# In Class #1

## Problem Statement

Generate a random sample of size 200 from Gamma(3,2). Using MC Integration estimate the expectation and variance of the three quantities.

## Code

%Generate Data

x = gamrnd(3,2,1,200);

% Square Root of X

root\_x = sqrt(x);

exp\_root\_x = mean(root\_x);

var\_root\_x = var(root\_x);

% Trimmed Mean

tr\_mean = zeros(1,1000);

for i = 1:1000

x = gamrnd(3,2,1,200);

tr\_mean(i) = trimmean(x,20);

end

exp\_tr\_mean= mean(tr\_mean);

var\_tr\_mean = var(tr\_mean);

%Third Quartile

q3 = zeros(1,1000);

for i= 1:1000

x = gamrnd(3,2,1,200);

quarts = quartiles(x);

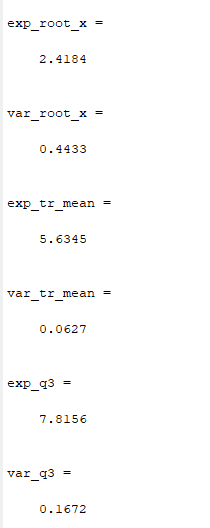
q3(i) = quarts(3);

end

exp\_q3= mean(q3);

var\_q3 = var(q3);

## Results



# In Class #2

## Problem Statement

Generate a random sample from Gamma(2,3) use an initial value of 2. Use a Normal as the candidate density.

1-2 Generate the data, plot the MC and calculate the mean and SD of the chain and compare with the true values.

3. Provide the kernel density estimation with the data you generated, plot this along with the historgram of your data and the true Gamma(2,3) distribution.

4. Repeat with exponential candidate.

## Code

% 1) Generate r.s of n=3000 using M-H, burn in the first 10%

n = 3000;

a = 2; b = 3; sig = sqrt(a\*b^2);

X1 = zeros(1,n);

rate=0;

X1(1) = 2;

for i = 2:n

y = normrnd(X1(i-1),sig);

u = rand(1);

alpha = min([1, gampdf(y, a, b)\*normpdf(X1(i-1), y, sig)/...

(gampdf(X1(i-1), a, b)\*normpdf(y, X1(i-1), sig))]);

if u <= alpha

X1(i) = y; rate=rate+1;

else

X1(i) = X1(i-1);

end

end

norm\_rate\_18 = rate/n

% Burn in 10%

n1=.1\*n;

X1=X1(n1+1:n);

%2 Plot the MCMC After Burn-In

figure(1)

plot(X1)

title('MCMC With Normal Candidate Distribution')

X1\_Bar = mean(X1)

true\_mean = a\*b

X1\_std = std(X1)

true\_std = sqrt(a\*b^2)

%3 Provide the Normal Kernel Density Estimation

n = length(X1);

hn = 1.06\*n^(-1/5)\*min(std(X1),iqr(X1)/1.348);

xx = linspace(0, max(X1)+1, 10000);

fhatN = zeros(size(xx));

for i=1:n

f=exp(-(1/(2\*hn^2))\*(xx-X1(i)).^2)/sqrt(2\*pi)/hn;

fhatN = fhatN+f/(n);

end

gam = gampdf(xx, a, b);

figure(2)

histogram(X1, 'Normalization', 'probability')

title('Histogram of Sample with Kernel density and True density')

hold on

plot(xx, fhatN, '-g')

plot(xx, gam, '-.r')

legend('MCMC Sample (Normal Candidate)', 'Kernel Density', 'Gamma pdf')

hold off

% Find out how the variance affects the mix rate

n = 3000;

a = 2; b = 3; sig = 1;

X1 = zeros(1,n);

rate=0;

X1(1) = 2;

for i = 2:n

y = normrnd(X1(i-1),sig);

u = rand(1);

alpha = min([1, gampdf(y, a, b)\*normpdf(X1(i-1), y, sig)/...

(gampdf(X1(i-1), a, b)\*normpdf(y, X1(i-1), sig))]);

if u <= alpha

X1(i) = y; rate=rate+1;

else

X1(i) = X1(i-1);

end

end

norm\_rate\_1 = rate/n

n = 3000;

a = 2; b = 3; sig = sqrt(50);

X1 = zeros(1,n);

rate=0;

X1(1) = 2;

for i = 2:n

y = normrnd(X1(i-1),sig);

u = rand(1);

alpha = min([1, gampdf(y, a, b)\*normpdf(X1(i-1), y, sig)/...

(gampdf(X1(i-1), a, b)\*normpdf(y, X1(i-1), sig))]);

if u <= alpha

X1(i) = y; rate=rate+1;

else

X1(i) = X1(i-1);

end

end

norm\_rate\_50 = rate/n

%Repeat 1-5 With Exponential Candidate

% 1) Generate r.s of n=3000 using M-H, burn in the first 10%

n = 3000;

a = 2; b = 3;

X2 = zeros(1,n);

rate=0;

X2(1) = 2;

for i = 2:n

y = exprnd(X2(i-1));

u = rand(1);

alpha = min([1, gampdf(y,a,b)\*exppdf(X2(i-1),y)/...

