

Catbond paper by Saeid Bayesian models

Arpita Chatterjee

2023-02-25

Model 1 (as in Equation 5.39)

Creating data for Bayesian model

```
x <- c(1290, 2062, 1, 1033, 173, 620, 1832, 1, 551, 2261, 952,
12, 259, 556,
847, 408, 136, 1026, 528, 45, 1108, 219, 635, 341, 32, 548, 303,
35,
223, 49, 502, 13, 879, 595, 25, 488, 576, 397, 269, 483, 865,
495,
743, 64, 701, 126, 120, 988, 358, 70, 101, 115, 7, 63, 104,
169,
736, 71, 108, 19, 160, 642, 163, 5, 8, 437, 54, 580, 114,
133,
485, 302, 64, 270, 180, 118, 343, 2078, 178, 613, 414, 217, 64 )
data <- list(w=x, m=length(x))
```

Creating initials for the Bayesian model

```
init <- function(){
  list(xi=c(0.1, 0.1))
}
```

Let's write the gamma model

```
gammodel <- function() {
  for (i in 1:m){
    w[i] ~ dgamma(xi[i], xi[i])
  }
  dummy <- 0
  dummy ~ dloglik(phi)
  phi <- log(xi[i]) - loggam(xi[i]) + ((xi[i]-1)*(log(xi[i]) -
(1/(2*xi[i]))))/exp(loggam(xi[i])) - xi[i])
  xi[i] ~ duniform(0.01, 1000)
  xi[i] ~ duniform(0.01, 1000)

  a <- xi[i]/xi[i] #This line will evaluate E(xi1/xi2|W)
  # b <- (log(xi[i]) - (1/(2*xi[i]))) - log(xi[i]) #This line will evaluate
E(psi(xi1) - log(xi2) | W)

  b <- (log(xi[i]+1) - (1/(2*(xi[i]+1)))) - (log(xi[i]) - (1/(2*xi[i])))
}
```

Next we'll execute the model

```
bayesout1<- bugs(data, ini ts, model .file = gammamodel ,
                 parameters = c("xi", "a", "b"), n.chai ns = 1, n.burni n=10000,
n.i ter = 40000,DIC=FALSE, codaPkg=TRUE, di gi ts=3, debug=F)
```

Formatting the posterior samples

```
out <- read.bugs(bayesout1)
```

```
## Abstracting a ... 30000 valid values
## Abstracting b ... 30000 valid values
## Abstracting xi [1] ... 30000 valid values
## Abstracting xi [2] ... 30000 valid values
```

out[[1]] is a matrix of posterior samples: (1) column 1 gives the samples of ξ_1/ξ_2 , (2) column 2 gives the samples of $\psi(\xi_1 + 1) - \psi(\xi_1)$, (3) column 3 gives the sample of ξ_1 , and (4) column 4 gives the samples of ξ_2 .

To obtain the two posterior expected values, $\mathbb{E}^{\mathbb{P}} \left[\frac{\xi_1}{\xi_2} \mid \mathbf{w} \right]$ and $\mathbb{E}^{\mathbb{P}} [\psi(\xi_1 + 1) - \psi(\xi_1) \mid \mathbf{w}]$, consider the column 1 and column 2 averages respectively.

We obtain the posterior estimates of ξ_1 and ξ_2 as,

$$\hat{\xi}_1 = \frac{1}{\mathbb{E}^{\mathbb{P}} [\psi(\xi_1 + 1) - \psi(\xi_1) \mid \mathbf{w}]}$$

$$\hat{\xi}_2 = \frac{\hat{\xi}_1}{\mathbb{E}^{\mathbb{P}} \left[\frac{\xi_1}{\xi_2} \mid \mathbf{w} \right]}$$
 and

```
xi 1. hat <- 1/(mean(out[[1]][, 2]))
xi 2. hat <- xi 1. hat/mean(out[[1]][, 1])
```

```
xi 1. hat
```

```
## [1] 2.473875
```

```
xi 2. hat
```

```
## [1] 0.01019517
```

To check the model convergence, try running multiple chains.

```
ini ts <- function(){
  list(xi=c(0.1, 0.1))
  list(xi=c(2, 2))
  list(xi=c(0.01, 0.01))}
```

```
bayesout_3chai ns<- bugs(data, ini ts, model .file = gammamodel ,
                        parameters = c("xi", "a", "b"), n.chai ns = 3, n.burni n=5000,
```

```
n.iter = 15000, DIC=FALSE, codaPkg=TRUE, digits=3, debug=F)
```

