Introduction to Environmental Data Analysis and Statistical Learning

Exercise 6 – Bayesian learning

Ka Hei Pinky, Chow


```
clear all
close all
clc

load ue_06_bayesian_learning_data
whos
```

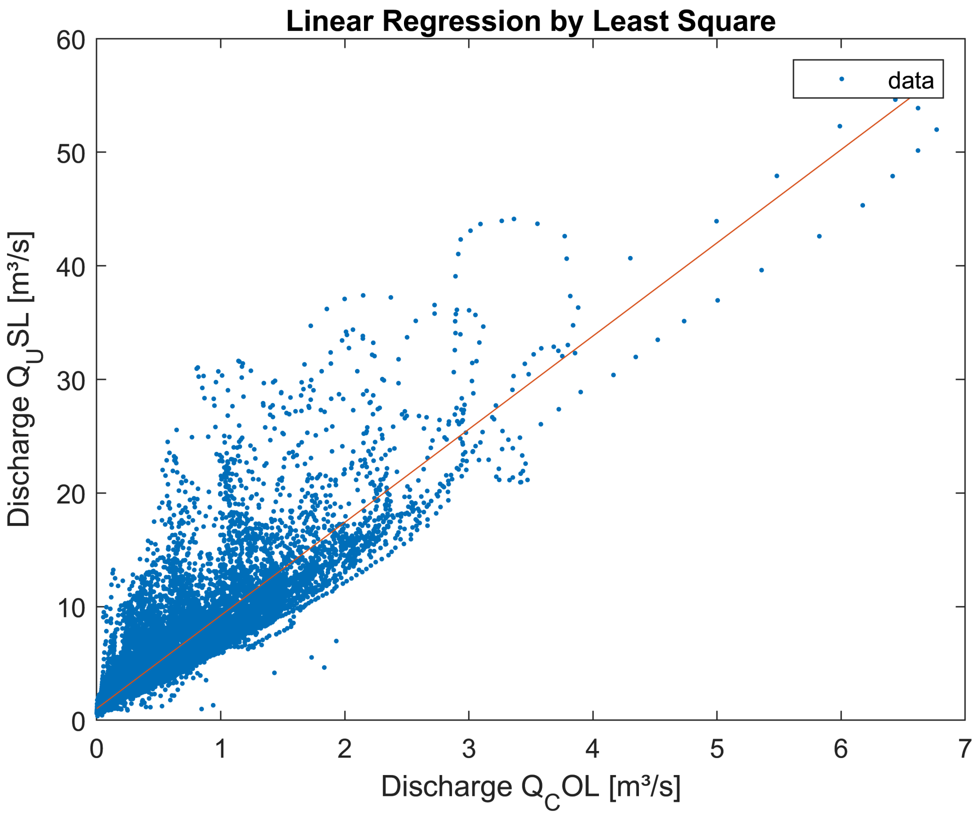
```
  Name           Size              Bytes  Class     Attributes

  Q_COL      43466x1              347728  double
  Q_USL      43466x1              347728  double
```

Method 1: Least Square Linear Regression

```
%assume Q_USL = a * Q_COL + b
a = polyfit(Q_COL,Q_USL,1);

display("Hence, a = " + a(1) + " and b = " + a(2));
```

```
    "Hence, a = 8.2007 and b = 1.0203"
```

```
y = polyval(a,Q_COL);
figure
plot(Q_COL,Q_USL,'.')
hold on
plot(Q_COL,y)
hold off
xlabel("Discharge Q_COL [m³/s]")
ylabel("Discharge Q_USL [m³/s]")
title("Linear Regression by Least Square")
legend("data")
```



Results: a = 8.2007; b = 1.0203

Method 2: Bayesian Learning

```
% define some variables
a2 = 1:0.1:20; %the range of the hypothesis
b2 = 1.0203; %[m³/s]
num_obs = length(Q_USL); %number of observation

%Construct the prior probability distribution of model
num_h = length(a2);  %the length of model (hypothesis)
p_prior = ones(num_h,1);
p_prior = p_prior*(1/num_h);

% construct the likelihood functions for each hypothesis
% rows = model (hypothesis), cols = height of the normpdf
```

