ISYE8843 Final

Jinyu Li

1. Question 1

I use Bayesian Wavelet Shrinkage to denoise. I choose a simple signal as following:

t=linspace(0,1,1024);
*sig = (sin(5*pi*t)+2.0*cos(10*pi*t)+3.0*sin(15*pi*t)).*exp(-t);*

The noise I add has different size, as 0.2, 0.4 and 0.6:

*sigma = 0.2*n; % n= 1:3*
randn('seed',1)
*sign = sig + sigma * randn(size(sig));*

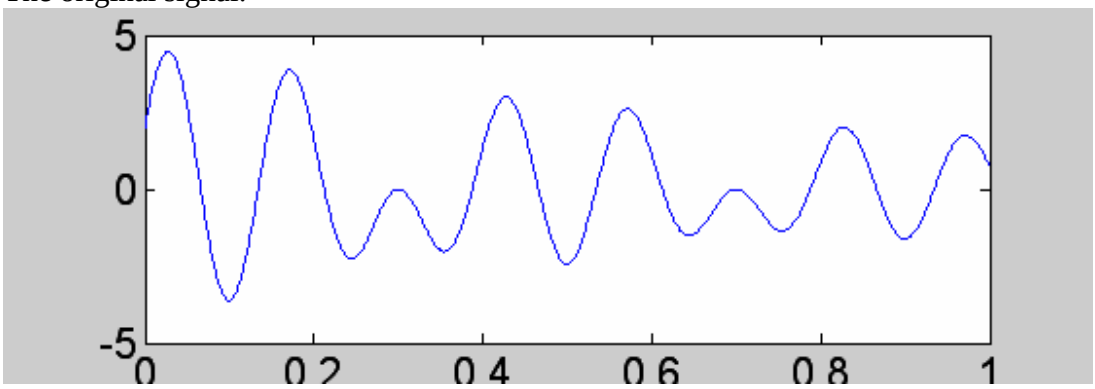
1.1 Matlab code

```
% Bayesian Wavelet Shrinkage
clear all
close all
% (i) Make a Signal on [0,1]
t=linspace(0,1,1024);
sig = (sin(5*pi*t)+2.0*cos(10*pi*t)+3.0*sin(15*pi*t)).*exp(-t);
% and plot it
figure(1)
plot(t, sig)

for n = 1: 3
% (ii) Add noise of size n*0.2. Make sure the noise is fixed
% by fixing the seed
sigma = 0.2*n;
randn('seed',1)
sign = sig + sigma * randn(size(sig));

% (iii) plot the noisy signal here.

figure(n+1)
subplot(2,1, 1)
plot(t, sign)

% (iv) Take the filter H, in this case this is SYMMLET 4

filt = [ -0.07576571478934  -0.02963552764595  ...
        0.49761866763246   0.80373875180522  ...
```

