Matlab Tutorials

April 2013

Tutorial 1 on Matlab

Solutions are provided in the directory Tutorial1Solutions :

- Question 1 : Tut1_Ex1.m

  Solutions : Copy and paste all the code in the Matlab 'command window'

- Question 2 : Tut1_Ex2.m
- Question 3 : Tut1_Ex3.m

Running a Matlab function :

- Click on the m file that you want to run
- Copy and paste in the 'command window' the first line without the word 'function'
- Choose adequate inputs for the function.
- For example (see Tut1_Ex2.m) : [Freq] = Tut1_Ex2(10000)

Codes contain a lot of explanations

Tutorial 1 on Matlab : Question 1

Bayesian Inferences using different techniques.

- $\mu | Y$ where $Y \sim t(\mu, \sigma^2, v)$

1. Draw $Y \sim t(\mu, \sigma^2, v)$ by using data augmentation (Gamma and Normal distribution)
2. Estimate μ and its standard error by Maximum likelihood
3. Comparison with Bayesian Inferences under different priors (Diffuse prior and Normal prior)
    - By deterministic integration
    - By Importance Sampling algorithm
    - By Metropolis-Hastings algorithm (MCMC)

Generate a sample of the student distribution from draws of a Inverse Gamma distribution and of a Normal distribution (see slide 143)

if $X|\lambda \sim N(\mu, \lambda\sigma^2)$ and $\lambda \sim IG_2(v, v)$

$$Y = \int f(X|\lambda)f(\lambda)d\lambda \sim t(\mu, \sigma^2, v)$$

- Draw from an Inverse Gamma distribution $(\lambda_t^2 \sim IG_2(df, df))$:

```
I_gam_draws = 1./gamrnd(df/2,2/df,N_draws,1);
```

- Combine the sample draws with the Normal distribution $(u_t \sim N(t\_mu, I\_gam\_draws\ \ t\_var^2))$:

```
t_mu + sqrt(I_gam_draws).*normrnd(0,1,N_draws,1)*sqrt(t_var);
```

In the program : $\mu = -1, \sigma^2 = 2$ and $v = 10$
Sample size : 5000

Compute the ML estimator of μ and its asymptotic standard error

- Run the fminunc Matlab command :

```
[MLE,not_used1,not_used2,not_used3,not_used4,hessian]= ...
fminunc(@(val_param) -1*log_like_stud(t_draws,val_param),...
Init_point,option);
```

Inputs :

- a objective function (here the student pdf)
- Initial point

  In complex minimisation problem, try different initial parameters.

Outputs :

1. The parameter that minimizes the function
2. Function value evaluated at the best parameter.
3. Hessian matrix at the best parameter

MLE : $\hat{\mu} = -0.98$ and st. error of $\hat{\mu} = 0.02$

Compute the posterior density of μ by Simpson's Integration

Deterministic integration :

- Trapezoidal :

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}$$

- Simpson :

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6}(f(a)+4f(\frac{a+b}{2})+f(b))$$

Drawback : Guess on the support of μ

- Define a grid of point where the pdf will be evaluated.

```
step = 0.001;
tau = (0.15:step:0.65)';
N_int = max(size(tau));
mapping = -log((1-tau)./tau);
```

- mapping $\in [-1.73, 0.62]$

Compute the posterior density of μ by Simpson's Integration

Deterministic integration :

- Trapezoidal :

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}$$

- Simpson :

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6}(f(a)+4f(\frac{a+b}{2})+f(b))$$

- Based on the grid, evaluation of the pdf :

```
for i=1:N_int
    param = [mapping(i);t_var;df];
    log_dens = log_like_stud(t_draws,param);
    diff_prior(i,1) = log_dens + log(grad(i));
    Normal_prior(i,1) = log_dens ...
    -0.5*log(2*pi*norm_var_prior)-0.5*(mapping(i)-norm_mu_prior)
    + log(grad(i));
end
```

- Two posterior distributions :
  - diff_prior denotes the diffuse prior : $f(\mu|Y) \propto f(Y|\mu)$
  - Normal_prior denotes the Normal prior : $f(\mu|Y) \propto f(Y|\mu)p(\mu)$

