Andy Jeong
Professor Keene
ECE302 Stochastics
April 4, 2019

Stochastics – MMSE and ML Estimators: Derivations

Scenario 1.

$$\text{System: } \mathrm{x} = \mathrm{h}\Theta + \mathrm{v}, \quad \text{where } \mathrm{v} \sim \mathrm{N}(0, \sigma^2)\ [0 - mean\ AWGN] \text{and } \ \mathrm{h} = 0.5$$

a) Let $\Theta \sim \mathrm{N}(\mu, \sigma^2)$.

$$\widehat{\Theta}_{LMMSE} = \frac{Cov(\Theta, \mathrm{X})}{Var(X)}\big[X - E[X]\big] + E[\Theta]$$

$$= \frac{Cov(\Theta, \mathrm{X})}{Cov(X, X)}\big[X - E[X]\big] + E[\Theta]$$

$$= \frac{\mathrm{E}[\Theta \mathrm{X}] - E[\Theta]\mathrm{E}[\mathrm{X}]}{E[XX] - E[X]^2}\big[X - E[X]\big] + E[\Theta] \ ,$$

$$\text{where } \mathrm{E}[\mathrm{XX}] = \mathrm{E}[(\mathrm{h}\Theta + \mathrm{v})^2] = \ h^2E[\Theta^2] + E[\mathrm{v}^2],$$

$$E[X]^2 = \mathrm{E}[\mathrm{h}\Theta + \mathrm{v}]^2 = h^2E[\Theta^2],$$

$$E[\Theta \mathrm{X}] = \ E[(\Theta)(\mathrm{h}\Theta + \mathrm{v})],$$

$$\mathrm{E}[\Theta]\mathrm{E}[\mathrm{X}] = \mathrm{E}[\Theta]\mathrm{E}[\mathrm{h}\Theta + \mathrm{v}],$$

$$\mathrm{E}[\mathrm{X}] = E[h\Theta + \mathrm{v}] = hE[\Theta] + \mathrm{E}[\mathrm{v}] = \mathrm{hE}[\Theta].$$

$$\therefore \ \widehat{\Theta}_{LMMSE} = \frac{E[(\Theta)(\mathrm{h}\Theta + \mathrm{v})] - \mathrm{E}[\Theta]\mathrm{E}[\mathrm{h}\Theta + \mathrm{v}]}{h^2E[\Theta^2] + E[\mathrm{v}^2] - h^2E[\Theta^2]}\big[(\mathrm{h}\Theta + \mathrm{v}) - hE[\Theta]\big] + E[\Theta]$$

$$= \frac{hE[\Theta^2] + E[\Theta]\mathrm{E}[\mathrm{v}] - hE[\Theta]^2 + E[\Theta]\mathrm{E}[\mathrm{v}]}{h^2(E[\Theta^2] - E[\Theta]^2) + Var(v)}\big[(\mathrm{h}\Theta + \mathrm{v}) - hE[\Theta]\big] + E[\Theta]$$

$$= \frac{hE[\Theta^2] - hE[\Theta]^2}{h^2(E[\Theta^2] - E[\Theta]^2) + Var(v)}\big[(\mathrm{h}\Theta + \mathrm{v}) - hE[\Theta]\big] + E[\Theta]$$

$$= \frac{hVar(\Theta)}{h^2Var(\Theta) + Var(v)}\big[(\mathrm{h}\Theta + \mathrm{v}) - E[\Theta]\big] + E[\Theta]$$

b) Let Θ: deterministic.

$$E[\mathrm{X}|\Theta] = hE[\Theta] + E[v] = h\Theta + 0 = \mathrm{h}\Theta$$

$$\widehat{\Theta}_{ML} = \frac{E[X|\Theta]}{h}$$

d) Running the Bayes MMSE estimator with incorrect priors estimated as Gamma, Exponential, and Uniform distributions showed an interesting result:

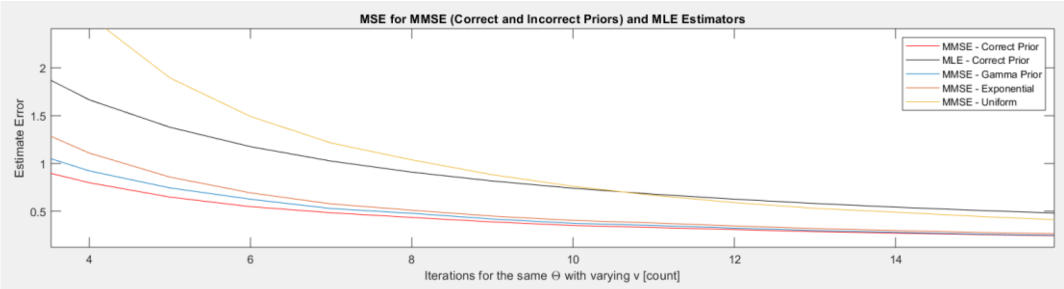


Figure 1. 1 - (d) Bayes MMSE Estimator with Incorrect Priors

All 3 incorrect prior MMSE estimators matched ML Estimator by around 10th iteration with the same theta and varying noise v because with increased iterations the variations are averaged and thus error decreases.

