Table of Contents

Stoch MMSE Project

Sophie Jaro and Rebecca Gartenberg March 12, 2020

```
% This project implements a Bayesian MMSE estimator and an MLE
 estimator.
% The examples and equations used in this project are based on the MIT
% notes titled "Chapter 8: Estimation with Minimum Mean Square Error"
% written by Oppenheim and Verhese.
% Scenario 1 implements examples 8.5 and 8.6.
% Scenario 2 implements example 8.8.

% Scenario 1: In scenario 1, a signal X is constructed from a noisy
% uniformly distrubuted signal, W, between -2 and 2, and a uniformly
% distributed signal, Y, between -1 and 1. The MMSE estimator is used
 to estimate Y
% based on the noisy signal X using equation 8.30 in the notes. The
 mean squared error
% is calculated and compared to the theoretical value of mean squared
% error. The same procedure was repeated for the LMMSE. The
% same signal X was used to estimate Y using the LMMSE estimator.
% The result was that the empirical was within 0.001 of the
% theoretical estimate for both the MMSE and the LMMSE.
% The theoretical and empirical estimates based on MMSE and LMMSE are
% reported in a table.

% Scenario 2: In scenario 2, the LMMSE was used to estimate Y based
% on several noisy signals. Y is a Gaussian signal of mean 1 and a
 specified variance.
% Several noise sources, R, are Gaussian with zero mean and specified
 variance.
% A noisy signal X is equal to Y + R. The LMMSE estimator is then used
 to estimate
% Y. The empirical and theoretical mean squared error is calculated
 based
% on the number of noise sources, which ranges from 1 to 100. This
 process
% is repeated for 10,000 trials, and the mean squared error for each
 number
% of noise sources is averaged.
% The empirical and theoretical mean squared error values decreased as
 a
% function of number of noise sources. This is plotted in figure 1 for
 4
```

```
% different combinations of variances of Y and R.

clc; close all; clear all;
```

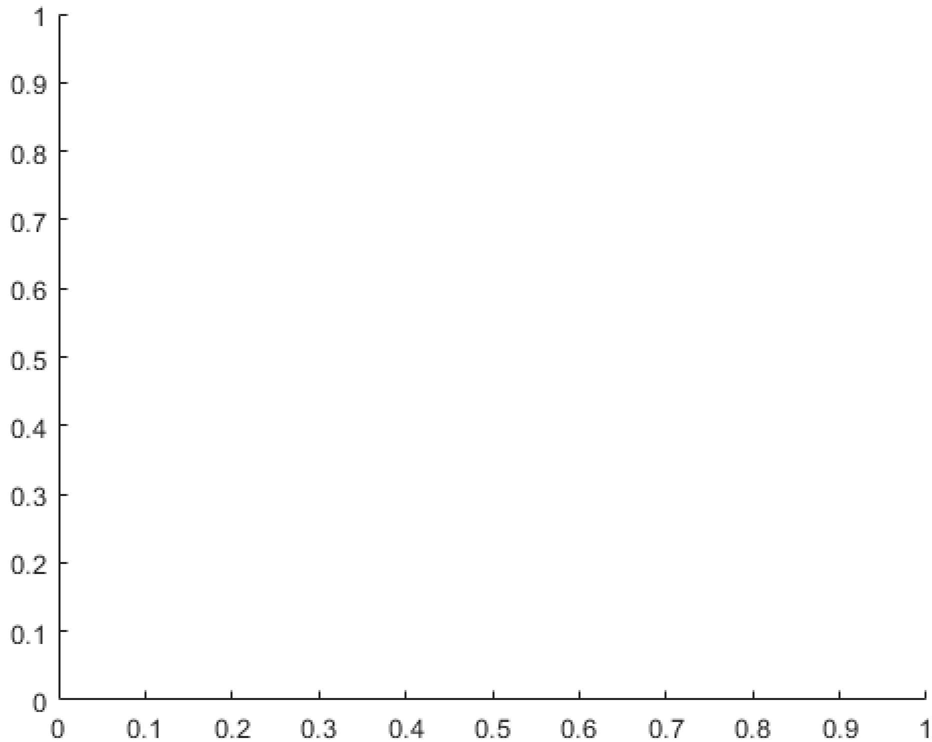
Scenario 1

```
N = 1e5; % number of trials

% Part a
Y = -1 + (1+1)*rand(1,N); % signal random uniform distribution [-1, 1]
W = -2 + (2+2)*rand(1,N); % noise random uniform distribution [-2, 2]

X = Y + W;

syms y_hat(x)
y_hat(x) = piecewise(-3<= x <-1, 1/2 + (1/2)*x, -1 <= x < 1, 0, 1 <= x
 <= 3,  -1/2 + (1/2)*x); % (8.30)

% converting symbolic function into a double
y_hat = double(y_hat(X)); % estimator

emp_MMSE = mean(((Y - y_hat).^2)); % empirical mean squared error of
 the estimator

% Part b

mu_X = mean(X);

var_Y = var(Y);
var_X = var(X);

cov_yx = cov(Y,X); % covariance matrix
cov_yx = cov_yx(1,2); % covariance value of x and y

ro_yx = cov_yx/sqrt((var_Y)*(var_X)); % correlation coefficient of x
 and y

% LMMSE estimator
Y_hatL = (1/5)*X; % linear estimator given in notes in example 8.6 on
 page 154
emp_LMMSE = mean((Y_hatL - Y).^2); % empirical mean squared eror of
 linear estimator

T = table([1/4 ; 4/15],[emp_MMSE ; emp_LMMSE],'VariableNames',...
    {'Theoretical', 'Empirical'}, 'RowNames', {'MMSE','LMMSE'});
disp(T)
```

