

auto regressive met code

Gian Marco Silvestri

28/8/2017

Mixture of Autoregressive Components

We used a mixture of two autoregressive process AR(1) and AR(2).

In general we have:

An autoregressive process $\{y_t\}$ of finite order ρ and mean μ is usually defined as:

$$y_t - \mu = \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) + \cdots + \phi_\rho(y_{t-\rho} - \mu) + \epsilon_t$$

where ϵ_t are i.i.d normal random variables with mean 0 and variance σ^2 .

If we denote:

$$\nu_t = \mu + \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) + \cdots + \phi_\rho(y_{t-\rho} - \mu) \quad (1)$$

the conditional distribution of y_t can be written as:

$$y_t | \mu, \phi, \rho, x_{t-1} \sim N(y_t | \nu_t, \sigma^2)$$

where $x_{t_1} = y_1, \dots, y_{t-1}$. Consider now k different AR process: each of them is characterized by a specific order ρ_j , a mean μ_j , a set of autoregressive parameters $\phi_j = (\phi_{1,j}, \dots, \phi_{\rho_j,j})$ and a variance of the error term σ_j^2 , for $j = 1, \dots, k$.

Suppose to adopt the following notation: $\rho = (\rho_1, \dots, \rho_k)$, $\mu = (\mu_1, \dots, \mu_k)$, $\sigma^2 = (\sigma_1^2, \dots, \sigma_k^2)$ and $\phi = (\phi_1, \dots, \phi_k)$.

The mixture of autoregressive components can be defined by:

$$y_t | \psi, \rho, k, x_{t-1} \sim \sum_{j=1}^k \omega_j N(y_t | \nu_t, \sigma^2)$$

where $\psi = (\omega, \mu, \sigma^2, \phi)$ and $\omega = (\omega_1, \dots, \omega_k)$ and:

$$\nu_j, t = \mu_j + \phi_{1,j}(y_{t-1} - \mu_j) + \phi_{2,j}(y_{t-2} - \mu_j) + \cdots + \phi_{\rho,j}(y_{t-\rho} - \mu_j)$$

for $j = 1, \dots, k$.

The mixing weights ω_j satisfy the usual constraints, i.e. $\omega_j > 0$ for $j = 1, \dots, k$ and $\sum_{j=1}^k \omega_j = 1$.

We set the following prior distributions for the parameters ω, μ, σ^2 :

$$\begin{aligned} \omega | k &\sim Dir(\omega | \delta, \dots, \delta) \\ \mu_j &\sim N(\mu_j | \mu_0, \tau^2) \quad j = 1, \dots, k \\ \sigma_j^2 &\sim Ig(\sigma^2 | \alpha, \beta) \quad j = 1, \dots, k \end{aligned} \quad (2)$$

with $\delta, \mu_0, \tau^2, \alpha$ and β assumed known.

Let $R = \max(x_T) - \min(x_T)$; following Richardson and Green (1997), we choose $\mu_0 = \min(x_T) + R/2$ and $\tau = cR$.

In particular the knowledge of the range of the data could be useful in setting the hyperparameters of σ^2 . In particular, β will be a multiple of $1/R^2$.

Suppose now the observable sample consist of T variables $x_T = (y_1, \dots, y_T)$. We shall consider the allocation variable $z = (z_1, \dots, z_T)$, where $z_t = j$ if the t-th observation comes from the j-th component, for $t = 1, \dots, T$ and $j = 1, \dots, k$.

Assuming that there are ρ_{\max} fixed observations before x_T the conditional likelihood is given by:

$$L(\theta|z) = \prod_{j=1}^k \prod_{t:z_t=j} N(y_t|\nu_{j,t}, \sigma_j^2)$$

for $t = 1, \dots, T$ and $\theta = (\mu, \sigma^2, \phi)$.