Scenario 2.

a) Mathematical Model
- Suppose a system with a BPSK signal, a BPSK interference signal, and AWGN, such that Y = X + c · Φ + Z,
where X = original signal, c = SNR, Φ = interference, Z = noise

For time range [1, n] = [1, t1], [t1+1, t2], [t2+1, n],

Symbol outcomes with probabilities (assuming equiprobability):

1. Without interference signal → +/- 1: p = ½
2. With interference signal → +/- 2: p = ¼, → 0: p = ½

$$f_{\bar{x}}(\bar{x}; t1, t2) = \prod_{i=1}^{t1} \frac{1}{2} [\, N(x_i\,;\, -1, \sigma^2) + N(x_i;\, -1, \sigma^2)\,]$$

$$\cdot \prod_{i=t1+1}^{t2} [\, \frac{1}{4} N(x_i\,;\, -2, \sigma^2) + \frac{1}{2} N(x_i;\, 0, \sigma^2)\; + \frac{1}{4} N(x_i;\, 2, \sigma^2)\,]$$

$$\cdot \prod_{i=t2+1}^{n} \frac{1}{2} [\, N(x_i\,;\, -1, \sigma^2) + N(x_i;\, -1, \sigma^2)\,]$$

** Notation Remark*

$(i)\; N(x_i;\, \mu, \sigma^2) = Gaussian\ Distribution\ of\ x_i\ with\ mean\ \mu\ and\ variance\ \sigma^2$

References

1. Keene, S., & Carruthers, J. B. (2009). Collision and Fade Localization within Packets for Wireless LANs. Wireless Personal Communications, 55(3), 379-394. doi:10.1007/s11277-009-9805-1

Scenario 3.

1) Procedure
    a. Load data from provided “data.mat”.
    b. Determine multiple iterations of ML estimates of each expected distributions (Exponential and Rayleigh).
    c. Plot QQ-plots (Quantile-Quantile) of estimates and data to see if which distribution estimates follow the data distributions better (if same, or very similar, distribution, the data points will lie on a straight line).
    d. Plot histograms of both estimates from both distributions and data, to see which overlap the most and more closely.

Contents


```
% ECE302 - Stochastic Processes
% Andy Jeong / Spring 2019
% Project 2: Estimation Techniques: MMSE, MLE
clear all; close all; clc;
```

Scenario 1

System: x = h θ + v; Conditions: (1) v is Gaussian Random Variable with 0 - μ, variance σ^2; (2) h = 0.5 Refer to separate derivations page

```
% theta and v have unique Gaussian distribution with each mean and
% variance (thus are independent). For a system of such joint Gaussian
% Random Variables, MMSE could be estimated using Linear MMSE Estimator.

% Initial settings for h, mean/variance of v
h = 0.5;
mu_v = 0;
variance_v = 1;

% Iterating for 1000 different values of v, 100 trials of different theta
% and later take mean for each v (same theta) across each row
rows = 1000;  % iterations for different theta values
columns = 100; % iterations for different v (noise) values

% v is given as a Gaussian Random Variable for both 1-(a) and 1-(b)
v = normrnd(mu_v, sqrt(variance_v), rows, columns);
```

1 - (a)

θ is Gaussian Random Variable with mean μ and variance $\sigma_o{}^2$ -- Determine the Bayes MMSE estimate of θ

```
% set the mean and variance of theta random variable
mu_theta = 5;
variance_theta = 3;
theta = repmat( normrnd(mu_theta, sqrt(variance_theta), [rows, 1]), [1, columns] );
h = 0.5;
x =  h*theta + v;
% theta is same across row; take average of x across row for each index
% cumulative sum is for taking average of x
x = cumsum(x,2)./repmat(1:columns,[rows,1]);
```

```
theta_mmse = mu_theta + repmat(h*variance_theta ./ (h^2*variance_theta + variance_v./(1:co
lumns)),[rows,1]).*(x-h*mu_theta);
error_mmse = mean((theta-theta_mmse).^2);
```

1 - (b)