	Theoretical	*Empirical*
MMSE	*0.25*	*0.25122*
LMMSE	*0.26667*	*0.26778*

Scenario 2

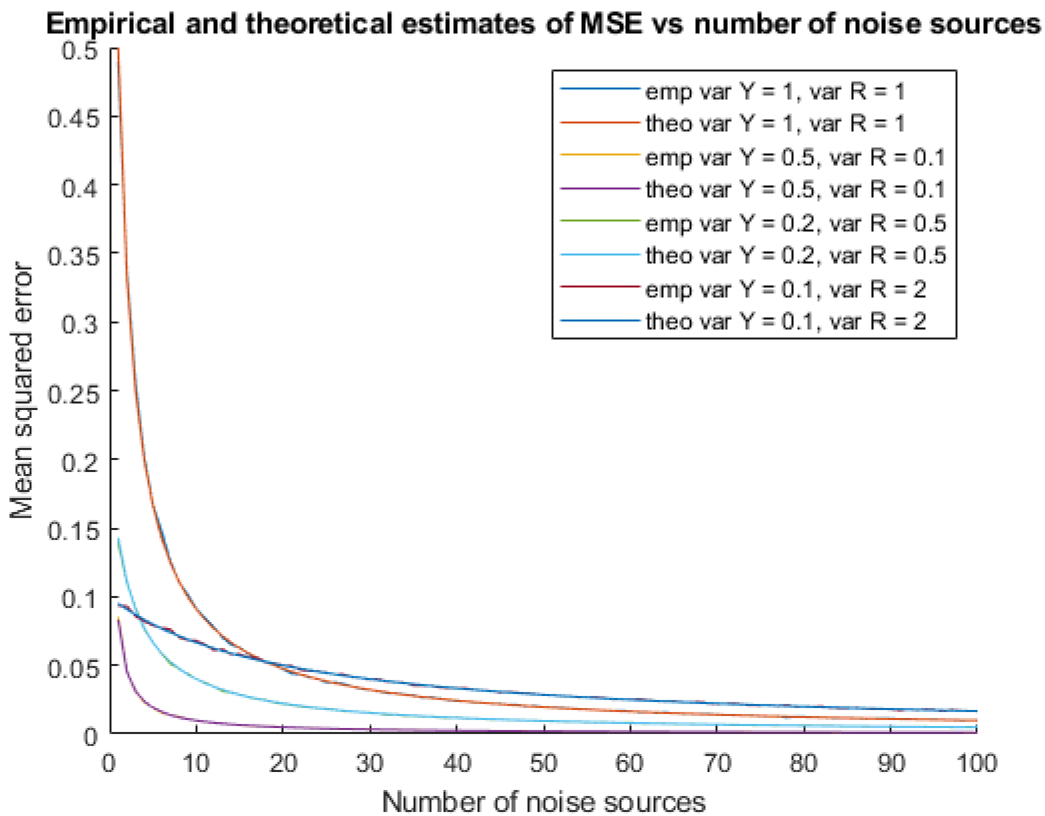
```
close all;
clc;

mu_Y = 1;
% four combinations of variances for signal Y and noise sources R
varvec_Y = [1, 0.5, 0.2, 0.1];
varvec_R = [1, 0.1, 0.5, 2];

sources = 100;
trials = 10000;

figure
hold on

% loop through combinatios of variances
for i = 1:length(varvec_Y)
    [MSE, theoretical] = noisy_estimate(mu_Y, varvec_Y(i),
 varvec_R(i), sources, trials);
    plot((1:sources), MSE)
    plot((1:sources), theoretical)
end

title('Empirical and theoretical estimates of MSE vs number of noise
 sources')
xlabel('Number of noise sources')
ylabel('Mean squared error')
legend('emp var Y = 1, var R = 1', 'theo var Y = 1, var R = 1', 'emp
 var Y = 0.5, var R = 0.1', 'theo var Y = 0.5, var R = 0.1',...
    'emp var Y = 0.2, var R = 0.5', 'theo var Y = 0.2, var R =
 0.5', 'emp var Y = 0.1, var R = 2', 'theo var Y = 0.1, var R = 2')
```



Function to compute MSE of noisy estimate

```
function [MSE, theoretical] = noisy_estimate(mu_Y, var_Y, var_R,
 sources, trials)

    % preallocating vectors to store info
    error = zeros(1, trials);
    theoretical = zeros(1, sources);
    MSE = zeros(1, sources);

    % computing standard deviation from variance
    sigma_Y = sqrt(var_Y);
    sigma_R = sqrt(var_R);

    for i = 1:sources
        for j = 1:trials

            Y = mu_Y + sigma_Y.*randn(1,1); % signal random gaussian
 distribution with mean (mu_Y = 1) and specified variance
            R = sigma_R*randn(1,i); % noise random gaussian
 distribution with mean 0 and specified variance
            % X is computed based on one value of Y and several noise
 sources, R
            X = repmat(Y, 1, i) + R; % (8.75)
```

```
            % Linear estimator
            Y_hatL = (var_R*mu_Y + var_Y*sum(X))/(i*var_Y+var_R); %
 (8.79)
            error(j) = mean((Y_hatL - Y).^2);

        end

        MSE(i) = mean(error); % mean squared error
        theoretical(i) = (var_Y*var_R)/((i*var_Y)+var_R); % (8.80)

    end
end
```



Published with MATLAB® R2019a