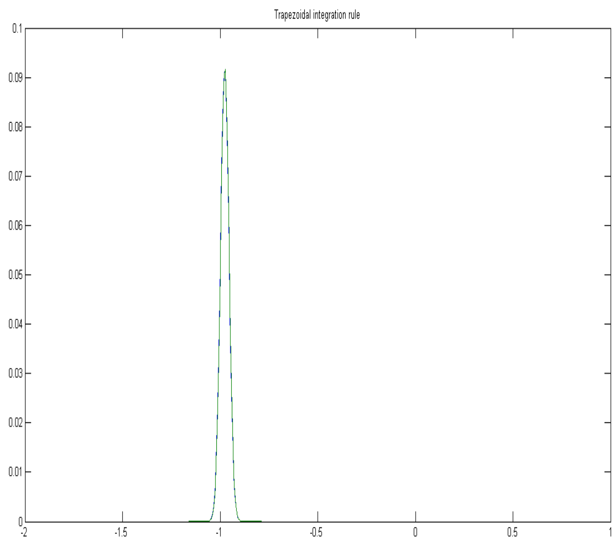
Compute the posterior density of μ by Simpson's Integration

Posterior distribution (where $Y \sim t(-1, 2, 10)$):

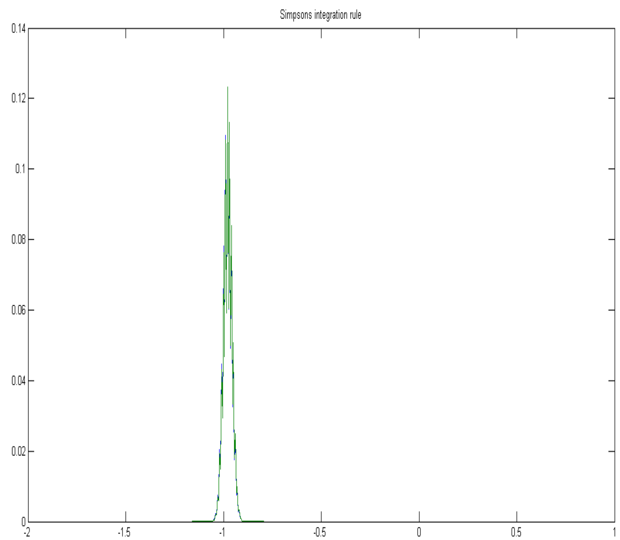
1. $E(\mu|Y) = -1$ and $std(\mu|Y) = 0.02$
2. MLE : $\hat{\mu} = -0.98$ and st. error of $\hat{\mu} = 0.02$

By deterministic integration :

- $E(\mu|Y) = -0.976$ and $std(\mu|Y) = 0.02$ with Trapezoidal rule
- $E(\mu|Y) = -0.978$ and $std(\mu|Y) = 0.02$ with Simpson's rule



Trapezoidal



Simpson

Compute the posterior density of μ by Importance sampling

$$
\begin{aligned}
1 &= \int f(\mu|Y)d\mu \\
1 &= \int \frac{f(Y|\mu)f(\mu)}{f(Y)}d\mu \\
f(Y) &= \int f(Y|\mu)f(\mu)d\mu \\
f(Y) &= \int \frac{f(Y|\mu)f(\mu)}{\iota(\mu)}\iota(\mu)d\mu \\
f(Y) &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(Y|\mu^i)f(\mu^i)}{\iota(\mu^i)} \quad \text{with i.i.d draws of } \iota(\mu)
\end{aligned}
$$

Drawback : Choice of the proposal distribution

The support of the proposal distribution $\iota(\mu)$
must be larger than the support of $\mu|Y$

Compute the posterior density of μ by Importance sampling

- Choice of the number of Importance sampling

```
N_IS = 5000;
```

- Choice of the proposal distribution

```
prop_var = 9*std_MLE(1)^2;
prop_draw = MLE(1) + randn(N_IS,1)*sqrt(prop_var);
```

- Importance sampling algorithm

```
parfor i=1:N_IS
    param = [prop_draw(i);t_var;df];
    log_dens_IS(i) = log_like_stud(t_draws,param);
    prop_dens_IS(i) = -0.5*log(2*pi*prop_var) - 0.5*(MLE_mu-prop_draw(i))^2/prop_var;
    diff_prior_IS(i) = log_dens_IS(i);
    Normal_prior_IS(i) = log_dens_IS(i) ...
    -0.5*log(2*pi*norm_var_prior)-0.5*(prop_draw(i)-norm_mu_prior)^2/norm_var_prior;
end
```

- Easily parallelized.