Let Φ be the stazionarity region for the j-th autoregressive model. Luckily, Barndorff-Nielsen and Schou (1973) pro- posed a reparametrization in terms of partial autocorrelations. Let $\pi_{h,j}$ be the partial autocorrelation coefficient at lag h for the j-th model. It is possible to show that $- < \pi_{h,j} < 1$, for $h = 1, \dots, \rho_j$ and $\pi_{h,j} = 0$ for $h > \rho_j$, $\rho_{1,j}$ is the first autocorrelation coefficient.

For a given component j, suppose now to construct an instrumental variable φ in the following recursive way:

$$\begin{aligned} \varphi_{1,1} &= \pi_{1,j} \\ \varphi_{h,m} &= \varphi_{h-1,m} - \varphi_{h,h}\varphi_{h-1,h-m} \quad m = 1, \dots, h-1 \\ \varphi_{h,h} &= \pi_{h,j} \end{aligned} \tag{3}$$

Finally, Barndorff-Nielsen and Schou (1973) proved that:

$$\phi_{h,j} = \varphi_{\rho_j,h} \quad h = 1, \dots, \rho_j$$

Hence, we established a very useful one to one transformation between $\phi_j = (\phi_{1,j}, \dots, \phi_{\rho_j,j})$ and $\pi_j = (\pi_{1,j}, \dots, \pi_{\rho_j,j})$.

In our case we have:

- $\rho_j = 1$
 $\phi_{1,j} = \pi_{1,j}$
- $\rho_j = 2$
 $\phi_{1,j} = \pi_{1,j}(1 - \pi_{2,j})$
 $\phi_{2,j} = \pi_{2,j}$

The prior specification is made in a straightforward way be- cause of the following result (Jones, 1987):

$$\phi_j \sim Unif \text{ on } \Phi_j \iff \pi_{i,j} \sim Beta_{[-1,1]} \left(\pi_{i,j} \middle| \left[\frac{i+1}{2} \right], \left[\frac{i}{2} + 1 \right] \right)$$

for $i = 1, \dots, \rho_j$ and $j = 1, \dots, k$ where $Beta_{[-1,1]}$ denotes a generalized beta distribution defined on $(-1, 1)$ and Φ_j is such that $\phi_j \in \Phi_j \iff |\pi_{h,j}| < 1$.

We implemented a Metropolis within Gibbs to estimate $\omega, \mu, \pi, \sigma^2, z$. In the Gibbs steps we used the following full conditonals:

- for ω

$$\begin{aligned}
\pi(\omega | \dots) &\propto L(\mu, \phi, \sigma^2, \beta, z, \rho) \pi(z|\omega) \pi(\omega) \pi(\mu) \pi(\phi|\rho) \pi(\rho) \pi(\sigma^2|\beta) \pi(\beta) \\
&\propto \pi(z|\omega) \pi(\omega) \\
&\propto \prod_{j=1}^k \omega_j^{n_j} \prod_{j=1}^k \omega_j^{\delta_j - 1} \\
&= Dir(\omega | \delta_1 + n_1, \delta_2 + n_2)
\end{aligned} \tag{4}$$

where $n_j = \sum_{t=1}^k I_{z_t=j}$ for $j = 1, \dots, k$

- for μ

$$\begin{aligned}
\pi(\mu | \dots) &\propto L(\mu, \phi, \sigma^2, \beta, z, \rho) \pi(\mu) \\
&= \prod_{j=1}^k \prod_{t:z_t=j} N(y_t | \nu_{j,t}, \sigma_j^2) \prod_{j=1}^k N(\mu_j | \mu_0, \tau^2)
\end{aligned} \tag{5}$$

Hence:

$$\pi(\mu_j | \dots) \propto \left[\prod_{t:z_t=j} N(y_t | \nu_{j,t}, \sigma_j^2) \right] N(\mu_j | \mu_0, \tau^2) \tag{6}$$

Writing the normal density function and substituting the (1), we obtain:

$$\pi(\mu_j | \dots) \propto \exp \left\{ -\frac{1}{2\sigma_j^2} \sum_{t:z_t=j} [y_t - \mu_j - \phi_{1,j}(y_{t-1} - \mu_j) - \dots - \phi_{\rho_j,j}(y_{t-\rho_j} - \mu_j)]^2 - \frac{1}{2\tau^2} (\mu_j - \mu_0)^2 \right\} \tag{7}$$