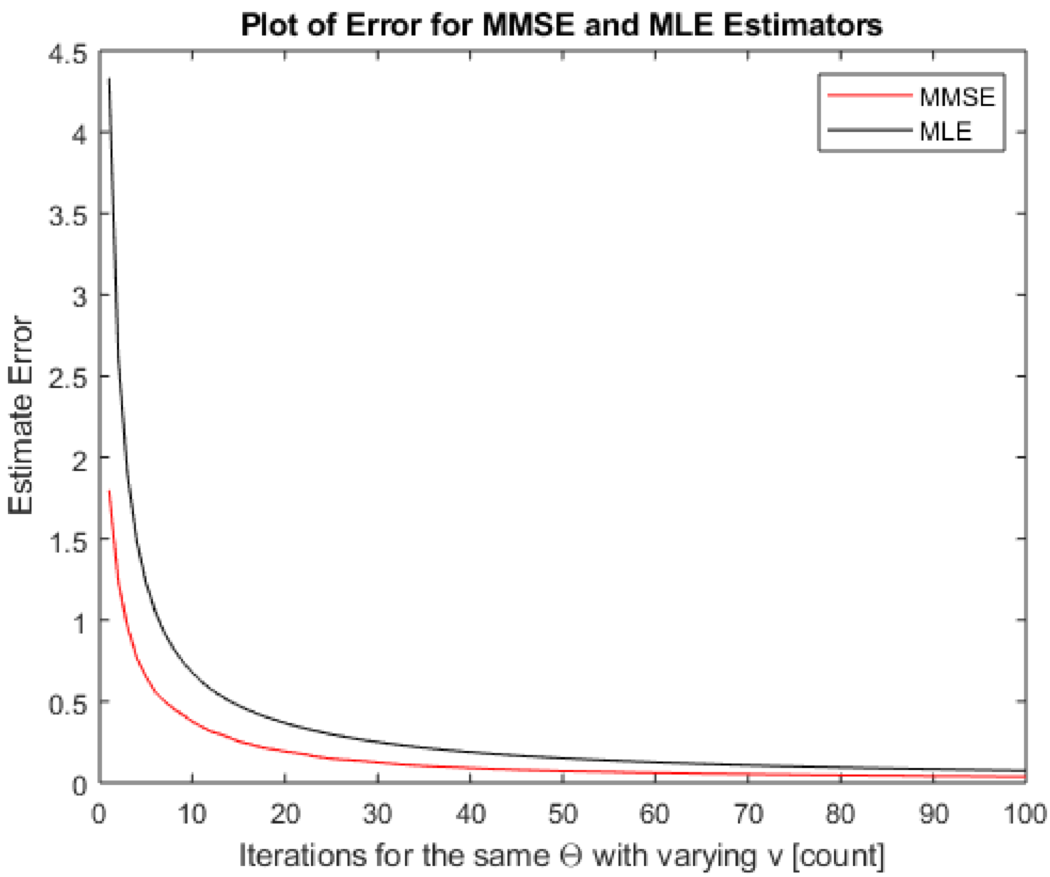
θ is deterministic,

```
% for maximum likelihood estimate of x, take derivate and set it equal to 0,
% which in this case would simply be
% MLE at nth measurement = mean of first n measurements (cumulative) / h
theta_mle = cumsum(x/h,2)./repmat(1:columns,[rows,1]);
error_mle = mean((theta-theta_mle).^2);
```

1 - (c)

The error for MMSE estimator is lower than that for MLE estimator. This implies that MMSE estimator performs better in low SNR.

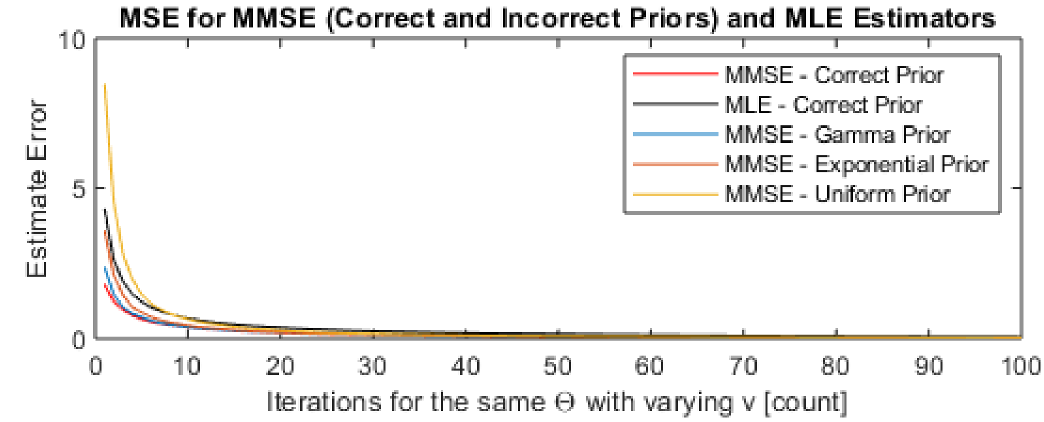
```
figure;
plot(1:columns,error_mmse,'r',1:columns,error_mle,'k');
title('Plot of Error for MMSE and MLE Estimators');
xlabel('Iterations for the same \Theta with varying v [count]')
ylabel('Estimate Error')
legend('MMSE','MLE')
```

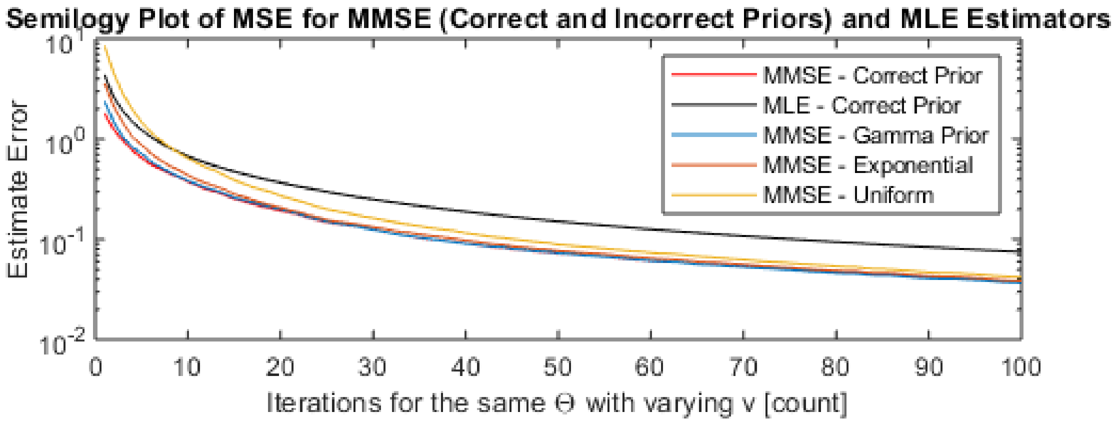


1 - (d)