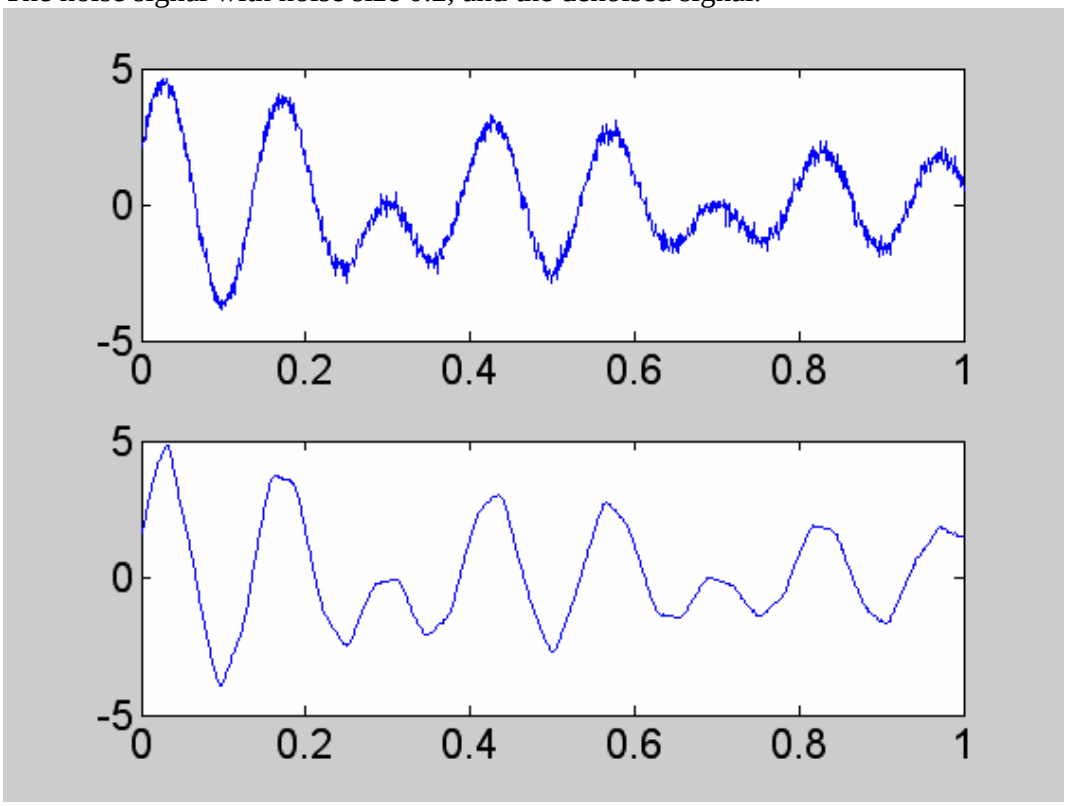
```
 0.29785779560554  -0.09921954357694  ...
-0.01260396726226   0.03222310060407];

% (v) Transfer the signal in the wavelet domain.
% Choose L=8, eight levels of decomposition

sw = dwtr(sign, 5, filt);

% At this point you may view the sw. Is it disbalanced?
% Is it decorrelated?

%(vi) Let's now apply Bayesian Shrinkage.
% Assume that the likelihood of a detail wavelet coefficient is
% normal (theta, sigma^2) and that the prior on theta
% is also normal (0, tau^2). The Bayes rule is:
%tau^2/(tau^2 + sigma^2) d. Take $tau^2=0.01 and
%sigma^2=0.1.
swt = sw;
swt(2^5+1:end) = swt(2^5+1:end).*0.01/0.11;
%swt = swt.*0.01/0.11;

% (vii) Return now thresholded object back to the time
% domain. Of course with the same filter and L.

a=idwtr(swt,5, filt);

% (viii) Check if made a good estimate...

subplot(2,1,2);
plot(t, a, '-')
end
```

