
```
%Project 2
%Do Hyung (Dave), Brian, Minh
```

P1

Bayes MMSE estimator

```
% Number of trials
N= 10000;

% Generating Y and W.
% Y is a uniform distribution from -1 to 1.
% W is a uniform distribution from -2 to 2.
Y = (2*rand(N,1)-1);
W = (4*rand(N,1)-2);

% Generating X = Y + W
X = Y + W;

% The Bayesian MMSE estimator is a piecewise function with the domains
% of X = [-3,-1), [-1,1), and [1,3]. To find these ranges we used
% logical
% indexing to find the indices where X falls into each of these ranges
x_ltneg1 = X<-1; %X = [-3,-1)
x_btwn = X <= 1 & X > -1; % X = [-1, 1)
x_else = ~(x_ltneg1 | x_btwn); % X = [1, 3]

% Preallocating the array for SE
SE = zeros(N,1);

% The piecewise function for each of the domains of X:
% The mean squared error is calculated by (Y-Y_hat)^2
% Y_hat isn't explicitly calculated
SE(x_ltneg1) = (Y(x_ltneg1)-(X(x_ltneg1)+1)/2).^2;
SE(x_btwn) = Y(x_btwn).^2;
SE(x_else) = (Y(x_else)-(X(x_else)-1)/2).^2;

% Defining the variable names for the table
var_names = {'Experimental_MSE', 'Theoretical_MSE'};

% Calculating the MSE from the SE of each of the trials
MSE_emp = mean(SE);
MSE_theo = .25;

% Bayes MMSE estimator in example 8.5 results
bayes_results_table = table(MSE_emp,MSE_theo , 'VariableNames',
    var_names)

% Linear MMSE estimator can be find from the mean of X and Y, variance
% of X
% and Y, and the covariance of X and Y.
```

```

x_mean = mean(X);
y_mean = mean(Y);
% w_var = var(W);
% Cxy = [[x_var c_xy ],
%       [c_xy y_var ]]
Cxy = cov(X,Y);
x_var = Cxy(1,1);
y_var = Cxy(2,2);
c_xy = Cxy(1,2);

% Applying LMMSE estimator formula to find LMMSE
y_hat = y_mean + (c_xy/(x_var))*(X-x_mean);

% Use the LMMSE estimator to find the MSE
MSE_emp = mean((Y-y_hat).^2);

% Theoretical MSE
MSE_theo = 4/15;

% Linear MMSE estimator in example 8.6 results
lin_mmse_results_table = table(MSE_emp,MSE_theo, 'VariableNames',
    var_names)

```

```
bayes_results_table =
```

```
1x2 table
```

<i>Experimental_MSE</i>	<i>Theoretical_MSE</i>
0.24719	0.25

```
lin_mmse_results_table =
```

```
1x2 table
```

<i>Experimental_MSE</i>	<i>Theoretical_MSE</i>
0.26229	0.26667

P2

```

clear all;
%defines the variances for the given Y and R
y_var = [1 5 1 5].';
r_var = [1 1 5 5].';

%Defines the number of observations
%By varying over the number of observations,

```

```

%we can see how the observations change the error of the estimator
N = 10000;

%defines the Y and R variables as the text does.
y_mean = 1;
size_y = size(y_var);
size_y = size_y(1);

%the number of noise signals seen
%similar to the number of observations, we will
%see how the number of noise signals changes the
%overall accuracy of our estimate
N_r = 20;

%Defining the messages in the legend
%By differentiating between empirical and theoretical
%estimates, we can see the relative accuracy between the two.
emp_message = "\\sigma^2_y=%d \\sigma^2_r=%d empirical";
theo_message = "\\sigma^2_y=%d \\sigma^2_r=%d theoretical";

figure; %Initializes the figure for plotting
legend_labels = [];

for i = 1:size_y
    legend_labels = [legend_labels sprintf(emp_message, y_var(i),
r_var(i))]; %uses the previous legend messages to create the legend
    legend_labels = [legend_labels sprintf(theo_message, y_var(i),
r_var(i))];
    MSE_emp = zeros(N_r,1); %Initializes the matrices for the MSE of
the theoretical and empirical distributions
    MSE_theo = zeros(N_r,1);
    for num_obs = 1:N_r %loops through the number of observed
signals. By changing the number of observed signals, we will see how
the number actually changes the accuracy of our estimator
        Y = repmat(sqrt(y_var(i)).*randn(N,1)+y_mean, 1, num_obs);
        %Creates the actual Y values from the mean and distributions defined
above
        R = sqrt(r_var(i)).*randn(N,num_obs); %Creates the R values
from the distributions above

        X = Y + R; %Defines the X as in the problem
        Cxx = cov(X); %Covariance
        a = (inv(Cxx))*(y_var(i)*ones(num_obs,1)); %element for the
LMMSE equation

        y_hat = y_mean; %With a lack of observation, the estimator is
the mean.
        for j = 1:num_obs
            y_hat = y_hat - a(j)*(y_mean-X(:, j)); %Changes the
estimator by making observations. We loop only through the number of
signals we are currently observing in the outer for loop.
        end
    end
end

```

```

        MSE_emp(num_obs) = mean((Y(:,1)-y_hat).^2, 'all');
        %calculates the empirical MSE. We take the square of the difference
        between the value and our estimate.
        MSE_theo(num_obs) = (y_var(i))*(r_var(i))/
(num_obs*y_var(i)+r_var(i)); %calculates the theoretical MSE from
textbook
    end

    %plotting of the different MSE vectors
    plot(1:N_r, MSE_emp);
    hold on;
    plot(1:N_r, MSE_theo);
    hold on;
    % var_names = {'N_r', 'Var_y', 'Var_r', 'Experimental_MMSE',
'Theoretical_MMSE'};
    % lin_mult_obs_results_table = table(ones(size_y,1)*N_r, y_var,
r_var, MMSE_emp, MMSE_theo, 'VariableNames', var_names)
end
legend(legend_labels)
title('MMSE vs Observations with different variances');

% Results of using the linear estimator for multiple noisy
observations

```