Gamma Distribution: mean = k${\theta}$; k arbitrarily chosen since mean is what is of interest

```
theta_gamma = repmat(gamrnd(mu_theta,1,[rows,1]),[1,columns]);
x =  h*theta_gamma + v;
x = cumsum(x,2)./repmat(1:columns,[rows,1]);
theta_gamma_mmse = mu_theta + repmat(h*variance_theta ./ (h^2*variance_theta + variance_v.
/(1:columns)),[rows,1]).*(x-h*mu_theta);
error_mmse_gamma = mean((theta_gamma-theta_gamma_mmse).^2);

% Exponential Distribution
theta_exponential = repmat(exprnd(mu_theta,[rows,1]),[1,columns]);
x =  h*theta_exponential + v;
x = cumsum(x,2)./repmat(1:columns,[rows,1]);
theta_exponential_mmse = mu_theta + repmat(h*variance_theta ./ (h^2*variance_theta + varia
nce_v./(1:columns)),[rows,1]).*(x-h*mu_theta);
error_mmse_exponential = mean((theta_exponential-theta_exponential_mmse).^2);

% Uniform Distribution
theta_uniform = repmat(2*mu_theta*rand([rows,1]),[1,columns]);
x =  h*theta_uniform + v;
x = cumsum(x,2)./repmat(1:columns,[rows,1]);
theta_uniform_mmse = mu_theta + repmat(h*variance_theta ./ (h^2*variance_theta + variance_
v./(1:columns)),[rows,1]).*(x-h*mu_theta);
error_mmse_uniform = mean((theta_uniform-theta_uniform_mmse).^2);

figure;
% Plot errors for mmse and mle for correct prior and mmse for incorrect priors
% ** correct: normal distribution, incorrect: gamma, exponential, uniform

% Notes:
% MMSE estimator error gives lower MSEs relative to ML estimator error
% because the former is less sensitive at low SNR values, so it can perform better
% By ~10th iteration, MSE for MMSE estimators with incorrect priors fall
% below MLE error, and as more iterations are taken,
% these all converge to the MSE of MMSE estimator with correct prior
subplot(211);
plot(1:columns,error_mmse,'r',1:columns,error_mle,'k', ...
    1:columns,error_mmse_gamma,1:columns,error_mmse_uniform,1:columns,error_mmse_exponenti
al);
title('MSE for MMSE (Correct and Incorrect Priors) and MLE Estimators');
xlabel('Iterations for the same \Theta with varying v [count]')
ylabel('Estimate Error')
legend('MMSE - Correct Prior','MLE - Correct Prior','MMSE - Gamma Prior','MMSE - Exponenti
al Prior','MMSE - Uniform Prior')

% same data plotted in semilogy plot
subplot(212);
semilogy(1:columns,error_mmse,'r',1:columns,error_mle,'k', ...
    1:columns,error_mmse_gamma,1:columns,error_mmse_uniform,1:columns,error_mmse_exponenti
al);
title('Semilogy Plot of MSE for MMSE (Correct and Incorrect Priors) and MLE Estimators');
xlabel('Iterations for the same \Theta with varying v [count]')
ylabel('Estimate Error')
legend('MMSE - Correct Prior','MLE - Correct Prior','MMSE - Gamma Prior','MMSE - Exponenti
al','MMSE - Uniform')
```





