FINAL EXAM (ISYE8843 FALL 2004)

Chengyuan Ma

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
gtg651t@mail.gatech.edu

1. BAYESIAN WAVELET SHRINKAGE

Wavelet Transformation transforms the original signal into two parts: approximation and details at different levels. From the perspective of signal reconstruction and de-noising, the details parts can be ignored or ”shrank” further, because the details can be regarded as the high frequency component of the signal, on the other hand, the coefficients of the details are really small.

Let’s return to Bayes rule, in general, Bayes rule will heavily shrink small arguments and only slightly large arguments. Obviously, if we use a proper Bayes rule on the coefficients in wavelet domain, we can expect it do a better job then the simple threshold method.

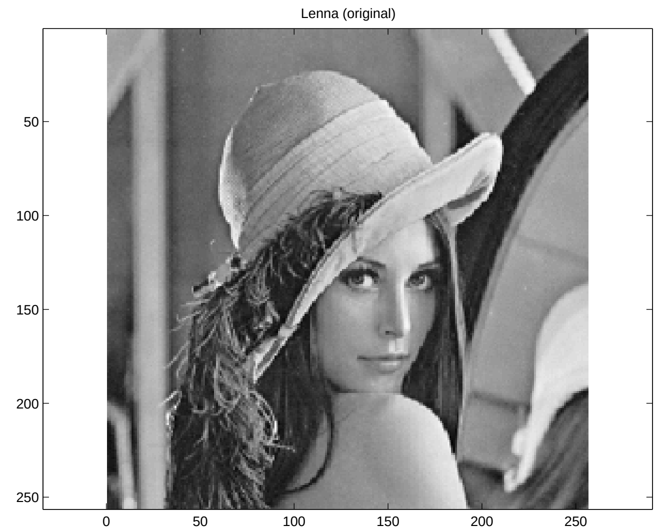
In this project, Bayes Wavelet Shrinkage is employed in image de-noising.

The Bayesian shrinkage rule we used is the following. Assume that the likelihood of a detail wavelet coefficient is $N(\theta, \sigma^2)$ and that the prior on θ is also $N(0, \tau^2)$. The Bayes rule for shrinkage is: $\frac{\tau^2}{\tau^2+\sigma^2}d$. Take $\tau^2 = 0.01$ and $\sigma^2 = 0.1$.

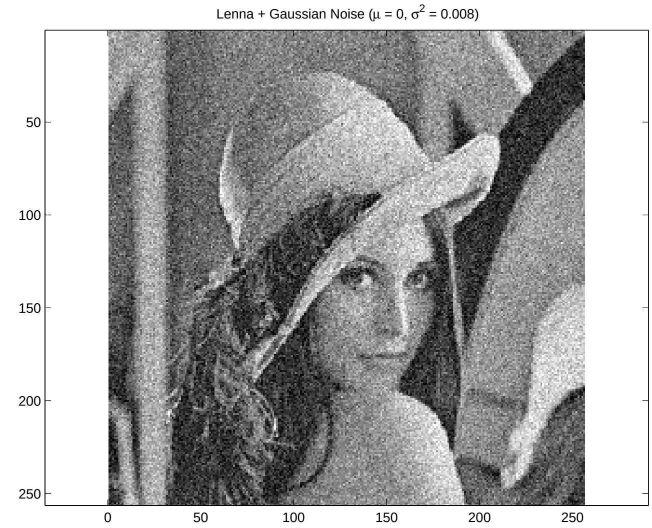
From our experiment result, we find that the performance of denoising is not good. One reason is that the model we imposed on the detail wavelet coefficient is not suitable for image. For images, Potts model or other statistical image models are expected to achieve better result.

```
close all ;
clear ;

% wavelet shrinkage
I = imread('D:\Lenna.bmp') ;

figure(1) ;
imagesc(I) ;

% add gaussian noise to the image
J = imnoise(I,'gaussian', 0, 0.008) ;
figure(2) ;
imagesc(J) ;


% 2-D wavelet transformation
[ca, ch, cv, cd] = dwt2(J, 'db1', 'mode', 'sym') ;

% display the wavelet coefficients
figure(3) ;
subplot(221) ;
```