Compute the posterior density of μ by Importance sampling

Posterior distribution (where Y $\sim t(-1, 2, 10)$):

1. $E(\mu|Y) = -1$ and $std(\mu|Y) = 0.02$
2. MLE : $\hat{\mu} = -0.98$ and st. error of $\hat{\mu} = 0.02$

By Importance Sampling :

- $E(\mu|Y) = -0.98$ and $std(\mu|Y) = 0.02$

Effective Sample Size (where N is the number of IS) :

$$\text{ESS} = \frac{1}{\sum_{i=1}^{N} w_i^2}$$

$$\text{ESS} \in [1, \text{N}]$$

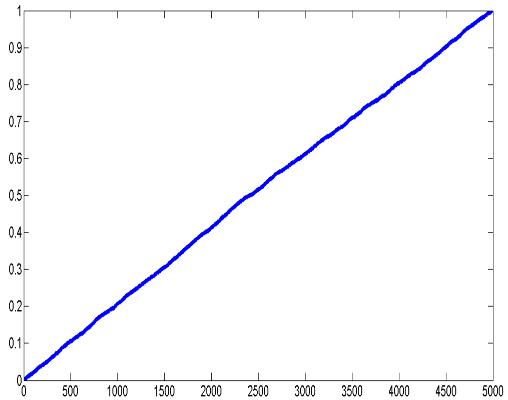
- ESS $= N$ if all the particles have the same weight.
- ESS $= 1$ if only one particle is relevant.

In the exercise :

$$\text{ESS} = 2262.47$$

Compute the posterior density of μ by Importance sampling

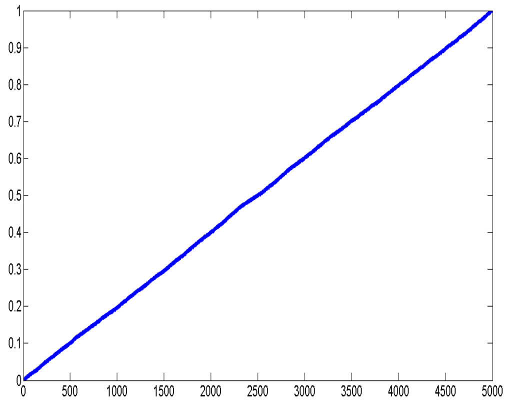
Posterior distribution (where Y $\sim t(-1, 2, 10)$):



$9\hat{\sigma}^2$ and $N = 5000$

ESS = 2262.47

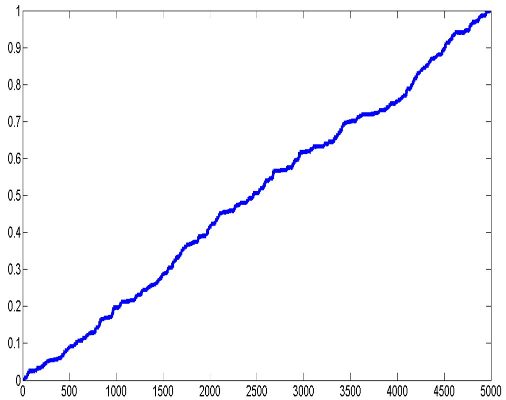
$E(\mu|Y) = -0.98$



$4\hat{\sigma}^2$ and $N = 5000$

ESS = 3246.93

$E(\mu|Y) = -0.98$



$1000\hat{\sigma}^2$ and $N = 5000$

ESS = 199.75

$E(\mu|Y) = -0.98$

Compute the posterior density of μ by MH algorithm

- Choice of the number of MH algorithm

```
nb_MCMC = 5000;
```

- Choice of the proposal distribution

```
prop_var = 9*std_MLE(1)^2;
prop_draw = MLE(1) + randn(N_IS,1)*sqrt(prop_var);
prop_dens_IS = -0.5*log(2*pi*prop_var) ...
- 0.5*(MLE_mu-prop_draw).^2/prop_var;
```