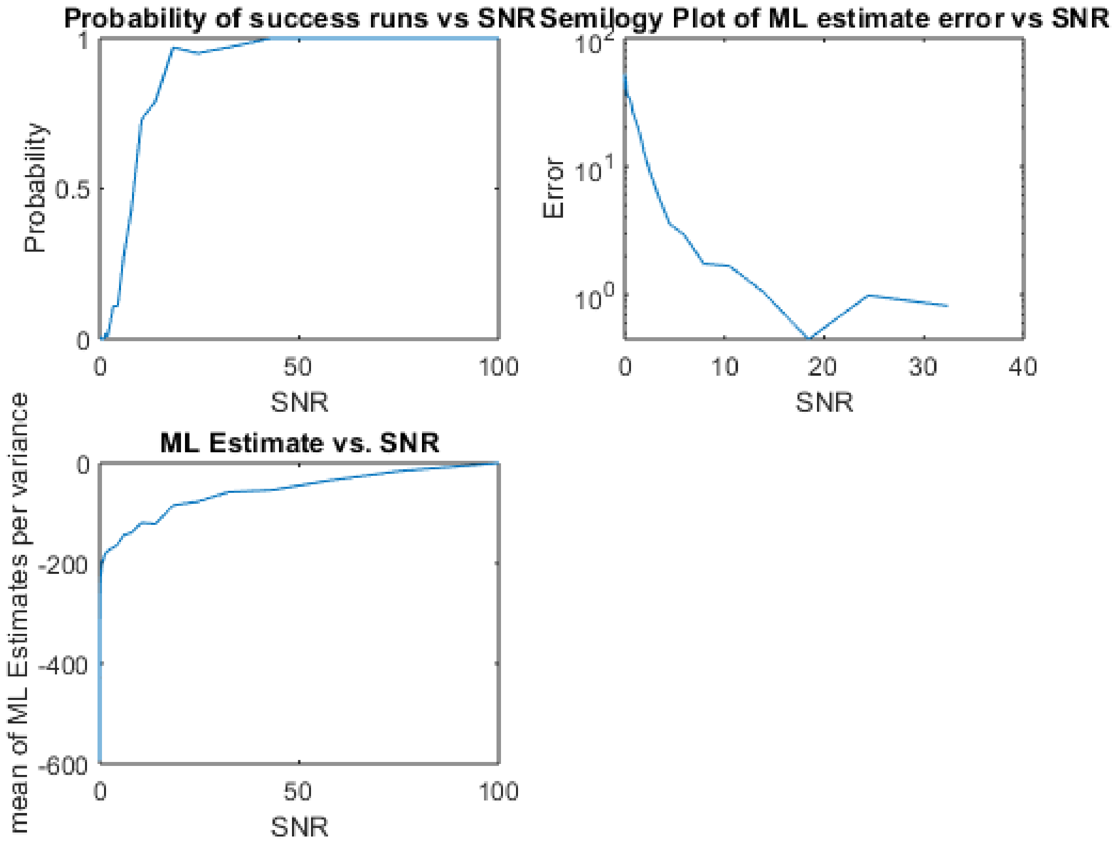
Scenario 2

```
% Y = X + c*phi + Z
% where the total resulting signal 'Y' is the sum of the original BPSK signal 'X',
% the interference BPSK signal 'phi' with SNR or SIR 'c', and some AWGN 'Z'
% Refer to separate derivations page for more information on the model.

% Define probability density functions to pass into likelihood() function
% These define pdf's for the three interval bounds set by the t1 and t2, which determine
% the duration of the interference signal to the original signal (both BPSK)
% 1 BPSK result: +/-1 (each probability = 1/2)
pdf1 = @(x_n, variance) 1/2 * ( normpdf(x_n, -1, sqrt(variance)) + normpdf(x_n, 1, sqrt(va
riance)) );
% 1 BPSK + 1 interfering BPSK result: +/-2 (each probability = 1/4), 0 (probability = 1/2)
pdf2 = @(x_n, variance) 1/4 * normpdf(x_n, -2, sqrt(variance)) + 1/2 * normpdf(x_n, 0, sqr
t(variance)) + 1/4 * normpdf(x_n, 2, sqrt(variance));

% For computational simplicity, experiment with total of 100 symbols
% and interference starting at 10th and ending at 80th symbol
numSymbols = 100;
startInterference = 10;
endInterference = 80;

% Set a range of variances during which the transmission estimates will be measured by ML
estimator
varianceSymbols = logspace(-2, 4, 50);

% Initialize vectors of interests (to be plotted)
probabilities = zeros( [1, length(varianceSymbols)] );
errors = zeros( [1, length(varianceSymbols)] );
maxEstimates = zeros( [1, length(varianceSymbols)] );

maxEstimate = 0;
```

```
minEstimate = 0;
tic;
% loop through the joint signals for each variance
for i = 1:length(varianceSymbols)
    variance = varianceSymbols(i);
    successCount = 0;
    difference = 0;
    % loop through each symbol
    for j = 1:numSymbols
        % generate two distinct symbols from a uniform discrete distribution
        % and assign them +/- 1 symbol
        signal = randi( 2, [1,numSymbols] );
        signal(signal == 2) = -1;

        % generate another BPSK signal (from uniform discrete distribution as well),
        % which will be intereference of the same length as the original signal,
        % but with +/- 1 for the specified duration of interference (collision)
        durationInterference = endInterference - startInterference + 1;
        interferenceBPSK = randi(2, [1, durationInterference]);
        interferenceBPSK(interferenceBPSK == 2) = -1;
        interference = zeros( [1,numSymbols] );
        interference(startInterference:endInterference) = interferenceBPSK;

        % Generate AWGN that lasts for the entire transmission period
        % the distribution is from Gaussian(Normal)
        noise = randn( [1,numSymbols] ) * sqrt(variance);
        receivedSignal = signal + interference + noise;

        % Assumption insights
        % Source:  Collision and Fade Localization within Packets for Wireless LANs
        % Authors: Sam Keene · Jeffrey B. Carruthers
        % Published online: 19 August 2009
        % Reference: Wireless Pers Commun (2010) 55:379?394, DOI 10.1007/s11277-009-9805-1
        %
        % Each packet is assumed to have a minimum collision size of 20
        % samples so that collisions of such short durations
        % that could potentially be indistinguishable from noise could be
        % avoided(discarded), provided that typical CTS/ACK packets tend to be of 38 bytes
 in length.

        % since interference BPSK signal could range from 10 to 80 symbols,
        % loop through the largest possible length of the original signal
        minLengthInterference = 10;
        maxEstimate = -1e6;
        maxTime1forSymbol = 0;
        maxTime2forSymbol = 0;
        % loop through period through which the interference signal lives
        % adjust +1 because inclusive range
        for time1 = 1:numSymbols-minLengthInterference+1
            minEndInterference = time1+(minLengthInterference-1);
            maxEndInterference = numSymbols;
                        % loop through ending times of the interference signal
            for time2 = minEndInterference:maxEndInterference
                % if the length of interference signal is greater than 80 (as specified in
itially),
                % stop iterations
                if time2-time1 > 80
                    break
                end
                % for the length of interference signal
                estimate = likelihood(receivedSignal, variance, time1, time2, pdf1, pdf2);
```

```
                % update estimate threshold to maximize
                if estimate > maxEstimate
                    maxEstimate = estimate;
                    maxEstimates(i) = estimate;
                    maxTime1forSymbol = time1;
                    maxTime2forSymbol = time2;
                end
            end % end time2
        end % end time1

        % successful transmission of the symbol if maximum t1 = 10, maximum t2 = 80
        if(maxTime1forSymbol == startInterference) && (maxTime2forSymbol == endInterferenc
e)
            successCount = successCount + 1;
        else
                        % add this total differnce in time length to the error vector
            difference = abs(maxTime1forSymbol - startInterference) + abs(maxTime2forSymbo
l - endInterference);
        end
        errors(i) = errors(i) + difference;
    end % end j
    errors(i) = errors(i)/numSymbols;
    probabilities(i) = successCount/numSymbols;
end % end i
toc
SNR = 1./varianceSymbols;

figure;
% Plot Success Probabilities versus SNR
% As SNR is increased, the probability that the interference will
% lie in the specified time interval increases
subplot(221);
plot(SNR, probabilities);
xlabel('SNR');
ylabel('Probability');
title('Probability of success runs vs SNR');

% Plot Logarithm-Scale Errors versus SNR
% Since the success probability is increased as SNR increases,
% the error, on the other hand, decreases
subplot(222);
semilogy(SNR, errors);
xlabel('SNR');
ylabel('Error');
title('Semilogy Plot of ML estimate error vs SNR');

% Plot ML Estimates versus SNR
% As SNR increases, the ML approximates the log of the pdf of the received
% signal better.
subplot(223);
plot(SNR, maxEstimates);
xlabel('SNR');
ylabel('mean of ML Estimates per variance');
title('ML Estimate vs. SNR');
```

```
Elapsed time is 848.597571 seconds.
```