Formatting the posterior samples

```
out_3chains <- read.bugs(bayesout_3chains)

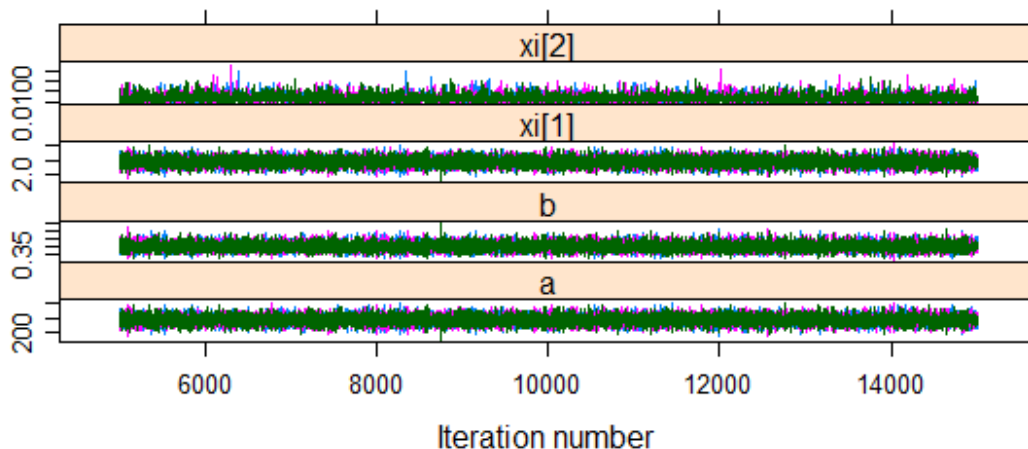
## Abstracting a ... 10000 valid values
## Abstracting b ... 10000 valid values
## Abstracting xi[1] ... 10000 valid values
## Abstracting xi[2] ... 10000 valid values
## Abstracting a ... 10000 valid values
## Abstracting b ... 10000 valid values
## Abstracting xi[1] ... 10000 valid values
## Abstracting xi[2] ... 10000 valid values
## Abstracting a ... 10000 valid values
## Abstracting b ... 10000 valid values
## Abstracting xi[1] ... 10000 valid values
## Abstracting xi[2] ... 10000 valid values
```

Convergence diagnostics:

1. History plot

```
#library(coda)
#library(lattice)
```

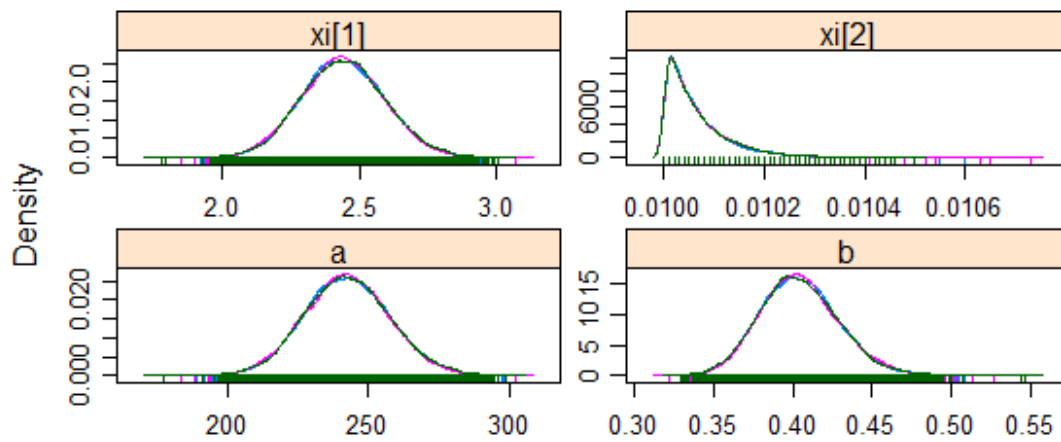
```
xyplot(out_3chains)
```



2.