- Independent MH algorithm

```
for i=2:nb_MCMC
  if(exp(log_dens_IS(i)-log_dens_MCMC(i-1,1)+prop_dens_MCMC(i-1,1)-prop_dens_IS(i))>rand())
        mu_MCMC(i,1) = prop_draw(i);
        log_dens_MCMC(i,1) = log_dens_IS(i);
        prop_dens_MCMC(i,1) = prop_dens_IS(i);
        accept_rate(1) = accept_rate(1) +1;
  else
        mu_MCMC(i,1) = mu_MCMC(i-1,1);
        log_dens_MCMC(i,1) = log_dens_MCMC(i-1,1);
        prop_dens_MCMC(i,1) = prop_dens_MCMC(i-1,1);
  end
end
```

- Not easily parallelized.
- Drawback : Dependent sample and no approximation of $f(Y)$

Compute the posterior density of μ by MH algorithm

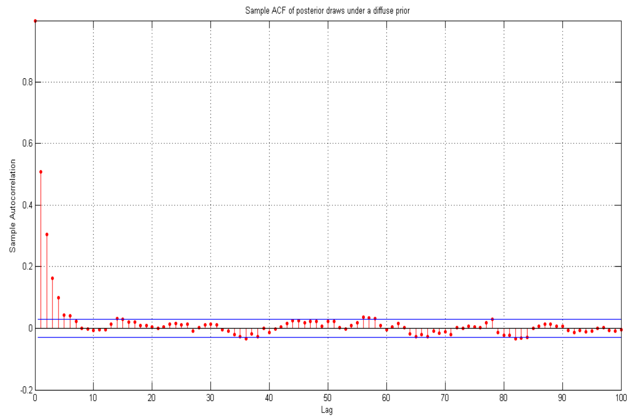
Posterior distribution (where $Y \sim t(-1, 2, 10)$):

1. $E(\mu|Y) = -1$ and $std(\mu|Y) = 0.02$
2. MLE : $\hat{\mu} = -0.98$ and st. error of $\hat{\mu} = 0.02$

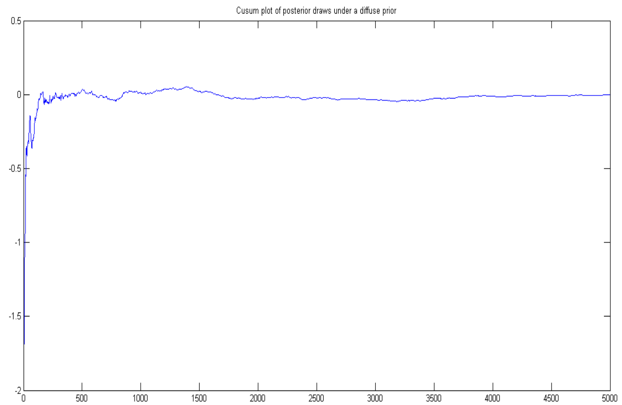
By Metropolis-Hastings algorithm :

- $E(\mu|Y) - 0.979$ and $std(\mu|Y) = 0.02$

Convergence test :