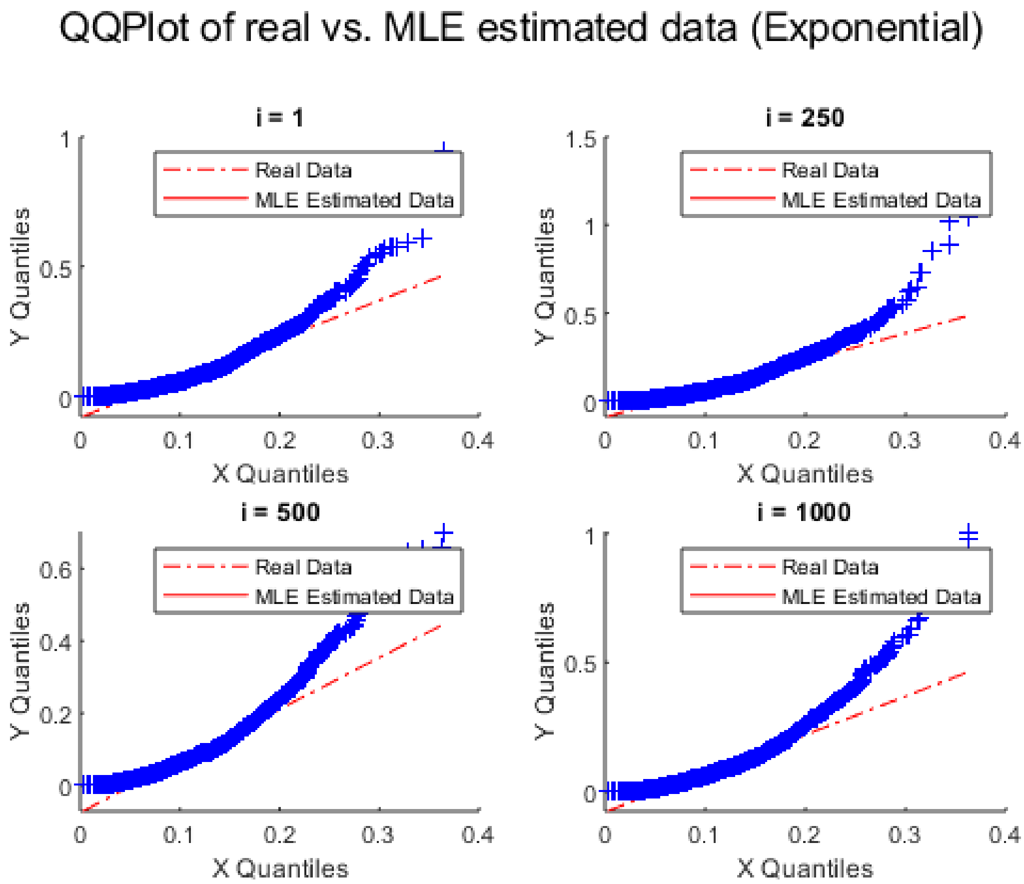
Scenario 3

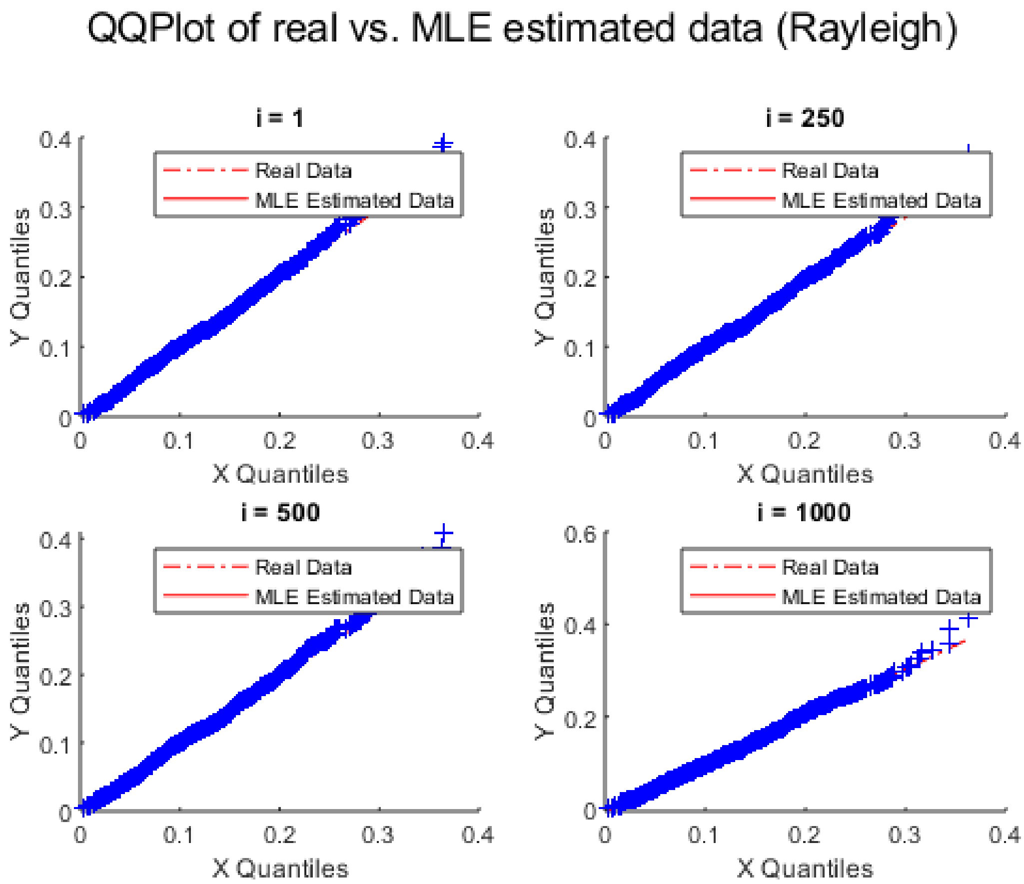
```
% Provided: [1x1000] data - either exponential or Rayleigh-distributed.
% Goal: determine the distribution of the data points by using ML Estimator
%              and comparing against estimated value from the possible distributions give
n data as parameter
```

3 - (a)

```
file = load('data.mat');
data = file.data;
% Procedure:
%   (1) determine multiple iterations of ML estimates of each
%       expected distributions (Exponential and Rayleigh)
%   (2) plot quantile-quantile plot of estimates and data and see if which
%       distribution estimates follow the straight line better. Data points lying
%       in the straight line implies the distributions of the two are quite
%       similar (or same).

% Exponential Distribution
% expfit: MLL estimates the mean 'mu' of an exponentially distributed sample data
muhat_exp = expfit(data);
% iterate through multiple trials of generating random exponentially
% distributed data points, and compare against actual data on quantile-quantile plot
synthetic_exp = zeros([1,1000]);
for i=1:1000
    synthetic_exp(i,:) = exprnd(muhat_exp,[1,1000]);
end
figure;
subplot(221); qqplot(data, synthetic_exp(1,:));
title(sprintf('i = %d',1));legend('Real Data','MLE Estimated Data');
subplot(222); qqplot(data, synthetic_exp(end/4,:));
```

```
title(sprintf('i = %d',length(synthetic_exp)/4));legend('Real Data','MLE Estimated Data');
subplot(223); qqplot(data, synthetic_exp(end/2,:));