Letting $v_{t,j} = y_t - \phi_{j,1}y_{t-1} - \dots - \phi_{j,\rho_j}y_{t-\rho_j}$ and $B = 1 - \phi_{j,1} - \dots - \phi_{j,\rho_j}$,

$$\begin{aligned}
\pi(\mu_j | \dots) &\propto \exp \left\{ -\frac{1}{2\sigma_j^2} \sum_{t:z_t=j} (v_{t,j} - \mu_j B)^2 - \frac{1}{2\tau^2} (\mu_j - \mu_0)^2 \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_j^2} \sum_{t:z_t=j} (v_{t,j} - \bar{v}_j + \bar{v}_j - \mu_j B)^2 - \frac{1}{2\tau^2} (\mu_j - \mu_0)^2 \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_j^2} \sum_{t:z_t=j} [(v_{t,j} - \bar{v}_j)^2 + n_j (\bar{v}_j - \mu_j B)^2] - \frac{1}{2\tau^2} (\mu_j - \mu_0)^2 \right\} \\
&\propto \exp \left\{ -\frac{1}{2\sigma_j^2} n_j (\bar{v}_j - \mu_j B)^2 - \frac{1}{2\tau^2} (\mu_j - \mu_0)^2 \right\} \\
&\propto \exp \left\{ \frac{1}{2} \frac{n_j B^2 \tau^2 + \sigma_j^2}{\sigma_j^2 \tau^2} \left(\mu_j - \frac{n_j \bar{v}_j B \tau^2 + \sigma_j^2 \mu_0}{n_j B^2 \tau^2 + \sigma_j^2} \right)^2 \right\} \\
&= N \left(\mu_j \left| \frac{n_j \bar{v}_j B \tau^2 + \sigma_j^2 \mu_0}{n_j B^2 \tau^2 + \sigma_j^2}, \frac{\sigma_j^2 \tau^2}{n_j B^2 \tau^2 + \sigma_j^2} \right. \right)
\end{aligned} \tag{8}$$

- for σ^2

$$\begin{aligned}\pi(\sigma^2 | \dots) &\propto L(\mu, \phi, \sigma^2, \beta, z, \rho) \pi(\sigma^2 | \beta) \\ &= \prod_{j=1}^k \prod_{t:z_t=j} N(y_t | \nu_{j,t}, \sigma_j^2) \prod_{j=1}^k Ig(\sigma_j^2 | \alpha, \beta)\end{aligned}\tag{9}$$

Hence:

$$\begin{aligned}\pi(\sigma_j^2 | \dots) &\propto \left[\prod_{t:z_t=j} N(y_t | \nu_{j,t}, \sigma_j^2) \right] Ig(\sigma_j^2 | \alpha, \beta) \\ &\propto \sigma_j^{-n_j} \exp \left\{ -\frac{1}{2} \sum_{t:z_t=j} \frac{(y_t - \nu_{j,t})^2}{\sigma_j^2} \right\} \sigma_j^{-2(\alpha-1)} \exp \left\{ -\frac{\beta}{\sigma_j^2} \right\} \\ &= \sigma_j^{-2(n_j/2 + \alpha - 1)} \exp \left\{ \left[-\frac{1}{2} \sum_{t:z_t=j} (y_t - \nu_{j,t})^2 \right] / \sigma_j^2 \right\} \\ &= Ig \left(\sigma_j^2 \middle| \alpha + \frac{1}{2} n_j, \beta + \frac{1}{2} \sum_{t:z_t=j} (y_t - \nu_{j,t})^2 \right)\end{aligned}\tag{10}$$

- For z :

$$\begin{aligned}\pi(z | \omega, \mu, \sigma^2, y) &\propto L(\mu, \sigma^2, z) * \pi(z | \omega) \\ &= \prod_{i=1}^n \sum_{j=1}^k \omega_j N(y_i | \mu_j, \sigma_j^2) * I_{z_i=j}\end{aligned}\tag{11}$$