1.2 Output Result

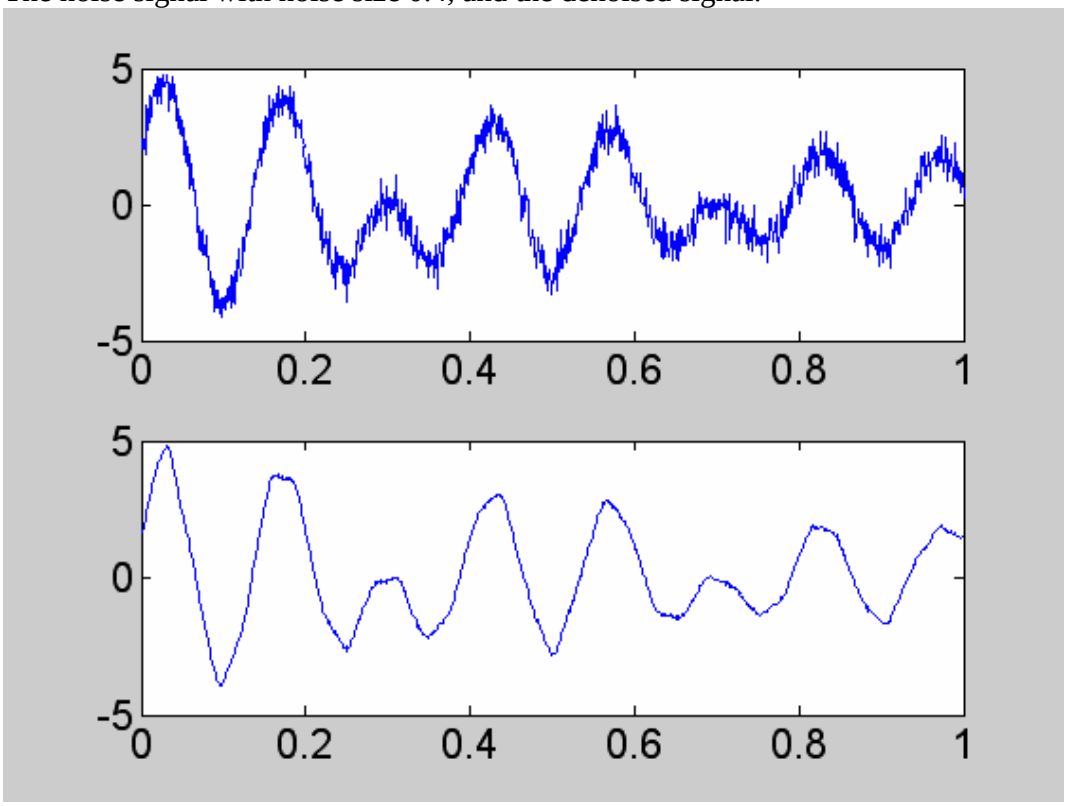
The original signal:



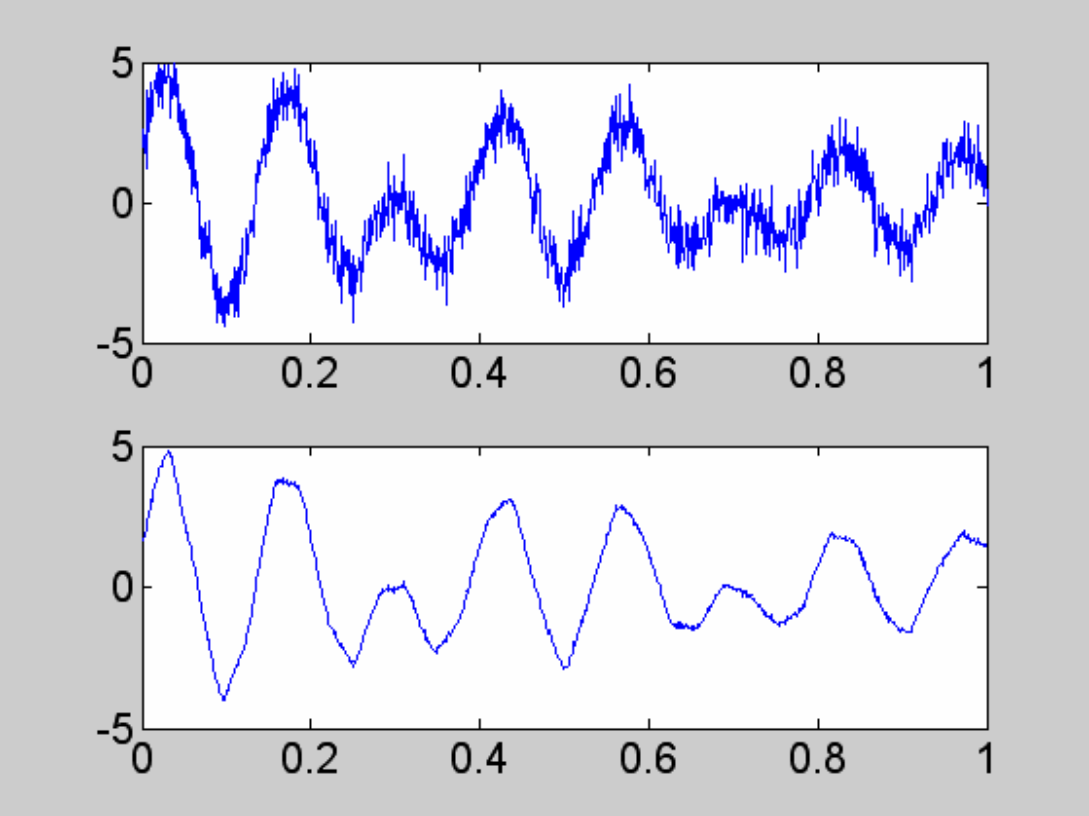
The noise signal with noise size 0.2, and the denoised signal.