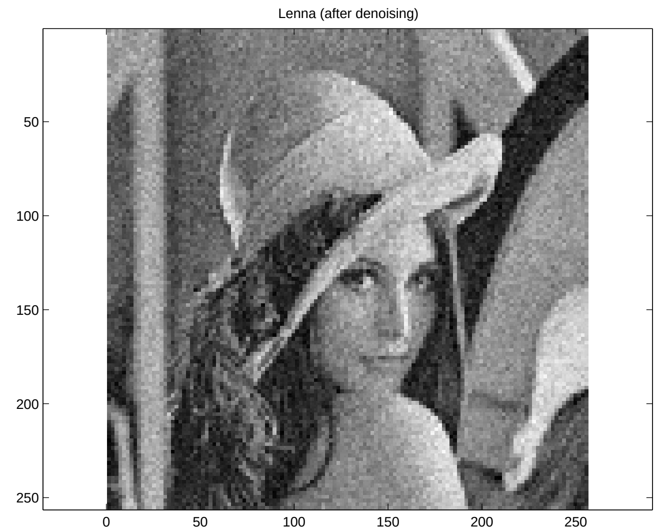
```
imagesc(ca) ;
subplot(222) ;
imagesc(ch) ;
subplot(223) ;
imagesc(cv) ;
subplot(224) ;
imagesc(cd) ;

% display the histogram of wavelet coefficients
figure(4) ;
subplot(221) ;
hist(reshape(ca,128ˆ2,1),100) ;
subplot(222) ;
hist(reshape(ch,128ˆ2,1),100) ;
subplot(223) ;
hist(reshape(cv,128ˆ2,1),100) ;
subplot(224) ;
hist(reshape(cd,128ˆ2,1),100) ;

% Let’s now apply Bayesian Shrinkage.
% Assume that the likelihood of a detail wavelet coefficient is
% normal (theta, sigmaˆ2) and that the prior on theta
% is also normal (0, tauˆ2). The Bayes rule is:
% tauˆ2/(tauˆ2 + sigmaˆ2) d. Take $tauˆ2=0.01 and
% sigmaˆ2=0.1.

ch = ch * 0.01/0.11 ;
cv = cv * 0.01/0.11 ;
cd = cd * 0.01/0.11 ;
Jnew = idwt2(ca, ch, cv, cd, ’db1’, ’mode’,’sym’) ;

figure(5) ;
imagesc(Jnew) ;
```



(a) Lenna (original)



(b) Lenna + Gaussian Noise ($\mu = 0, \sigma^2 = 0.008$)



(c) Lenna (after denoising)