title(sprintf('i = %d',length(synthetic_exp)/2));legend('Real Data','MLE Estimated Data');
subplot(224); qqplot(data, synthetic_exp(end,:));
title(sprintf('i = %d',length(synthetic_exp)));legend('Real Data','MLE Estimated Data');
suptitle({'QQPlot of real vs. MLE estimated data (Exponential)',''});

% Rayleigh Distribution
% raylfit: returns the maximum likelihood estimates of the parameter 'b' of the Rayleigh d
istribution given the data in the vector data
bhat_rayl = raylfit(data);
% iterate through multiple trials of generating random exponentially
% distributed data points, and compare against actual data on quantile-quantile plot
synthetic_rayl = zeros([1,1000]);
for i=1:1000
    synthetic_rayl(i,:) = raylrnd(bhat_rayl,[1,1000]);
end
figure;
subplot(221); qqplot(data, synthetic_rayl(1,:));
title(sprintf('i = %d',1)); legend('Real Data','MLE Estimated Data');
subplot(222); qqplot(data, synthetic_rayl(end/4,:));
title(sprintf('i = %d',length(synthetic_rayl)/4));legend('Real Data','MLE Estimated Data')
;
subplot(223); qqplot(data, synthetic_rayl(end/2,:));
title(sprintf('i = %d',length(synthetic_rayl)/2));legend('Real Data','MLE Estimated Data')
;
subplot(224); qqplot(data, synthetic_rayl(end,:));
title(sprintf('i = %d',length(synthetic_rayl)));legend('Real Data','MLE Estimated Data');
suptitle({'QQPlot of real vs. MLE estimated data (Rayleigh)',''});

% Since Rayleigh distribution ML Estimates follow the data in a straight
% line better than Exponential distribution ML Estimates, the data points
% are predicted to follow a Rayleigh distribution.
```