One-by-one bayesian learning

```
% loop over all observations
for o = 1 : num_obs %number of obs
    %Q_USL = a2 * Q_COL + b2
    %Q_USL_sim = [1:0.1:20] * Q_COL + 1.0203
    Q_USL_sim = a2 * Q_COL(o) + b2;
    num_o = length(Q_USL_sim); %the number of outcome
    sigma = 10; %standard deviation
```

```
    likelihood = normpdf(Q_USL_sim,Q_USL,sigma);

    % calculate the marginal probability of the observation by summing up,
    % over all models, the likelihood of the observation multiplied by the occurrence
    % probability of the model (prior)

        p_observation = 0; % initialize the marginal probability of the observation

        % loop over all models
        for m = 1 : num_h
            p_observation = p_observation + likelihood(m) * p_prior(m);
        end

    % apply Bayes' theorem to get the posterior probabiltiy of the
    % models given the observation

        p_posterior = NaN(num_h,1); % initialize the array for posterior distribution of models
        % loop over all models
        for m = 1 : num_h
            p_posterior(m) = ( likelihood(m) * p_prior(m) ) / p_observation;
        end

    % update the probability distribution of models
    p_prior = p_posterior;

    % express the progress
    if rem(o,500) == 0
        display("running..." + round(o*100/length(Q_USL)) + "%")
    end

end
```

```
"running...1.1503%"

"running...2.3006%"

"running...3.451%"

"running...4.6013%"

"running...5.7516%"

"running...6.9019%"

"running...8.0523%"

"running...9.2026%"

"running...10.3529%"

"running...11.5032%"

"running...12.6536%"

"running...13.8039%"

"running...14.9542%"

"running...16.1045%"
```

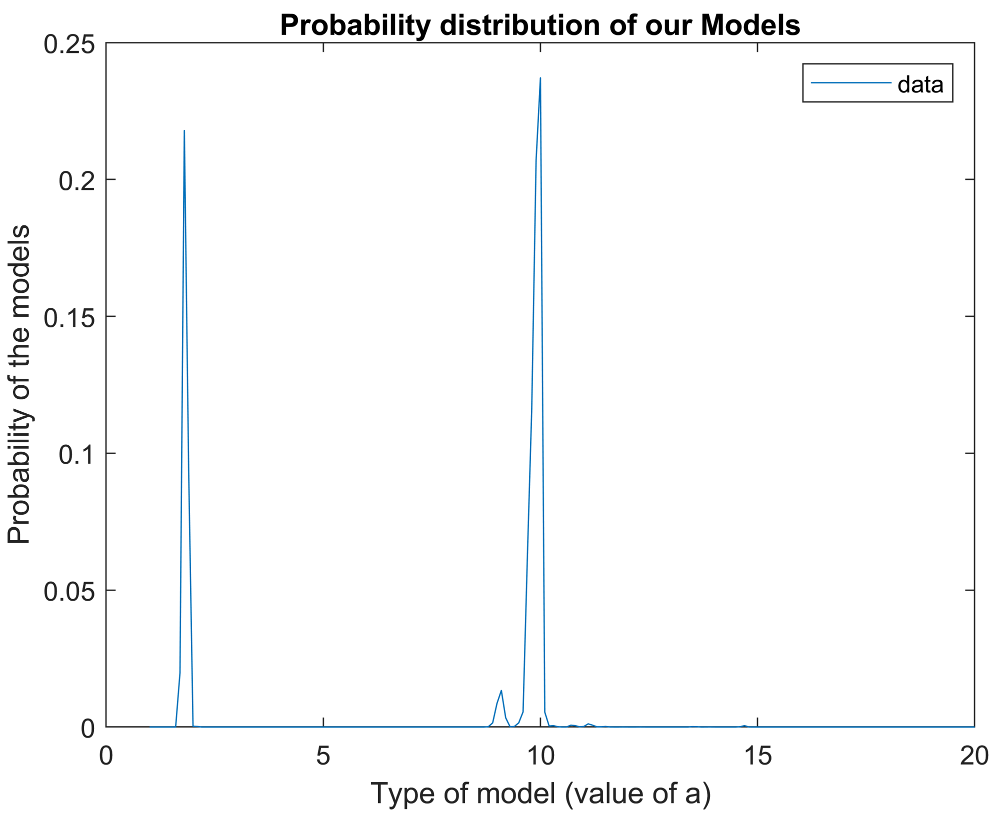
```
"running...17.2549%"
"running...18.4052%"
"running...19.5555%"
"running...20.7058%"
"running...21.8562%"
"running...23.0065%"
"running...24.1568%"
"running...25.3071%"
"running...26.4575%"
"running...27.6078%"
"running...28.7581%"
"running...29.9084%"
"running...31.0588%"
"running...32.2091%"
"running...33.3594%"
"running...34.5097%"
"running...35.6601%"
"running...36.8104%"
"running...37.9607%"
"running...39.111%"
"running...40.2614%"
"running...41.4117%"
"running...42.562%"
"running...43.7123%"
"running...44.8627%"
"running...46.013%"
"running...47.1633%"
"running...48.3136%"
"running...49.4639%"
"running...50.6143%"
"running...51.7646%"
"running...52.9149%"
```

```
"running...54.0652%"

"running...55.2156%"

"running...56.3659%"

"running...57.5162%"

"running...58.6665%"

"running...59.8169%"

"running...60.9672%"

"running...62.1175%"

"running...63.2678%"

"running...64.4182%"

"running...65.5685%"

"running...66.7188%"

"running...67.8691%"

"running...69.0195%"

"running...70.1698%"

"running...71.3201%"

"running...72.4704%"

"running...73.6208%"

"running...74.7711%"

"running...75.9214%"

"running...77.0717%"

"running...78.2221%"

"running...79.3724%"

"running...80.5227%"

"running...81.673%"

"running...82.8234%"

"running...83.9737%"

"running...85.124%"

"running...86.2743%"

"running...87.4247%"

"running...88.575%"

"running...89.7253%"

"running...90.8756%"
```

```
"running...92.026%"

"running...93.1763%"

"running...94.3266%"

"running...95.4769%"

"running...96.6272%"

"running...97.7776%"

"running...98.9279%"
```