Fig. 1. Original Image, Noised Image and Denoised Image

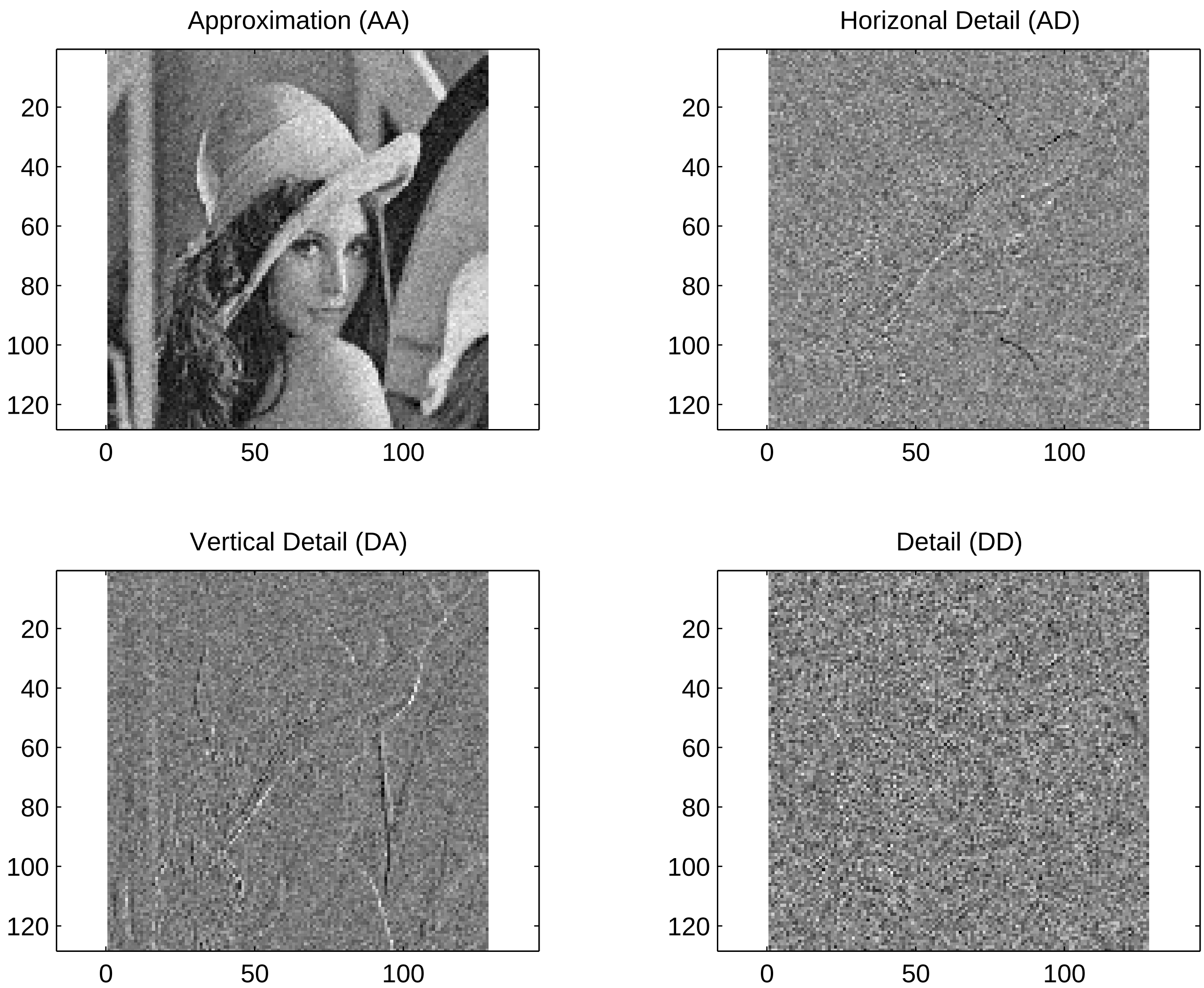


Fig. 2. 2-D Wavelet Transformation of Original Image

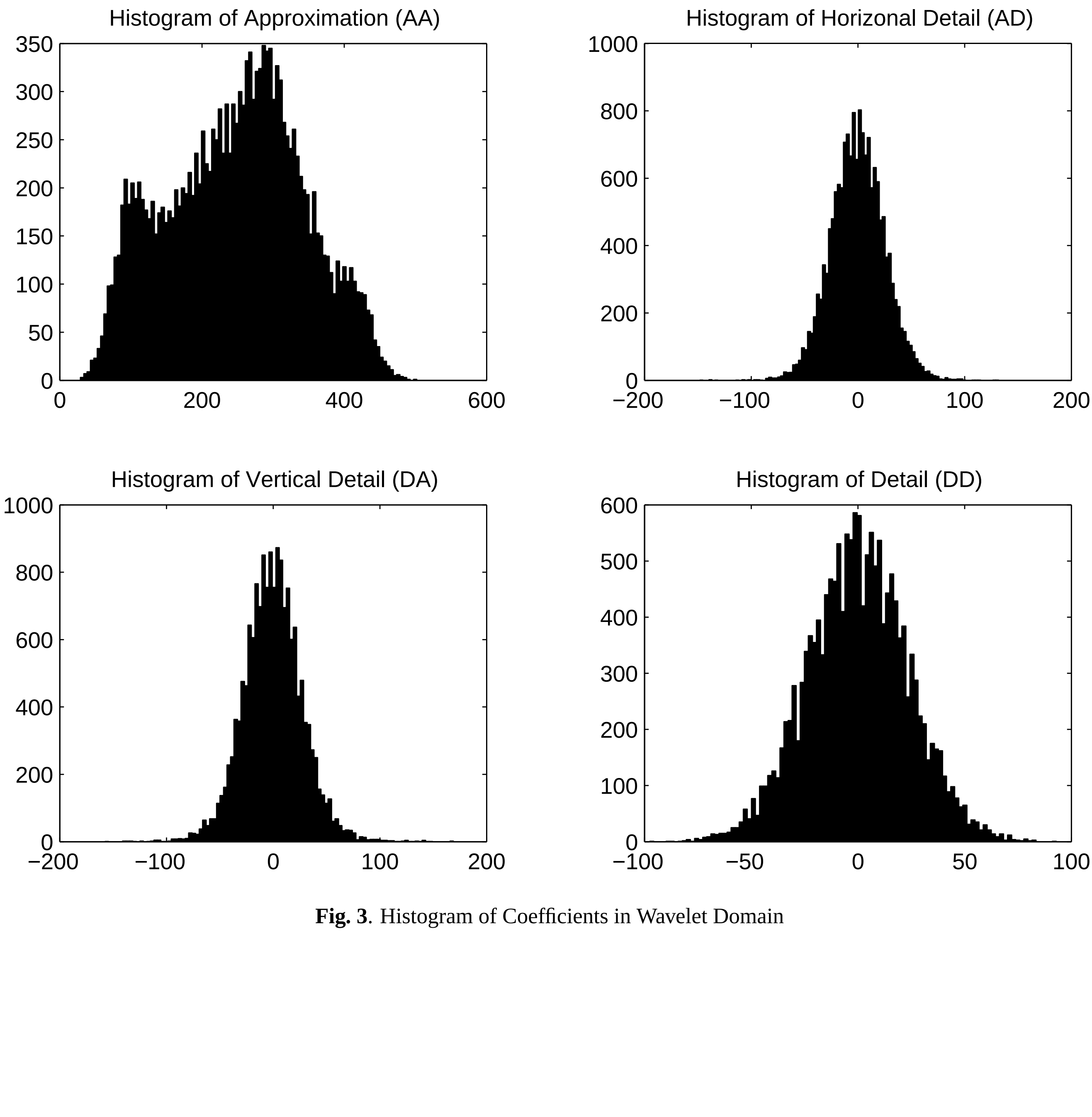


Fig. 3. Histogram of Coefficients in Wavelet Domain

2. ALARM - BAYESIAN NETWORK

2.1. Exact Solution

$$\begin{aligned}
P(J1, M1|B1) &= \frac{1}{P(B1)} P(B1, J1, M1) \\
&= \frac{1}{P(B1)} \sum_{E,A} P(B1, E, A, J1, M1) \\
&= \frac{1}{P(B1)} [P(J1|A1)P(M1|A1)P(A1|B1, E1)P(B1)P(E1) \\
&+ P(J1|A0)P(M1|A0)P(A0|B1, E1)P(B1)P(E1) \\
&+ P(J1|A1)P(M1|A1)P(A1|B1, E0)P(B1)P(E0) \\
&+ P(J1|A0)P(M1|A0)P(A0|B1, E0)P(B1)P(E0)] \\
&= 0.59224
\end{aligned}$$