Hence:

$$\begin{aligned}\pi(z_i | \omega, \mu, \sigma^2, y) &\propto \sum_{j=1}^k \omega_j N(y_i | \mu_j, \sigma_j^2) * I_{z_i=j} \\ &\propto \sum_{j=1}^k \left[\frac{\omega_j}{\sigma_j} \exp \left\{ \frac{(y_i - \mu_j)^2}{2\sigma_j^2} \right\} I_{z_i=j} \right]\end{aligned}\tag{12}$$

We used the Metropolis-Hastings to update π_j in the following way. A candidate π_j^* is generated by a normal density truncated in $[-1, +1]$ and centered in the current state of the chain π_j :

$$q(\pi_j, \pi_j^*) = N_{[-1,1]}(\pi_j^* | \pi_j, \sigma_q^2) \quad \text{for } i = 1, 2$$

The acceptance probability is $\min(1, R)$ where can be represented as:

$$(\text{Likelihood ratio}) \times (\text{Prior ratio}) \times (\text{Proposal ratio})$$

The likelihood ratio is given by:

$$\text{Likelihood ratio} = \frac{\prod_{j=1}^k \prod_{t:z_t=j} N(y_t | \nu_{j,t}^*, \sigma_j^2)}{\prod_{j=1}^k \prod_{t:z_t=j} N(y_t | \nu_{j,t}, \sigma_j^2)}$$

The prior ratio is given by the ratio of two generalized beta:

$$\text{Prior ratio} = \frac{\prod_{i=1}^{\rho_j} Beta_{[-1,1]} \left(\pi_i^* \middle| \left[\frac{i+1}{2} \right], \left[\frac{i}{2} + 1 \right] \right)}{\prod_{i=1}^{\rho_j} Beta_{[-1,1]} \left(\pi_i \middle| \left[\frac{i+1}{2} \right], \left[\frac{i}{2} + 1 \right] \right)}$$

The proposal ratio is given by:

$$\begin{aligned} \text{Proposal ratio} &= \frac{\prod_{i=1}^{\rho_j} N_{[-1,1]}(\pi_{i,j} | \pi_{j,i}^*, \sigma_q^2)}{\prod_{i=1}^{\rho_j} N_{[-1,1]}(\pi_{i,j}^* | \pi_{j,i}, \sigma_q^2)} \\ &= \prod_{i=1}^{\rho_j} \frac{F_N(1 | \pi_{j,i}, \sigma_q^2) - F_N(-1 | \pi_{j,i}, \sigma_q^2)}{F_N(1 | \pi_{j,i}^*, \sigma_q^2) - F_N(-1 | \pi_{j,i}^*, \sigma_q^2)} \end{aligned} \quad (13)$$