Density plot:

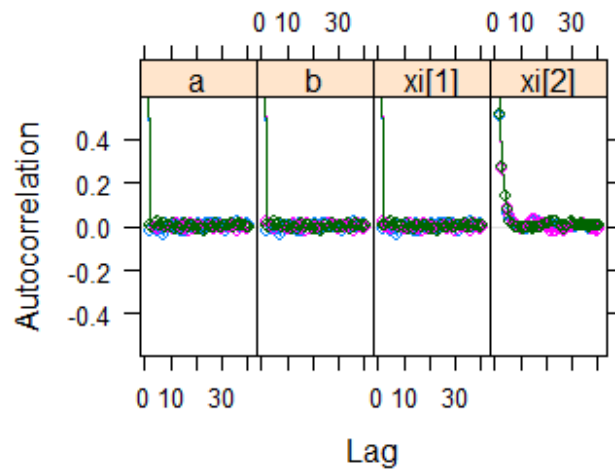
```
densi typlot(out_3chains)
```



3. ACF

plot

```
acfplot(out_3chains)
```



Model 2 (Mixture of Exponential)

```
mixexp <- function() {
  for (i in 1:m){
    gr[i] ~ dcat(p[])
    w[i] ~ dexp(lambda[gr[i]])
  }

  p[1:2] ~ ddirich(alpha[])
  for(k in 1:2){
    lambda[k] ~ dgamma(tau[k], psi[k])
  }
}
```

```
}
```

Creating data for Bayesian model

```
data <- list(w=x, m=length(x), tau=c(0.01, 0.01), psi=c(0.01, 0.01),  
alpha=c(0.5, 0.5))
```

Creating initials for the Bayesian model

```
inits <- function(){  
  list(lambda=c(0.1, 0.1))}
```

Let's execute the model

```
bayesout_mlx <- bugs(data, inits, model.file = mixexp,  
                    parameters = c("lambda"), n.chains = 1, n.burnin=10000, n.iter  
= 40000, DIC=FALSE, codaPkg=TRUE, digits=3, debug=F)
```

```
out_mlxexp <- read.bugs(bayesout_mlx)
```

```
## Abstracting lambda[1] ... 30000 valid values
```

```
## Abstracting lambda[2] ... 30000 valid values
```

Let's obtain the posterior estimates under the squared error loss

```
bayes.est <- colMeans(out_mlxexp[[1]])  
bayes.est
```

```
## lambda[1] lambda[2]
```

```
## 0.2140161 0.3399165
```

```
#95% credible interval
```

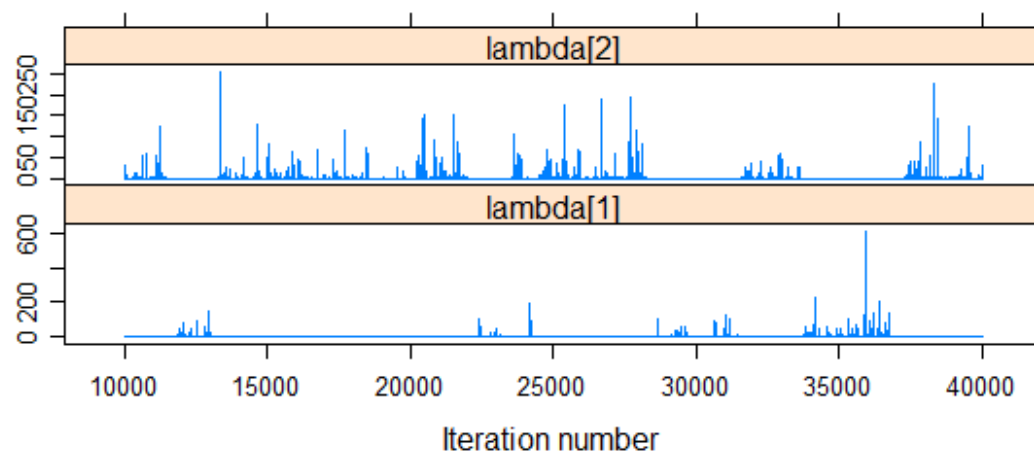
```
cred.int <- cbind(quantile(out_mlxexp[[1]][, 1], prob=c(0.025, 0.975)),  
quantile(out_mlxexp[[1]][, 2], prob=c(0.025, 0.975)))
```

Convergence diagnostics:

1. History plot

```
#library(coda)  
#library(lattice)
```

```
xyplot(out_mlxexp)
```



The model doesn't converge well.