The noise signal with noise size 0.4, and the denoised signal.



The noise signal with noise size 0.6, and the denoised signal.



As we see, all the denoised signals have nearly the same structure, and are similar to the original signal. The effect of Bayesian Wavelet Shrinkage is good for such signal.

2. Question 2

2.1 Exact Calculating

$$P(J1,M1|B1) = \frac{1}{P(B1)} P(B1,J1,M1)$$

$$= \frac{1}{P(B1)} \sum_{E,A} P(B1,E,A,J1,M1)$$

$$= \frac{1}{P(B1)} \sum_{E,A} P(B1) * P(E) * P(A|B1,E) * P(J1|A) * P(M1|A)$$

$$= \sum_{A} P(J1|A) * P(M1|A) * \left(\sum_{E} P(E) * P(A|B1,E) \right)$$

$$\sum_{E} P(E) * P(A0|B1,E) = P(E0) * P(A0|B1,E0) + P(E1) * P(A0|B1,E1)$$

$$= 0.998 * 0.06 + 0.002 * 0.05$$

$$= 0.05998$$

$$\sum_{E} P(E) * P(A1|B1,E) = P(E0) * P(A1|B1,E0) + P(E1) * P(A1|B1,E1)$$

$$= 0.998 * 0.94 + 0.002 * 0.95$$

$$= 0.94002$$

$$P(J1,M1|B1) = \sum_{A} P(J1|A) * P(M1|A) * \left(\sum_{E} P(E) * P(A|B1,E) \right)$$

$$= P(J1|A0) * P(M1|A0) * \left(\sum_{E} P(E) P(A0|B1,E) \right) + P(J1|A1) * P(M1|A1) * \left(\sum_{E} P(E) P(A1|B1,E) \right)$$

$$= 0.05 * 0.01 * 0.05998 + 0.9 * 0.7 * 0.94002$$

$$= 0.5922$$

2.2 Using Kevin Murphy's BNT

2.2.1 Matlab code:

```
N = 5;
dag = zeros(N, N);
B = 1; E =2; A = 3; J = 4; M =5;
false = 1; true = 2;
dag([B,E], A) = 1;
dag(A,[J,M]) = 1;
node_sizes = 2* ones(1, N);
bnet = mk_bnet(dag, node_sizes, 'names', {'B', 'E', 'A', 'J','M'});
%draw_graph(bnet.dag);
bnet.CPD{B} = tabular_CPD(bnet, B, [.999, .001]);
bnet.CPD{E} = tabular_CPD(bnet, E, [.998, .002]);
bnet.CPD{J} = tabular_CPD(bnet, J, [.95, .10, .05, .90]);
bnet.CPD{M} = tabular_CPD(bnet, M, [.99, .30, .01, .70]);
```

```
bnet.CPD{A} = tabular_CPD(bnet, A, [.999, .06, .71, .05, .001, .94, .29, .95]);
engine = jtree_inf_engine(bnet);

%   get p(M1 | B1)
evidence = cell(1,N);
evidence{B} = true;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, M);
p_m1_b1 = marg.T(true)

evidence = cell(1,N);
evidence{M} = true;
evidence{B} = true;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, J);
p_j1_m1b1 = marg.T(true)

p = p_m1_b1 * p_j1_m1b1
```