where F_N is the normal cumulative distribution function.

```
library(truncnorm)
library(MCMCpack)

prob_z1 = function(p1,p2,s1,s2,x,ni1,ni2){
  out = p1/s1*exp(-1/2*((x-ni1)^2)/s1^2)/(p1/s1*exp(-1/2*((x-ni1)^2)/s1^2)+p2/s2*exp(-1/2*((x-ni2)^2)/s2^2))
  return(out)
}

M=50000
y = arima.sim(list(ar = 0.5) ,n=1000)
prior_d = 1
sam = c(10,50,100,500,1000)
weig=matrix(NA,nrow=M,ncol=5)
rhof=c(NA,5)
med=matrix(NA,nrow=M,ncol=5)

for(t in 1:5){
  y_2 = y[1:sam[t]]
  n = length(y_2)

  # Initial Values
  w = matrix(0.5,nrow = M, ncol = 2)
  mu_1 = rep(1,M)
  mu_2 = rep(1,M)
  s_1 = rep(NA,M)
  s_2 = rep(NA,M)
  ni_1 = rep(0.5,n)
  ni_2 = rep(0.5,n)
  ni_1s= rep(0.5,n)
```

```

ni_2s=rep(0.5,n)
v_1 = rep(50,n)
v_2 = rep(50,n)
pi_11 = rep(0.5,M)
pi_22 = rep(0.5,M)
phi_11 = rep(0.5,M)
phi_12 = rep(0.5,M)
phi_22 = rep(0.4,M)
rho1=rep(NA,M-1)

z_r1 = sample(c(1,2), n,replace=TRUE, prob=c(0.5,0.5))
z = matrix(NA, nrow = M, ncol = n)
z[1,] = z_r1
n_1=sum(z[1,]==1)
n_2=sum(z[1,]==2)
v_1bar=mean(y_2)
v_2bar=mean(y_2)
B_1= 0.5
B_2= 0.5
R = max(y_2)-min(y_2)

tau=R
mu_0=min(y_2)+R/2
s_1[1]=var(y_2)
s_2[1]=var(y_2)
a = 2
b = 0.1*R^(-2)

m_1y = mean(y_2)
sq=0.1
d_1 = prior_d
d_2 = prior_d

#gibbs steps
for(i in 2:M){

  w[i,] = rdirichlet(1,c(d_1+n_1,d_2+n_2))

  mu_1[i] = rnorm(1,(n_1*v_1bar*B_1*tau^2+s_1[i-1]*mu_0)/
    (n_1*B_1^2*tau^2+s_1[i-1]),
    sqrt(s_1[i-1]*tau^2/(n_1*B_1^2*tau^2 + s_1[i-1])))

  mu_2[i] = rnorm(1,(n_2*v_2bar*B_2*tau^2+s_2[i-1]*mu_0)/
    (n_2*B_2^2*tau^2+s_2[i-1]),
    sqrt(s_2[i-1]*tau^2/(n_2*B_2^2*tau^2 + s_2[i-1])))

  s_1[i] = rinvgamma(1,a+1/2*n_1, b + 1/2*sum((y_2[z[i-1],]==1] - ni_1[z[i-1],]==1])^2)
  s_2[i] = rinvgamma(1,a+1/2*n_2, b + 1/2*sum((y_2[z[i-1],]==2] - ni_2[z[i-1],]==2])^2)
}