(gampdf(X2(i-1),a,b)\*exppdf(y,X2(i-1)))]);

if u <= alpha

X2(i) = y; rate=rate+1;

else

X2(i) = X2(i-1);

end

end

exp\_rate = rate/n

% Burn in 10%

n1=.1\*n;

X2=X2(n1+1:n);

%2 Plot the MCMC After Burnin

figure(3)

plot(X2)

title('MCMC With Exponential Candidate Distribution')

X2\_Bar = mean(X2)

true\_mean = a\*b

X2\_std = std(X2)

true\_std = sqrt(a\*b^2)

%3 Provide the Normal Kernel Density Estimation

n = length(X2);

hn = 1.06\*n^(-1/5)\*min(std(X2),iqr(X2)/1.348);

xx = linspace(0, max(X2)+1, 10000);

fhatN = zeros(size(xx));

for i=1:n

f=exp(-(1/(2\*hn^2))\*(xx-X2(i)).^2)/sqrt(2\*pi)/hn;

fhatN = fhatN+f/(n);

end

gam = gampdf(xx, a, b);

figure(4)

histogram(X2, 'Normalization', 'probability')

title('Histogram of Sample with Kernel density and True density')

hold on

plot(xx, fhatN, '-g')

plot(xx, gam, '-.r')

legend('MCMC Sample (Exponential Candidate)', 'Kernel Density', 'Gamma pdf')

## Results

Plot the MCMC



X1\_Bar = 6.1251

true\_mean = 6

X1\_std = 4.4135

true\_std = 4.2426

Mix Rate = 0.6210

Use Normal Kernel Density Estimation and plot the results with the histogram of the data and the true distribution



Repeat with Exponential

Plot the MCMC



X2\_Bar = 5.9520

true\_mean = 6

X2\_std = 4.0628

true\_std = 4.2426

mix\_rate = 0.5560

Use Normal Kernel Density Estimation and plot the results with the histogram of the data and the true distribution



Return to using the normal candidate. Experiment with how changing the variance affects the mix rate

With a Sigma of 1 we get a mix rate of .89

With a sigma of 100 we get a mix rate of .04

## Discussion

Both the normal and exponential candidate distributions did a good job of capturing the Gamma distribution. In terms of capturing the correct mean and standard deviation, both performed about the same. The mix rate of the exponential distribution was lower than that of the normal distribution, which could explain why the histogram of the exponential data shows a higher bump around the mean. Presumably the chain got stuck more often in this region and produced more data in it.

Returning to the Normal candidate and adjusting the sigma the following was observed. A higher standard deviation resulted in a much lower mix rate. This is to be expected as the larger sigma means a much wider range of candidates values are being generated, and they are much more likely to be rejected.

# In Class Problem #3a

## Problem Statement

Generate a hypothesized sample of size 100 from Bernoulli (p = 0.2). Use an M-H sampler to generate a MC of size 2000 whose invariant distribution is given by the posterior distribution the parameter p. Burn in 25% to calculate the mean and variance.

## Code

% 1. Use MH Sampler

% Generate 100 Samples From Bernoulli with p = 0.2

x = rand(100,1);

data = x>.8;

%Liklihood function

strg = 'theta^y\*(1-theta)^(100-y)';

L = inline(strg,'y','theta');

%MCMC

n = 2000;

theta1 = zeros(1,n);

theta1(1) = rand();

y = sum(data);

rate=0;

for i = 2:n

% Generate variate from proposal distribution.

v = rand(1);

% Generate variate from uniform.

u = rand(1);

% Calculate alpha.

alpha = min([1,L(y,v)/L(y,theta1(i-1))]);

if u <= alpha

% Then set the chain to the y.

theta1(i) = v; rate=rate+1;

else

theta1(i) =theta1(i-1);

end

end

accept\_rate\_1=rate/n;

n1=.25\*n;

theta1=theta1(n1+1:n);

MC\_MH\_Mean = mean(theta1);

MC\_MH\_Var = var(theta1);

figure(1)

subplot(2,1,1)

plot(theta1)

subplot(2,1,2)

histogram(theta1, 'Normalization', 'probability');

## Results



MC\_MH\_Mean = 0.2364

MC\_MH\_Var = 0.0018

accept\_rate\_1 = 0.1380

## Discussion

A prior of Uniform(0,1) was chosen, as its domain matched with the support of the target parameter. Each time a candidate was chosen it’s liklihood was compared with the likelihood of the previous values. More likely values were more likely to be accepted as the new choice. This algorithm did a pretty good job of approximating the true parameter value (2). It’s acceptance rate was low, but enough to generate variation in the candidates.

# In Class Problem #3b

## Problem Statement

In fact we do know the posterior distribution (Beta). Generate an MC for Beta using a Random Walk Metropolis Hastings sampler. Calculate the mean and standard deviation.

## Code

%2 Use known posterior to form a RW MH

theta2 = zeros(1,n);

theta2(1) = rand();

rate = 0;

for i = 2:n

% Generate variate via random walk

v = theta2(i-1) + (rand()-0.5)\*(1/sqrt(12));

% Generate variate from uniform.

u = rand(1);

a = y + 1; b = 101 - y ;

alpha = min([1,betapdf(v,a,b)/betapdf(theta2(i-1),a,b)]);

if u <= alpha

% Then set the chain to the y.

theta2(i) = v; rate=rate+1;

else

theta2(i) = theta2(i-1);

end

end

accept\_rate\_2=rate/n

n1=.25\*n;

theta2=theta2(n1+1:n)

MC\_RW\_Mean = mean(theta2)

MC\_RW\_Var = var(theta2)

figure(2)

subplot(2,1,1)

plot(theta2)

subplot(2,1,2)

histogram(theta2, 'Normalization', 'probability');



MC\_RW\_Mean = 0.2372

MC\_RW\_Var = 0.0019

accept\_rate\_2 = 0.4600

## Discussion

Using the Random-Walk MH sampler with the known posterior lead to an estimate of the parameter that was very similar to that of the first M-H sampler. If we look at the histogram of the data we notice that the tails in the latter sampler are a lot fatter. This is to be expected when we look at the acceptance rate for the random walk sampler. It is much higher than that of the first M-H sampler, meaning the value of the parameter estimate changes more frequently as the chain progresses. This will create a higher variation in the guesses.