2.2.2 Output

P(M1|B1) = 0.6586
P(J1|M1,B1) = 0.8992
P(J1,M1|B1) = P(M1|B1) * P(J1|M1,B1) = 0.6586 * 0.8992 = 0.5922

2.3 Using WinBUGs

2.3.1 WinBUGs code

To Compute P(M1|B1)

```
model  {
  burglary   ~ dcat(p.burglary[]);
  earthquake  ~ dcat(p.earthquake[]);
  alarm ~ dcat(p.alarm[burglary,earthquake,]);
  Marycalls ~ dcat(p.Marycalls[alarm,]);
  Johncalls ~ dcat(p.Johncalls[alarm,])
  }

DATA IN:
list(
 burglary = 2,
 p.burglary = c(0.999,0.001),
 p.earthquake = c(0.998,0.002),
 p.alarm  = structure(.Data = c(0.999,   0.001,
                                0.71,    0.29,
                                0.06,    0.94,
                                0.05,    0.95), .Dim = c(2,2,2)),
p.Marycalls  =  structure(.Data = c(0.99, 0.01,
                                    0.30, 0.70), .Dim = c(2,2)),
p.Johncalls  =  structure(.Data = c(0.95, 0.05,
                                    0.10, 0.90), .Dim = c(2,2))   )
```

```
INTIS:
list(
 Marycalls = 1, Johncalls = 1, earthquake = 1, alarm = 1)
```

To compute P(J1|M1,B1)

```
model  {
  burglary   ~ dcat(p.burglary[]);
  earthquake  ~ dcat(p.earthquake[]);
  alarm ~ dcat(p.alarm[burglary,earthquake,]);
  Marycalls ~ dcat(p.Marycalls[alarm,]);
  Johncalls ~ dcat(p.Johncalls[alarm,])
  }

DATA IN:
list(
 Marycalls = 2, burglary = 2,
 p.burglary = c(0.999,0.001),
 p.earthquake = c(0.998,0.002),
 p.alarm  = structure(.Data = c(0.999,   0.001,
                                0.71,    0.29,
                                0.06,    0.94,
                                0.05,    0.95), .Dim = c(2,2,2)),
p.Marycalls  =  structure(.Data = c(0.99, 0.01,
                                    0.30, 0.70), .Dim = c(2,2)),
p.Johncalls  =  structure(.Data = c(0.95, 0.05,
                                    0.10, 0.90), .Dim = c(2,2))   )

INTIS:
list(
 Johncalls = 1, earthquake = 1, alarm = 1)
```

2.3.2 Output

P(M1|B1) = 0.663

node	mean	sd	MC error	2.50%	median	97.50%	start	sample
Marycalls	1.663	0.4727	0.004821	1	2	2	1001	10000

P(J1|M1,B1) = 0.902

node	mean	sd	MC error	2.50%	median	97.50%	start	sample
Johncalls	1.902	0.2976	0.002765	1	2	2	1001	10000

P(J1,M1|B1) = P(M1|B1) * P(J1|M1,B1) = 0.663 * 0.902 = 0.5980

3. Question 3

3.1 Exact Posterior Computation

$$P(\tau,\lambda,\mu,Y)=\left(\prod_{i=1}^{\tau}P(Y_i|\tau,\lambda)\right)*P(\lambda)*\left(\prod_{i=\tau+1}^{n}P(Y_i|\tau,\mu)\right)*P(\mu)*P(\tau)$$