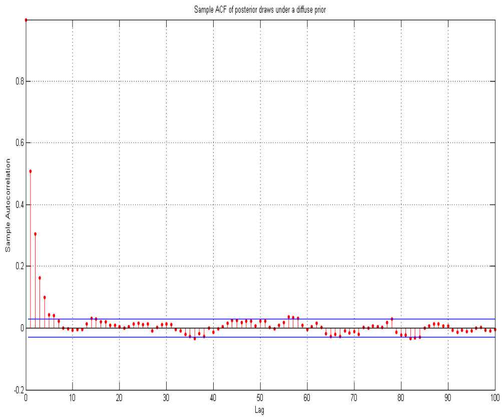
ACF



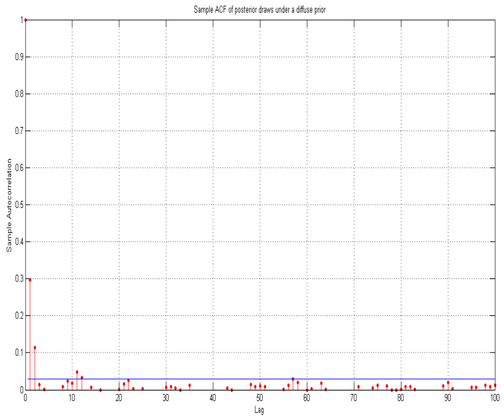
Cumsum plot

Compute the posterior density of μ by MH algorithm

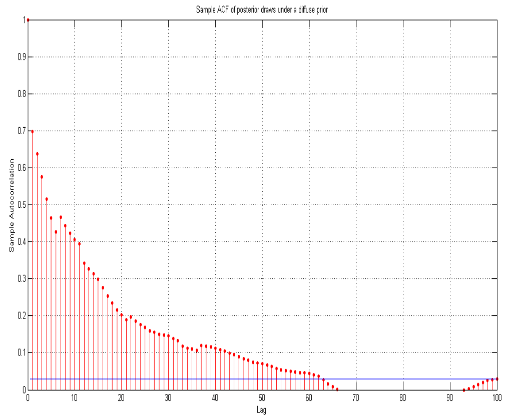
Posterior distribution (where $Y \sim t(-1, 2, 10)$):



$9\hat{\sigma}^2$ and $N = 5000$



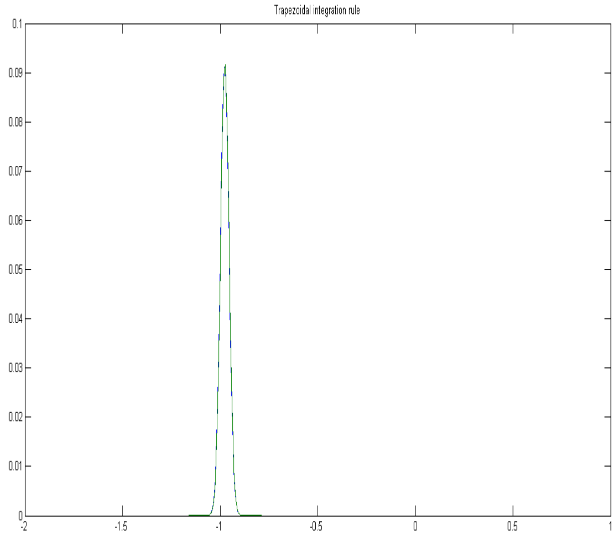
$4\hat{\sigma}^2$ and $N = 5000$



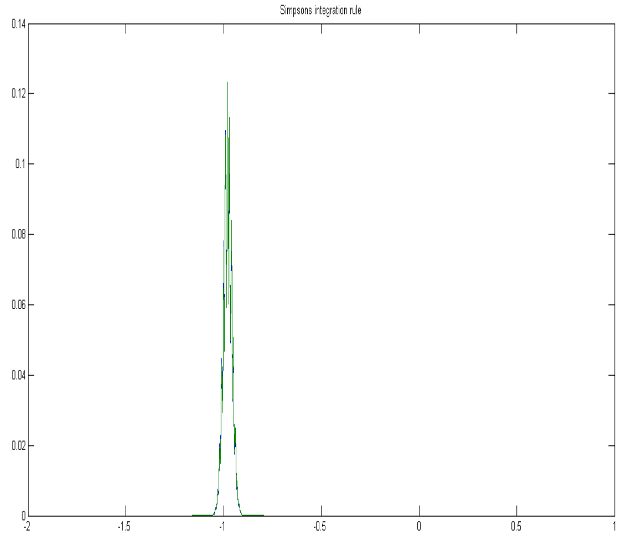
$1000\hat{\sigma}^2$ and $N = 5000$

Acc. rate = 0.39	Acc. rate = 0.57	Acc. rate = 0.04
$E(\mu \mid Y) = -0.98$	$E(\mu \mid Y) = -0.98$	$E(\mu \mid Y) = -0.98$