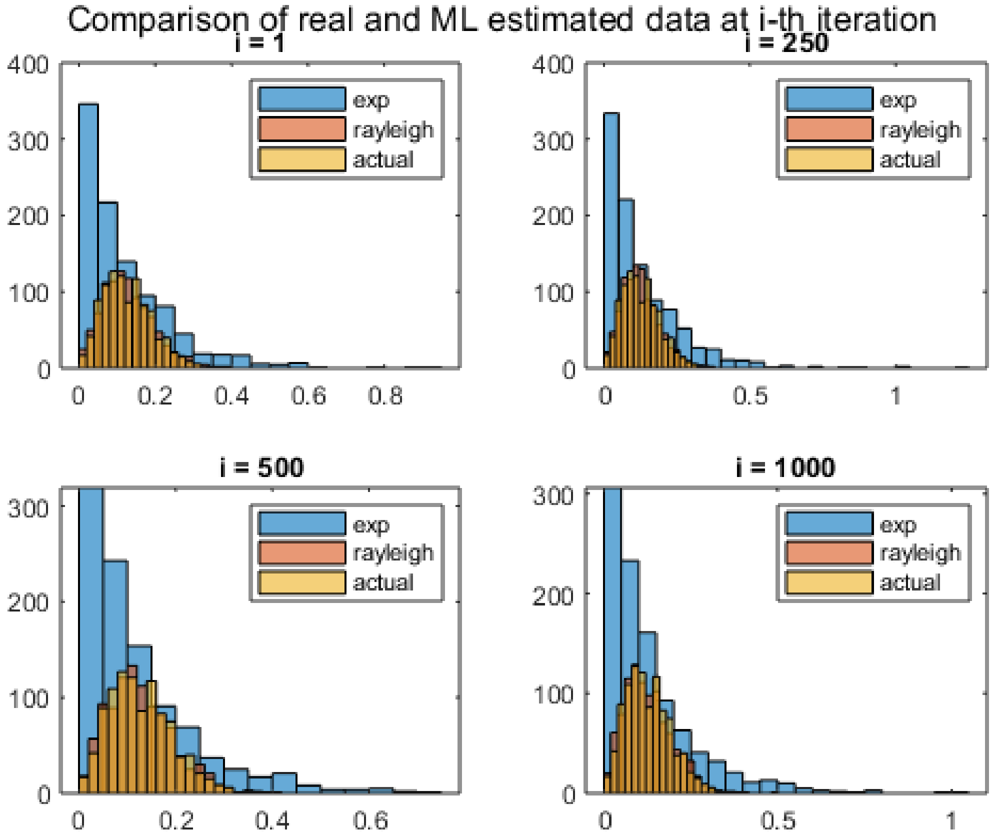




3 - (b)

```
% Compare histograms of real data, exponential ml estimates, and rayleigh ml estimates
% for 1st, quarter-way point iteration, half-way point, and final iteration
figure; t = suptitle({'Comparison of real and ML estimated data at i-th iteration',''});
```

```
set(t,'FontSize',12,'FontWeight','normal')
subplot(221);
histogram(synthetic_exp(1,:)); hold on;
histogram(synthetic_rayl(1,:)); hold on;
histogram(data);
title('i = 1');
legend('exp','rayleigh','actual');

subplot(222);
histogram(synthetic_exp(end/4,:)); hold on;
histogram(synthetic_rayl(end/4,:)); hold on;
histogram(data);
title(sprintf('i = %d',length(synthetic_exp)/4));
legend('exp','rayleigh','actual');

subplot(223);
histogram(synthetic_exp(end/2,:)); hold on;
histogram(synthetic_rayl(end/2,:)); hold on;
histogram(data);
title(sprintf('i = %d',length(synthetic_exp)/2));
legend('exp','rayleigh','actual');

subplot(224);
histogram(synthetic_exp(end,:)); hold on;
histogram(synthetic_rayl(end,:)); hold on;
histogram(data);
title(sprintf('i = %d',length(synthetic_exp)));
legend('exp','rayleigh','actual');

% the histogram comparison among the actual,
% Exponential-distribution-estimated, and Rayleigh-distribution-estimated
% data show that the actual data points mostly follow Rayleigh estimates
% determined from ML estimate (of the parameter). Thus one could predict
% the input data to be of Rayleigh distribution.
```



Comparison of real and ML estimated data at i-th iteration
i = 1
i = 250
i = 500
i = 1000
exp
rayleigh
actual