$$=\left(\prod_{i=1}^{\tau}\left(\frac{\lambda^{Y_i}}{Y_i!}e^{-\lambda}\right)\right)*\frac{\beta_\lambda{}^{\alpha_\lambda}}{\Gamma(\alpha_\lambda)}*\lambda^{\alpha_\lambda-1}*e^{-\lambda\beta_\lambda}*\left(\prod_{i=\tau+1}^{n}\left(\frac{u^{Y_i}}{Y_i!}e^{-\mu}\right)\right)*\frac{\beta_\mu{}^{\alpha_\mu}}{\Gamma(\alpha_\mu)}*\mu^{\alpha_\mu-1}*e^{-\mu\beta_\mu}*\frac{1}{n}$$

So, the posteriors are:

$$P(\tau=k|\lambda,\mu,Y)\propto\left(\prod_{i=1}^{k}\left(\lambda^{Y_i}e^{-\lambda}\right)\right)*\lambda^{\alpha_\lambda-1}*e^{-\lambda\beta_\lambda}*\left(\prod_{i=k+1}^{n}\left(u^{Y_i}e^{-\mu}\right)\right)*\frac{\beta_\mu{}^{\alpha_\mu}}{\Gamma(\alpha_\mu)}*\mu^{\alpha_\mu-1}*e^{-\mu\beta_\mu}$$

$$=\lambda^{\sum_{i=1}^{K}Y_i}*e^{-k\lambda}*\lambda^{\alpha_\lambda-1}*e^{-\lambda\beta_\lambda}*u^{\sum_{i=k+1}^{n}Y_i}*e^{-(n-k)\mu}*\mu^{\alpha_\mu-1}*e^{-\mu\beta_\mu}$$

$$=\lambda^{\sum_{i=1}^{K}Y_i+\alpha_\lambda-1}*e^{-(k+\beta_\lambda)\lambda}*u^{\sum_{i=k+1}^{n}Y_i+\alpha_\mu-1}*e^{-((n-k)+\beta_\mu)\mu}$$

It should be normalized to get the sum probability 1 as

$$\sum_{k=1}^{n}P(\tau=k|\lambda,\mu,Y)=1$$

So, I get the normalization factor as

$$\sum_{k=1}^{n}\{\lambda^{\sum_{i=1}^{K}Y_i+\alpha_\lambda-1}*e^{-(k+\beta_\lambda)\lambda}*u^{\sum_{i=k+1}^{n}Y_i+\alpha_\mu-1}*e^{-((n-k)+\beta_\mu)\mu}\}$$

$$P(\lambda|\tau,\mu,Y)\sim\lambda^{\sum_{i=1}^{\tau}Y_i+\alpha_\lambda-1}*e^{-(\tau+\beta_\lambda)\lambda}$$

$$\sim gamma\left(\sum_{i=1}^{\tau}Y_i+\alpha_\lambda,\tau+\beta_\lambda\right)$$

$$P(\mu|\tau,\lambda,Y)\sim u^{\sum_{i=k+1}^{n}Y_i+\alpha_\mu-1}*e^{-((n-k)+\beta_\mu)\mu}$$

$$\sim gamma\left(\sum_{i=k+1}^{n}Y_i+\alpha_\mu,(n-k)+\beta_\mu\right)$$

3.2 MCMC Implementation

By ignoring constant terms the probability $P(\tau=k|\lambda,\mu,Y)$ is simplified as:

$$P(\tau=k|\lambda,\mu,Y)=\frac{e^{-(\lambda-\mu)k}*\left(\lambda/\mu\right)^{\sum_{i=1}^{K}Y_i}}{\sum_{k=1}^{n}\{e^{-(\lambda-\mu)k}*\left(\lambda/\mu\right)^{\sum_{i=1}^{K}Y_i}\}}$$