Plotting the final probability distribution of a

```
%Plotting the final probability distribution of a
plot(a2, p_prior)
xlabel("Type of model (value of a)")
ylabel("Probability of the models")
title("Probability distribution of our Models")
legend("data")
```



Find the value of a with the highest posterior probability

```
[max_value, idx] = max(p_prior);
a_BL = a2(idx);
```

```
display("The value of a from Least squares regression is: " + a(1))
```

```
"The value of a from Least squares regression is: 8.2007"
```

```
display("The value of a from Bayesian learning model is: " + a_BL)
```

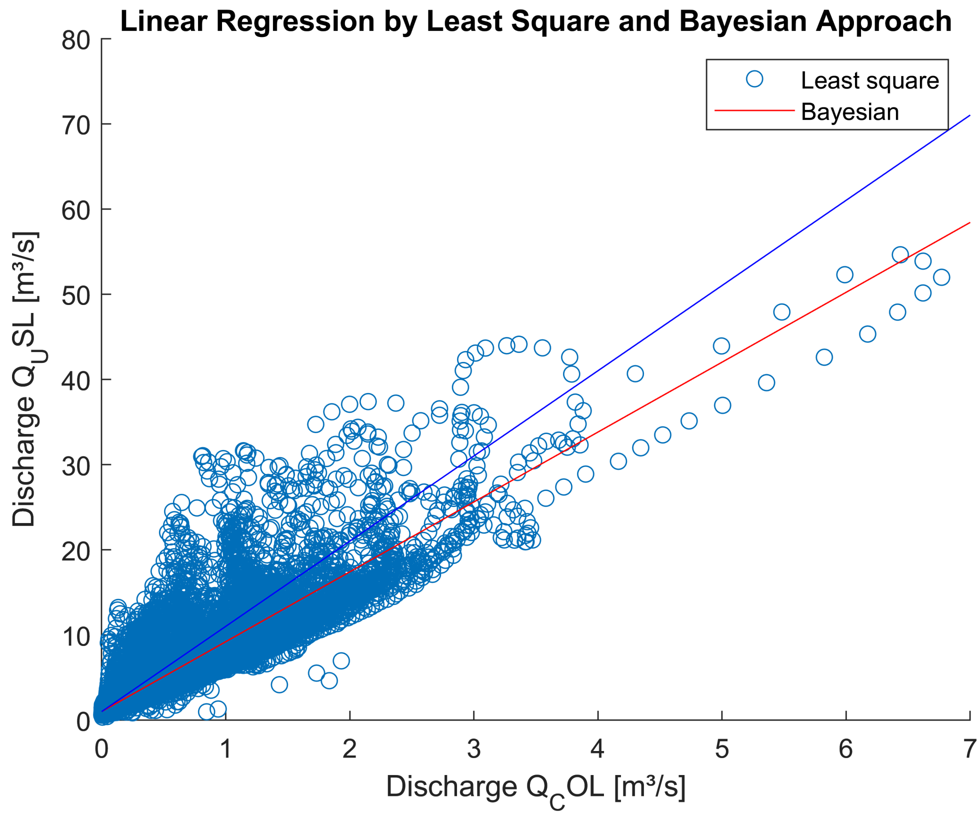
```
"The value of a from Bayesian learning model is: 10"
```

Observation:

The "a" value obtained form the least squares regression (a = 8.2) is close to the "a" value obtained in the Bayesian learning model (a = 10). It implies without using the frequentist approach, we can also use the introduction of prior value to formulate linear regression with probability distribution.

Plotting Results from both Methods

```
scatter(Q_COL,Q_USL,"o")
x1 = 0:0.1:round(max(Q_COL)); %Least square
y1 = a(1)*x1 + a(2);
hold on
plot(x1,y1,'color','r')
x2 = x1; %Bayesian
y2 = a_BL*x1 + b2;
plot(x2,y2,'color','b')
hold off

xlabel("Discharge Q_COL [m³/s]")
ylabel("Discharge Q_USL [m³/s]")
title("Linear Regression by Least Square and Bayesian Approach")
legend("Least square","Bayesian")
```


Linear Regression by Least Square and Bayesian Approach
Least square
Bayesian
Discharge Q_USL [m³/s]
Discharge Q_COL [m³/s]
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5
6
7