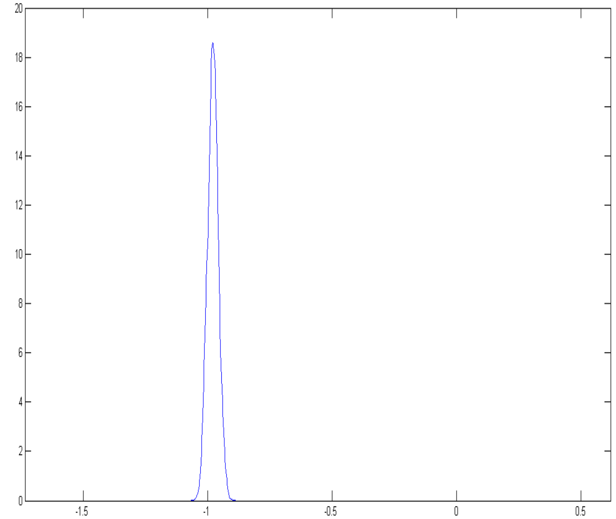
Summary : Posterior Density of μ



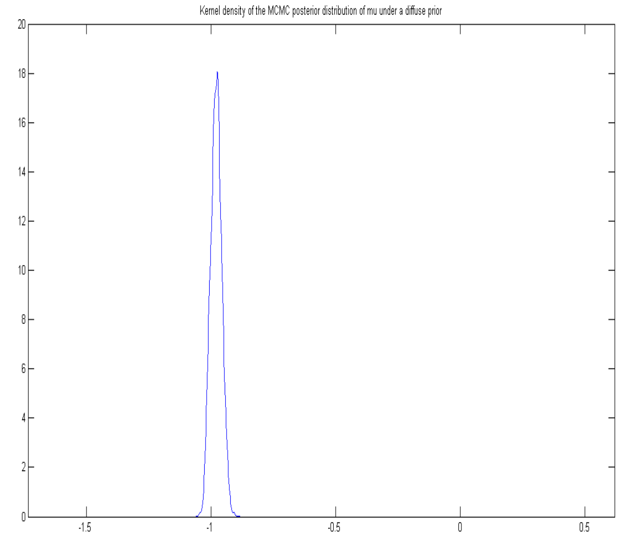
Trapezoidal



Simpson



IS



MH

Tutorial 1 on Matlab : Question 2

Bivariate Normal Distribution with $\mu = (1 - 1)'$*, and* $\Sigma = \begin{pmatrix} 0.25 & 0.2 \\ 0.2 & 0.64 \end{pmatrix}$

From N bivariate Normal draws $\Theta_i \sim N(\mu, \Sigma)$:

- Estimate μ

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \Theta_i$$

```
Freq.Expectation(i,:) = mean(post_draw(1:diff_N(i),:))
```

- Estimate Σ : Empirical Variance-Covariance matrix.

```
Freq.Cov_mat(:,:,i) = cov(post_draw(1:diff_N(i),:))
```

Bivariate Normal Distribution with $\mu = (1 \quad -1)'$, *and* $\Sigma = \begin{pmatrix} 0.25 & 0.2 \\ 0.2 & 0.64 \end{pmatrix}$

From N bivariate Normal draws $\Theta_i \sim N(\mu, \Sigma)$:

- Estimate $\Pr(\theta_1 \in A_1)$ where $A_1 = (0.5, 1.2)$

$$\frac{1}{N}\sum_{i=1}^{N} \delta_{\Theta_i \in A_1}$$

```
sum((post_draw(1:diff_N(i),1)>0.5).*(post_draw(1:diff_N(i),1)<1.2))/diff_N(i);
```

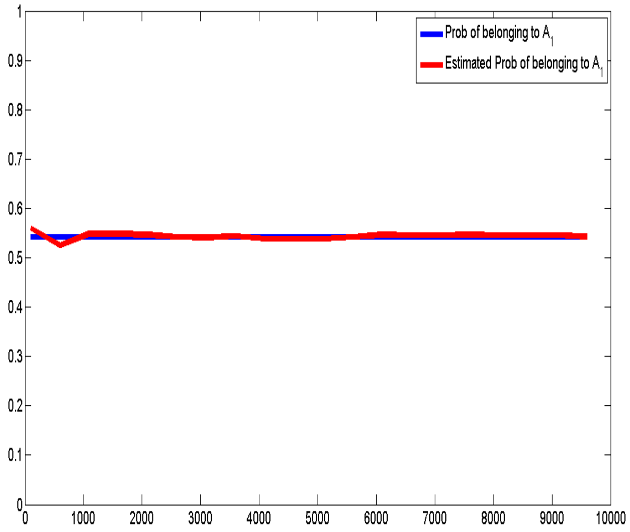
- Estimate $\Pr(\theta \in A)$ where $A_2 = (-1.5, 0)$ A as the Cartesian product of A_1 and A_2