Use this probability expression to do the sampling of τ

Matlab code

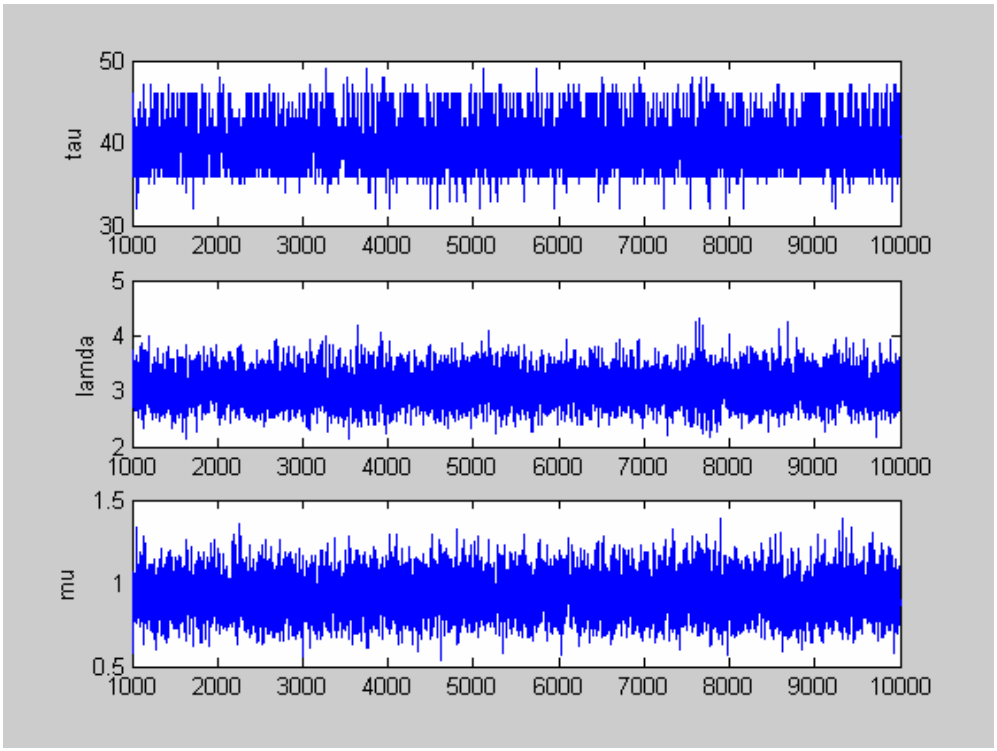
```
clear all
close all
```

```
NN = 10000;
burn_in = 1000;
bin = 100;

minedata =      [ 4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,...
                4,2,5,2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,...
                2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,...
                2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1];

% randomly select the initial value
tau_old = 1;
lamda_old = 1;
mu_old = 1;

% randomly select the parameter
alpha_lamda = 1;
alpha_mu = 1;
beta_lamda = 1;
beta_mu = 1;

taus = [];
lamdas = [];
mus = [];

%   Update parameters
for nn = 1: NN
   for k = 1: length(minedata)
      p_tau(k) = exp(-(lamda_old-
mu_old)*k)*((lamda_old/mu_old)^(sum(minedata(1:k))));
   end
   p_tau_norm = sum(p_tau);
   p_tau = p_tau/p_tau_norm;
   u = rand(1);
   t = 0;
   index = 1;
   for k = 1: length(minedata)
      t = t + p_tau(k);
      p_tau_sum(k) = t;
         if( t >= u)
            index = k;
            break;
         end
   end

   tau_new = index;
```