```

```

for(j in 1:(n)){
  prob = prob_z1(w[i,1],w[i,2],sqrt(s_1[i]),sqrt(s_2[i]),y_2[j],ni_1[j],ni_2[j])
  z[i,j] = sample(c(1,2),1,replace = TRUE,prob=c(prob,1-prob))
}

n_1=sum(z[i,]==1)
n_2=sum(z[i,]==2)
B_1=1-phi_11[i-1]
B_2=1-phi_12[i-1]-phi_22[i-1]

#metropolis steps
prop_11 = rtruncnorm(1,-1,1,phi_11[i-1],sq)
prop_22 = rtruncnorm(1,-1,1,phi_22[i-1],sq)

phi11 = prop_11
phi12 = (prop_11*(1-prop_22))
phi22 = prop_22

ni_1s[1] = mu_1[i] + phi11*(m_1y-mu_1[i])
ni_2s[1] = mu_2[i] + phi12*(m_1y-mu_2[i])+ phi22*(m_1y-mu_2[i])
ni_1[1] = mu_1[i] + phi_11[i-1]*(m_1y-mu_1[i])
ni_2[1] = mu_2[i] + phi_12[i-1]*(m_1y-mu_2[i])+ phi_22[i-1]*(m_1y-mu_2[i])
v_1[1]=y_2[1]-phi_11[i-1]*m_1y
v_2[1]=y_2[1]-phi_12[i-1]*m_1y-phi_22[i-1]*m_1y

ni_1s[2] = mu_1[i] + phi11*(y_2[1]-mu_1[i])
ni_2s[2] = mu_2[i] + phi12*(y_2[1]-mu_2[i])+ phi22*(m_1y-mu_2[i])
ni_1[2] = mu_1[i] + phi_11[i-1]*(y_2[1]-mu_1[i])
ni_2[2] = mu_2[i] + phi_12[i-1]*(y_2[1]-mu_2[i])+ phi_22[i-1]*(m_1y-mu_2[i])
v_1[2]=y_2[2]-phi_11[i-1]*y_2[1]
v_2[2]=y_2[2]-phi_12[i-1]*y_2[1] -phi_22[i-1]*m_1y

for(j in 3:n){
  ni_1s[j] = mu_1[i] + phi11*(y_2[j-1]-mu_1[i])
  ni_2s[j] = mu_2[i] + phi12*(y_2[j-1]-mu_2[i])+ phi22*(y_2[j-2]-mu_2[i])
  ni_1[j] = mu_1[i] + phi_11[i-1]*(y_2[j-1]-mu_1[i])
  ni_2[j] = mu_2[i] + phi_12[i-1]*(y_2[j-1]-mu_2[i])+ phi_22[i-1]*(y_2[j-2]-mu_2[i])
  v_1[j]=y_2[j]-phi_11[i-1]*y_2[j-1]
  v_2[j]=y_2[j]-phi_12[i-1]*y_2[j-1]-phi_22[i-1]*y_2[j-2]