$$\frac{1}{N}\sum_{i=1}^{N} \delta_{\Theta_i \in A}$$

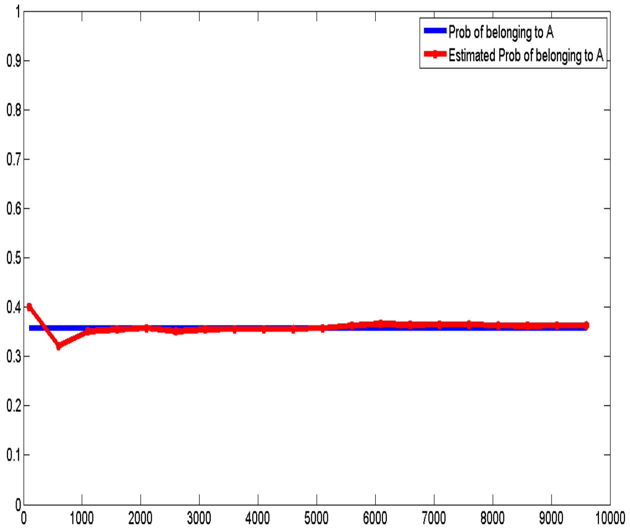
```
sum((post_draw(1:diff_N(i),1)>0.5).*(post_draw(1:diff_N(i),1)<1.2)...
.*(post_draw(1:diff_N(i),2)>-1.5).*(post_draw(1:diff_N(i),2)<0))/diff_N(i);
```

Bivariate Normal Distribution with $\mu = (1 - 1)'$, *and* $\Sigma = \begin{pmatrix} 0.25 & 0.2 \\ 0.2 & 0.64 \end{pmatrix}$

From N bivariate Normal draws $\Theta_i \sim N(\mu, \Sigma)$:



Prob $\in A_1$



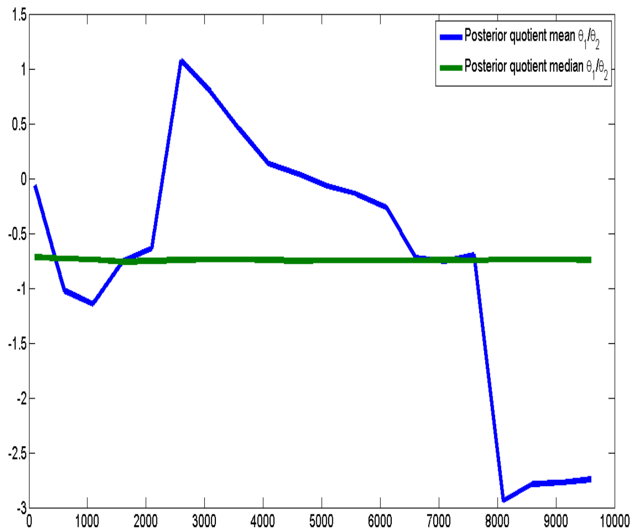
Prob $\in A$

Bivariate Normal Distribution with $\mu = (1 \quad -1)'$, *and* $\Sigma = \begin{pmatrix} 0.25 & 0.2 \\ 0.2 & 0.64 \end{pmatrix}$

From N bivariate Normal draws $\Theta_i \sim N(\mu, \Sigma)$:

- Estimate the mean and the median of $\frac{\theta_1}{\theta_2}$

```
mean(post_draw(1:diff_N(i),1)./post_draw(1:diff_N(i),2));
median(post_draw(1:diff_N(i),1)./post_draw(1:diff_N(i),2));
```



Ratio θ_1/θ_2

Tutorial 1 on Matlab : Question 3

Bivariate Normal Distribution with $\mu = (0 \ \ 0)'$, *and* $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

Using MCMC to draw from the Bivariate Normal Distribution

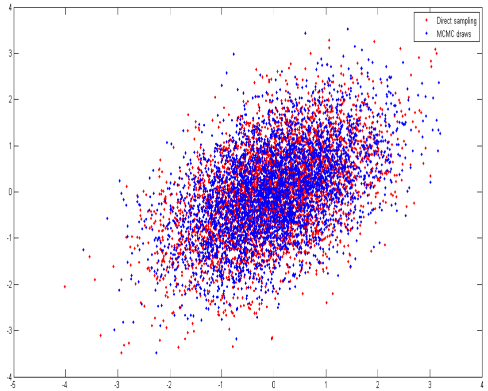
1. $\theta_1|\theta_2 \sim N(\rho\theta_2, \sqrt{1-\rho^2})$
2. $\theta_2|\theta_1 \sim N(\rho\theta_1, \sqrt{1-\rho^2})$

Matlab Code :