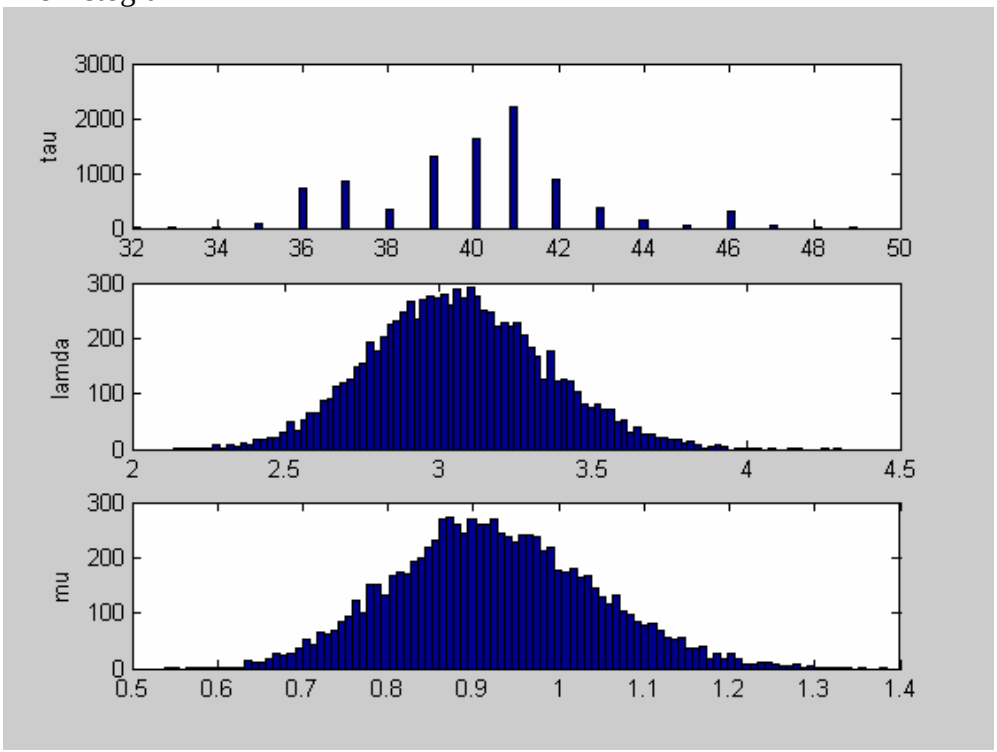
```
    lamda_new = rand_gamma(alpha_lamda+sum(minedata(1:tau_new)),
beta_lamda+tau_new);
    mu_new = rand_gamma(alpha_mu+sum(minedata(tau_new+1:length(p_tau))),
beta_mu+(length(p_tau)-tau_new));
    tau_old = tau_new;
    lamda_old = lamda_new;
    mu_old = mu_new;
    taus = [taus, tau_new];
    lamdas = [lamdas, lamda_new];
    mus = [mus, mu_new];
end

%   Plot the traces
figure(1);
subplot(3, 1, 1);
plot((burn_in:NN), taus( burn_in:NN));
ylabel('tau');
subplot(3,1,2);
plot((burn_in:NN), lamdas(burn_in:NN));
ylabel('lamda');
subplot(3,1,3);
plot((burn_in:NN), mus(burn_in:NN));
ylabel('mu');

%   Plot the histograms
figure(2);
subplot(3,1,1);
hist(taus(burn_in:NN), bin);
ylabel('tau');
subplot(3,1,2);
hist(lamdas(burn_in:NN), bin);
ylabel('lamda');
subplot(3,1,3);
hist(mus(burn_in:NN), bin);
ylabel('mu');

 disp('mean of tau');
 mean(taus(burn_in:NN))
disp('mean of lamda');
 mean(lamdas(burn_in:NN))
disp('mean of mu');
 mean(mus(burn_in:NN))
```

3.3 Output

The trace



The histogram



mean of tau
ans =40.0933

mean of lamda
ans =3.0607

mean of mu
ans =0.9246

3.4 Discussion

$\tau(mean) = 40.0933$

$\lambda(mean) = 3.0607$

$\mu(mean) = 0.9246$

The mean of τ is about 40. That is to say, around 1890, there is a change point of the accident numbers. λ and μ are the expectation accident numbers before and after that change point. As we can see, from their mean values, before that change point, the accident number is about 3. After that change point, the accident number is less than 1.