}

if(n_1==0){v1_bar=0}else{
  v_1bar=sum(v_1[z[i,]==1])/n_1
}
if(n_2==0){v2_bar=0}else{
  v_2bar=sum(v_2[z[i,]==2])/n_2
}

cond =
#lik ratio
(-1/(2*s_1[i])*sum(((y_2[z[i,]==1]-ni_1s[z[i,]==1])-  

(y_2[z[i,]==1]-ni_1[z[i,]==1]))^2)+1/(2*s_2[i])*  

sum(((y_2[z[i,]==2]-ni_2s[z[i,]==2])-  

(y_2[z[i,]==2]-ni_2[z[i,]==2]))^2))+
```

```

#prior ratio
round((1)/2)*log(prop_22+1)+round((2)/2)*log(1-prop_22)-
round((1)/2)*log(pi_22[i-1]+1)-round((2)/2)*log(1-pi_22[i-1])+

#proposal ratio
log(pnorm(1,pi_11[i-1],sq)-pnorm(-1,pi_11[i-1],sq))-log(pnorm(1,prop_11,sq)-pnorm(-1,prop_11,sq))+log(pnorm(1,pi_22[i-1],sq)-pnorm(-1,pi_22[i-1],sq))-log(pnorm(1,prop_22,sq)-pnorm(-1,prop_22,sq))

if(is.finite(cond)){
  acc = min(1,exp(cond))
}else{acc=1}

rho = runif(1)<acc
rho1[i-1]=runif(1)<acc
pi_11[i] = rho*prop_11 + (1-rho)*pi_11[i-1]
pi_22[i] = rho*prop_22 + (1-rho)*pi_22[i-1]

phi_11[i] = pi_11[i]
phi_12[i] = (pi_11[i]*(1-pi_22[i]))
phi_22[i] = pi_22[i]
}

weig[,t]=w[,1]

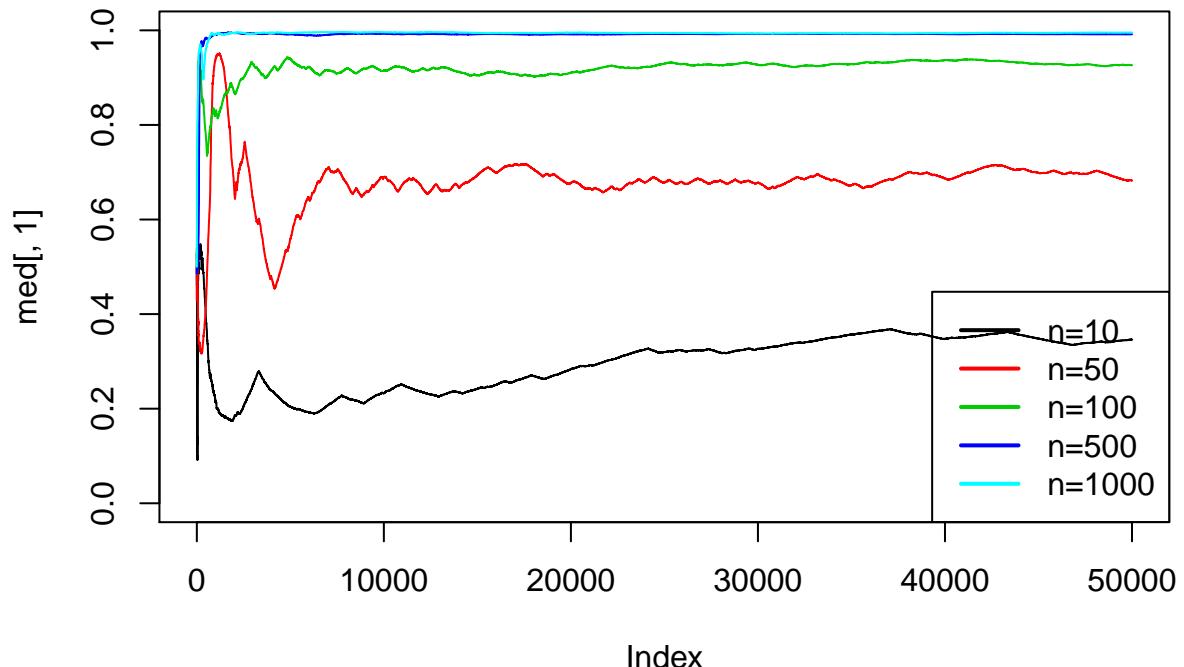
rhof[t]=sum(rho1)/M

for(j in 1:M){
  med[j,t]=median(w[1:j,1])
}
}

# Graphic Diagnostic
plot(med[,1],ty="l", main = "Posterior medians of the true model weight", ylim=c(0,1))
for(i in 2:5{
  lines(med[,i], col = i)
}
legend("bottomright", legend = c("n=10","n=50","n=100","n=500","n=1000"),
       col = c(1,2,3,4,5), lty = 1, lwd = 2)

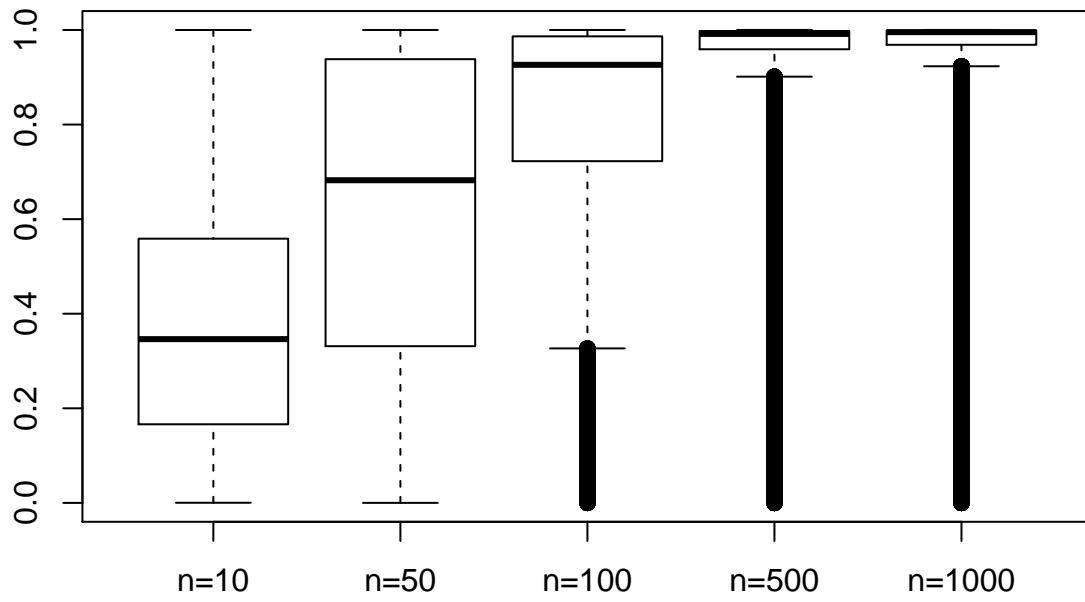
```