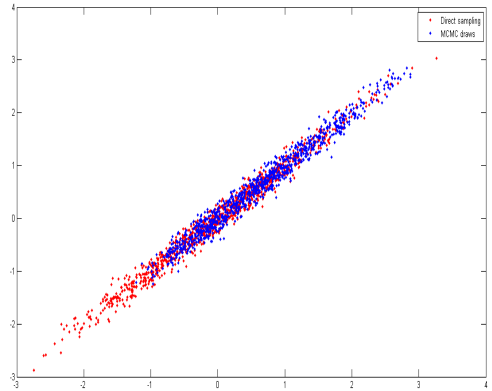
```
for i=1:nb_MCMC
    if(i>burn_in)
        post_theta(iter,:) = theta';
        iter = iter +1;
    end
    theta(1) = rho*theta(2) + sqrt((1-rho^2))*randn();
    theta(2) = rho*theta(1) + sqrt((1-rho^2))*randn();
end
```

Bivariate Normal Distribution with $\mu = (0 \;\; 0)'$*, and* $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

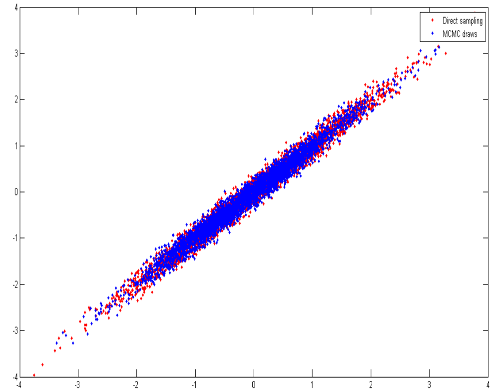
Using MCMC to draw from the Bivariate Normal Distribution



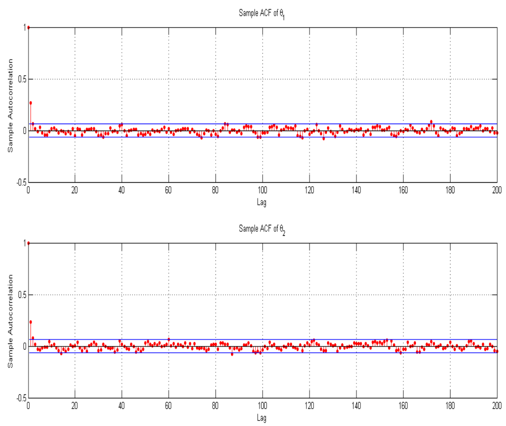
$\rho = 0.5$ and $N = 1000$



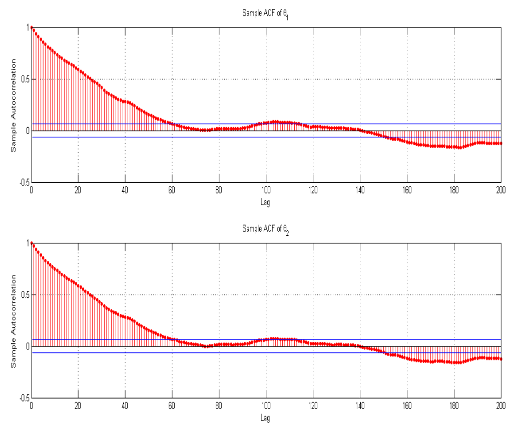
$\rho = 0.99$ and $N = 1000$



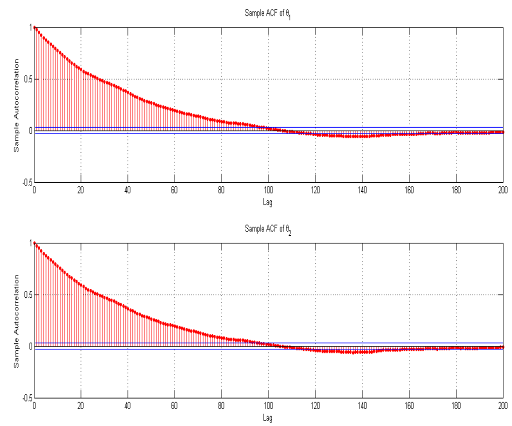
$\rho = 0.99$ and $N = 4000$



ACF $\rho = 0.5$



ACF $\rho = 0.99$



ACF $\rho = 0.99$