2.2. Kevin Murphy's BNT

```
N = 5;
dag = zeros(N,N);
B = 1; E = 2; A = 3; JC = 4; MC = 5;
dag([B,E],A) = 1;
dag(A,[JC,MC]) = 1;
node_sizes = 2*ones(1,N);
bnet = mk_bnet(dag, node_sizes, 'names', {'B','E','A','JC','MC'});
draw_graph(bnet.dag);
false = 1; true = 2;
bnet.CPD{B} = tabular_CPD(bnet, B, [.999, .001]);
bnet.CPD{E} = tabular_CPD(bnet, E, [.998, .002]);
bnet.CPD{A} = tabular_CPD(bnet, A, [.999, .06, .71, .05, .001, .94, .29, .95]);
bnet.CPD{JC} = tabular_CPD(bnet, JC, [.95, .10, .05, .90]);
bnet.CPD{MC} = tabular_CPD(bnet, MC, [.99, .30, .01, .70]);

engine = jtree_inf_engine(bnet);
evidence = cell(1,N);
evidence{B} = true;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, MC);
>> marg.T

ans =

    0.3414
    0.6586

evidence = cell(1,N);
evidence{B} = true;
evidence{MC} = true;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, JC);
>> marg.T
ans =
```

```
    0.1008
    0.8992

>> 0.8992*0.6586
ans = 0.5922
```

2.3. WinBUGS

```
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,])
}

list(
#hard evidence , uncomment and instantiate...
 burglary = 2,
#earthquake = 1,
#alarm = 1,
 marycalls = 2,
#johncalls = 1,

#initial distributions
p.burglary = c(0.999,0.001),
p.earthquake = c(0.998,0.002),

# conditionals
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)))
```

Table 1. Parameters Statistics

node	mean	sd	MC error	2.5%	median	97.5%
marycalls	1.656	0.475	0.001459	1.0	2.0	2.0
johncalls	1.899	0.3015	9.347E-4	1.0	2.0	2.0

$$
\begin{aligned}
P(J1, M1|B1) &= P(J1|M1, B1)P(M1|B1) \\
&= 0.899 * 0.656 = 0.5897
\end{aligned}
$$

3. CHANGE POINT ANALYSIS

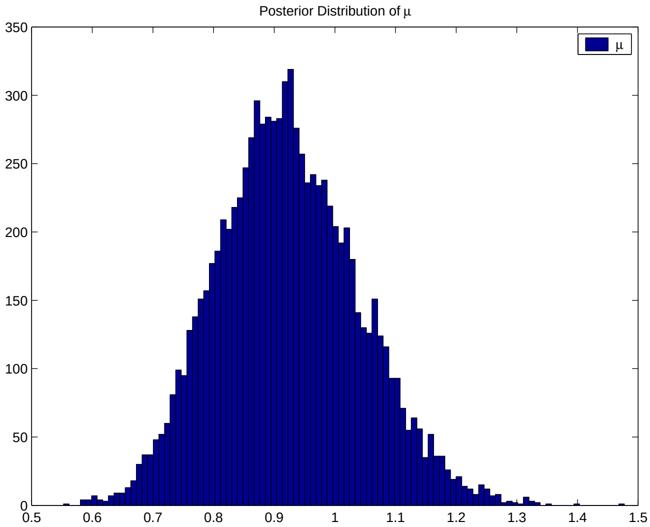
3.1. Matlab Implementation of MCMC method

```
function final3
clear all ;
close all ;

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];

% set random generator seed
randn('seed',1)

% set the parameters to be simulated
tau = [] ;
lambda = [] ;
mu = [] ;

% set the parameters
alphalambda = 1 ;
betalambda  = 1 ;

alphamu = 1 ;
betamu  = 1 ;

N = size(minedata, 2) ;

M = 10000 ;
burn = 1000 ;

% generate initial values
tau1 = 30 ;

% Starting simulation
for ii = 1 : M
    ii
    lambda1 = rand_gamma(alphalambda + sum(minedata(1:tau1)), betalambda + tau1) ;
    mu1 = rand_gamma(alphamu + sum(minedata(1+tau1:N)), betamu + N - tau1) ;
    tau1 = rand_categorical(minedata, lambda1, mu1) ;
    lambda = [lambda lambda1] ;
    mu     = [mu mu1] ;
    tau    = [tau tau1] ;
end

figure(1) ;
hist(mu(burn:M),100) ;
title('Posterior Distribution of \mu') ;
legend('\mu') ;


figure(2) ;
```