Posterior medians of the true model weight



```
boxplot(weig, names=c("n=10", "n=50", "n=100", "n=500", "n=1000") ,
       ylim = c(min(w[2:M,1]),1),
       main = "Boxplots of the posterior median for different n")
```

Boxplots of the posterior median for different n



```
detach("package:MCMCpack", unload = T)
library(invgamma)
```

```
# Compute the Bayes Factor
```

```

mod_1 = arima(y, order=c(1,0,0),include.mean = F)
vc_1 = mod_1$var.coef

mod_2 = arima(y, order=c(2,0,0),include.mean = F)
vc_2 = mod_2$var.coef

det(vc_1)

## [1] 0.0007806248
det(vc_2)

## [1] 7.79622e-07
n_BF = sam[t]

lik_1 = function(y,M,vc_1,mu_1,s_1,n){
  suma=matrix(NA,nrow = M, ncol = (n-1))

  for(i in 1:(n-1)){
    suma[,i]=((y[i+1]-mu_1)-phi_11*(y[i]-mu_1))^2
  }

  out = (-n/2)*log((2*pi*s_1))+log(det(vc_1))*  

    (1/2)+((-1/(2*s_1^2))*(vc_1[1,1]*(y[1]-mu_1)^2 +  

      apply(suma,1,sum)))
  return(out)
}

lik_2 = function(y,M,vc_2,mu_2,s_2,n){

  suma_21=matrix(NA,nrow = M, ncol = 4)
  k=1
  for(i in 1:2){
    for(j in 1:2){
      suma_21[,k]=vc_2[i,j]*(y[i]-mu_2)*(y[j]-mu_2)
      k=k+1
    }
  }

  suma_22 = matrix(NA,nrow = M, ncol = (n-2))

  for(i in 1:(n-2)){

    suma_22[,i]=((y[i+2]-mu_2)-phi_12*(y[i+1]-mu_2)-phi_22*(y[i]-mu_2))^2
  }
  out = (-n/2)*log(2*pi*s_2)+log(det(vc_2))*  

    (1/2)+(-1/(2*s_2^2)*(apply(suma_21,1,sum)+  

      apply(suma_22,1,sum)))
  return(out)
}

m_1 = cumsum(lik_1(y,M,vc_1,mu_1,s_1, n_BF)[2:M]+  

  dnorm(mu_1,mu_0,tau,log=T)[2:M]+log(dinvgamma(s_1,a,b))[2:M]-

```

```

dnorm(mu_1,(n_1*v_1bar*B_1*tau^2+s_1*mu_0)/(n_1*B_1^2*tau^2+s_1),
      sqrt(s_1*tau^2/(n_1*B_1^2*tau^2 + s_1)), log = T)[2:M] -
dinvgamma(s_1,shape=a+1/2*n_1,scale=b+1/2*
sum((y_2[z[M,]==1]-ni_1[z[M,]==1])^2),log=T)[2:M])/(1:(M-1))

m_2 = cumsum(lik_2(y,M,vc_2,mu_2,s_2, n_BF)[2:M]+
dnorm(mu_2,mu_0,tau,log=T)[2:M]+log(dinvgamma(s_2,a,b))[2:M]-
dnorm(mu_2,(n_2*v_2bar*B_2*tau^2+s_2*mu_0)/(n_2*B_2^2*tau^2+s_2),
      sqrt(s_2*tau^2/(n_2*B_2^2*tau^2 + s_2)), log = T)[2:M] -
dinvgamma(s_2,shape = a+1/2*n_2, scale = b + 1/2*
sum((y_2[z[M,]==2] - ni_2[z[M,]==2])^2),log=T)[2:M])/(1:(M-1))

# Bayes Factor
BF = mean((m_1)-(m_2))
BF

## [1] 36866694212

#Posterior probability of the model
pmodel=BF/(1+BF)
p_car = as.character(pmodel)
p_car

## [1] "0.999999999972875"

```