generating the comparison plot

Selecting the maximum of a sampled data is selecting setting a statics order. Provided that the data is orders from the smallest to the largest element $X_{(i)}$ the ith order statics of the random data $\{X_1, X_2, ...X_N\}$. Since the data is ordered, the probability of selecting the max (.i.e X_N) is the same as the probability of the last statics order (.i.e $X_{(N)}$). From the provided reference, the kth order statics follows a beta distribution as $X_{(k)} \sim \beta(k, N+1-k)$ and thus, the Nth order statistics and the estimator $\hat{\theta}_{MLE}$ follows also a beta distribution .i.e $X_{(N)} \sim \beta(N, 1)$ and $\hat{\theta}_{MLE} \sim \beta(N, 1)$.

Approximating the uniform distribution to a Gaussian distribution done setting the variance and mean of the new approximation to the variance and mean of the original variance and mean of the uniform distribution. This means that the uniform distribution $U[0, \theta]$ is approximated by a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ having $\mu = \frac{\theta}{2}$ and $\sigma^2 = var\{U(0, \theta)\} = \frac{\theta^2}{12}$. Since the approximated distributions Gaussian, its mean could be approximated by the samples mean .i.e $\hat{B} = \frac{1}{N}\sum_{n=1}^{N} x[n]$. The MVUE(and MLE) estimator for the approximated distribution $\mathcal{N}(\mu, \frac{\sigma^2}{N})$ .i.e $\hat{B} \sim \mathcal{N}(\frac{\theta}{2}, \frac{\theta^2}{12N})$. Now, since $\hat{A}$ represent twice the sample mean, it could be written as $\hat{A} = \frac{2}{N}\sum_{n=1}^{N} x[n] = 2\hat{B}$ and thus $\hat{A} \sim \mathcal{N}(\mu_{\hat{A}}, \sigma^2_{\hat{A}})$ (because there's an affine transformation between $\hat{A}$ and $\hat{B}$ the latter follows a Gaussian distribution) where $\mu_{\hat{A}} = 2\mu_{\hat{B}} = \theta$ and $\sigma^2_{\hat{A}} = (2)^2\sigma^2_{\hat{B}} = \frac{\theta^2}{3N}$ .i.e $\hat{A} \sim \mathcal{N}(\theta, \frac{\theta^2}{3N})$.

The code below generates the pdf of each estimator and also generates Figure 2, showing the theoretical and simulated pdfs of both estimator. A normalization to range needed to be made in order to match the amplitudes of the simulated and theoretical pdfs.

```
f2 = figure ;

H1 = histogram(theta_ML,'Normalization','probability','DisplayStyle',"stairs") ;
hold on
H2 = histogram(A_est_mean,'Normalization','probability','DisplayStyle',"stairs") ;
hold on

X2 = linspace(H2.BinLimits(1),H2.BinLimits(2),100) ; % range of x for theoretical results

ML_theor_PDF = betapdf(X2,N,1) ;
ML_theor_PDF_normalized = normalize(ML_theor_PDF, ...
    'range',[min(H1.Values(:)) max(H1.Values(:))]) ;
plot(X2, ML_theor_PDF_normalized,'g--','Linewidth',1) ;


A_est_mean_theor_PDF = normpdf(X2,theta,sqrt((theta)^2/(3*N))) ;
A_est_mean_theor_PDF_normalized = normalize(A_est_mean_theor_PDF, ...
    'range',[min(H2.Values(:)) max(H2.Values(:))]) ;
plot(X2, A_est_mean_theor_PDF_normalized,'k--','Linewidth',1) ;
```

```
grid minor
legend('Simulated $\hat{\theta}_{ML}$','Simulated $\hat{A}$','Theoretical $\hat{\theta}_{ML}$', ...
    'Theoretical $\hat{A}$ ','interpreter','latex','Location','best') ;
xlabel('$\hat{\theta}$','interpreter','latex') ;
ylabel('PDF','interpreter','latex') ;
```

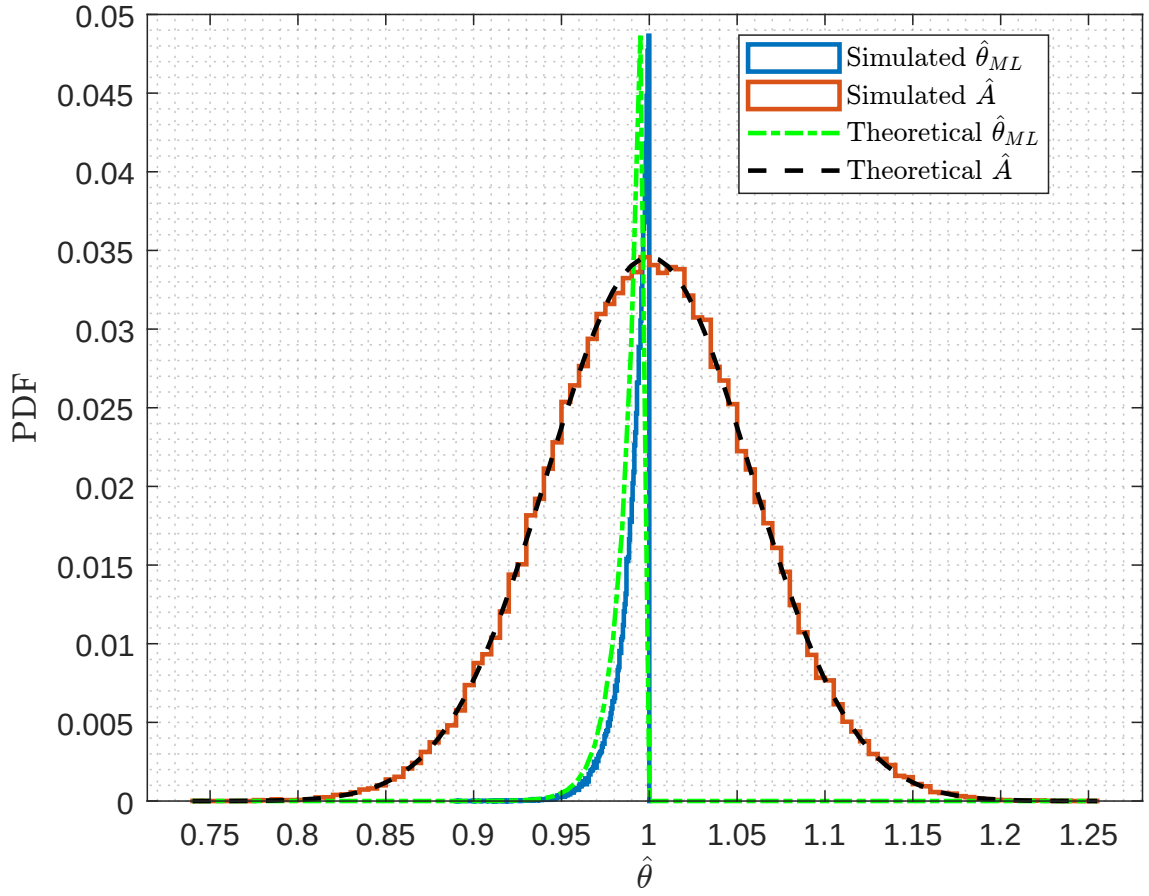

Figure 2: The simulated and theoretical PDFs of the estimators $\hat{\theta}_{ML}$ and $\hat{A}$