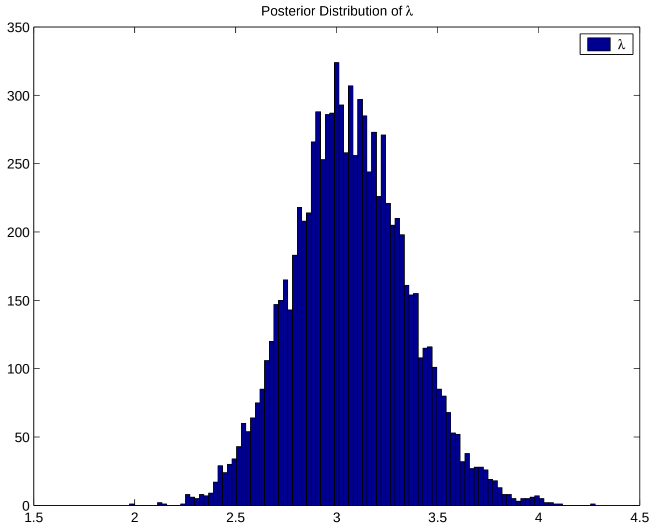
```
hist(lambda(burn:M), 100 ) ;
title('Posterior Distribution of \lambda') ;
legend('\lambda') ;

figure(3) ;
hist(tau(burn:M), 100) ;
title('Posterior Distribution of \tau') ;
legend('\tau') ;

function tau = rand_categorical(minedata, lambda1, mu1)
%
N = size(minedata, 2) ;

for ii = 1 : N
    p(ii) = exp((mu1 - lambda1)*ii + log(lambda1/mu1) * sum(minedata(1:ii)));
end

q = cumsum(p) ;
q = q/max(q) ;

temp = rand(1,1) ;
for ii = 1 : N
    if temp > q(ii)
        continue
    else
        tau = ii ;
        break ;
    end
end
```

3.2. Conclusion

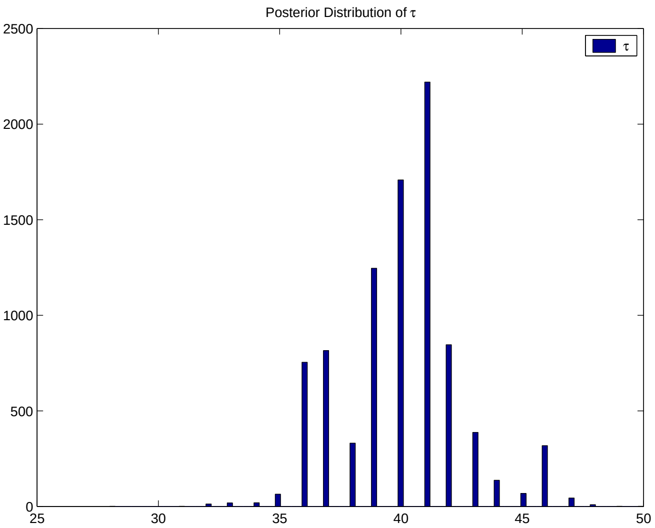
From the posterior distribution of τ we can find the mode of τ is about 41, and the mean of λ and μ are 3.06 and 0.92 respectively. So we can conclude that after year 1892, there are more successive coal mining accident happened.



(a) Posterior Distributio of μ



(b) Posterior Distributio of λ



(c) Posterior Distributio of τ

Fig. 4. Posterior Distributio