ST5222-Assessment-2

Name:Zhu Xu
User ID:E0337988
Student ID:A0131944H

Nov 2018

Let the Poisson distribution with parameter $\lambda > 0$ be written $P(\lambda)$ and donate the corresponding density function as $\vartheta(x;\lambda)$.We consider data $x_1, \cdots, x_n$ assumed to be conditionally independent given parameters $\lambda_{1:k}$ and mixture proportions $\pi_{1:k-1}$ with probability density function:

$$p(x_i|\lambda_{1:k}, \pi_{1:k-1}) = \sum_{j=1}^{k} \pi_j \vartheta(x_i; \lambda_j) \qquad i \in \{1, \cdots, n\}$$

where $\pi_j > 0$, $\sum_{j=1}^{k-1} \pi_j \leq 1$. We consider the following Batesian model, with the prior as to be described. Independently for each $j \in \{1, \cdots, k\}$ and of all other parameters:

$$\lambda_j \sim Ga(\alpha, \beta)$$

Then, independent of all parameters $\pi_{1:k-1} \sim D(\delta)$.

Consider an additional n conditionally independent variables $z_i \in \{1, \cdots, k\}$ (z_i is the component of the mixture of which the i^{th} data-point belongs to.) and a joint density of the observations and missing data as:

$$p(x_{1:n}, z_{1:n}|\theta_{1:k}, \pi_{1:k-1}) = \prod_{i=1}^{n} f(x_i; \theta_{z_i}) \pi_{z_i}$$

Given a prior on $\theta_{1:k}, \pi_{1:k-1}$ (write it $p(\theta_{1:k}, \pi_{1:k-1})$), then the posterior is:

$$p(\theta_{1:k}, \pi_{1:k-1}, z_{1:n}|x_{1:n}) \propto p(z_{1:n}, x_{1:n}|\theta_{1:k}, \pi_{1:k-1}) p(\theta_{1:k}, \pi_{1:k-1})$$

We write the mixture model as:

$$p(x_i|\lambda_{1:k}, \pi_{1:k-1}) = \sum_{j=1}^{k} \pi_j \vartheta(x_i; \lambda_j) \qquad i \in \{1, \cdots, n\}$$

The prior structure is then, for $j \in \{1, \cdots, k\}$:

$$\lambda_j \sim Ga(\alpha, \beta)$$

($Ga(\alpha, \beta)$ is a gamma distribution with mean $\frac{\alpha}{\beta}$)

For the mixture weights:

$$\pi_{1:k-1} \sim D(\delta)$$

($D(\delta)$ is a symmetric dirichlet distribution)

$$p(\pi_{1:k-1}) = \frac{\Gamma(k\delta)}{\Gamma(\delta)^k} \prod_{j=1}^{k} \pi_j^{\delta-1} \qquad \pi_j \geq 0, \sum_{j=1}^{k-1} \pi_j \leq 1$$

For $j \in \{1, \cdots, k\}$, the number of data points that are "allocated" to component j:

$$n_j = \sum_{i=1}^{n} I_{\{j\}}(z_i)$$

Then we can compute the posterior for the $\lambda_{1:k}$:

Let $I_{\{j\}}(z_i) = c$

$$p(\lambda_{1:k}, x_{1:n}, z_{1:n}, \pi_{1:k-1}) = p(x_{1:n}, z_{1:n}|\lambda_{1:k}, \pi_{1:k-1})p(\lambda_{1:k}, \pi_{1:k-1})$$

$$p(\lambda_{1:k}|x_{1:n}, z_{1:n}, \pi_{1:k-1}) \propto [\prod_{i=1}^{n} \pi_{z_i}\vartheta(x_i|\lambda_{z_i})][\prod_{j=1}^{k} f(\lambda_j|\alpha,\beta)]p(\pi_{1:k-1})$$

$$p(\lambda_j|x_{1:n}, z_{1:n}, \pi_{1:k-1}) \propto \prod_{i=1}^{n}[\vartheta(x_i;\lambda_j)]^c f(\lambda_j|\alpha,\beta)$$

$$\Rightarrow \quad p(\lambda_j|x_{1:n}, z_{1:n}, \pi_{1:k-1}) \propto \prod_{i=1}^{n}[\tfrac{\lambda_j^{x_i}}{x_i!}e^{-\lambda_j\pi_j}]^c \lambda_j^{\alpha-1}\tfrac{\beta^\alpha e^{-\beta\lambda_j}}{\Gamma(\alpha)}$$

$$\Rightarrow \quad p(\lambda_j|x_{1:n}, z_{1:n}, \pi_{1:k-1}) \propto \lambda_j^{\sum_{i=1}^{n} x_i I_{\{j\}}(z_i)+\alpha-1} e^{-(\beta+n_j)\lambda_j}$$

Thus:

$$\lambda_j|\cdots \sim Ga(\textstyle\sum_{i=1}^{n} x_i I_{\{j\}}(z_i) + \alpha, \beta + n_j)$$

"$|\cdots$" denotes conditioning on all other variables.

For missing data:

$$p(z_{1:n}|x_{1:n}, \lambda_{1:k}, \pi_{1:k-1}) \propto [\prod_{i=1}^{n} \pi_{z_i}\vartheta(x_i|\lambda_{z_i})][\prod_{j=1}^{k} f(\lambda_j|\alpha,\beta)]p(\pi_{1:k-1})$$

$$\Rightarrow \quad p(z_i = j|x_{1:n}, \lambda_{1:k}, \pi_{1:k-1}) \propto \vartheta(x_i|\lambda_j)\pi_j$$

$$\Rightarrow \quad p(z_i = j|x_{1:n}, \lambda_{1:k}, \pi_{1:k-1}) \propto \pi_j\lambda_j^{x_i}e^{-\lambda_j}$$

For the weights:

$$p(\pi_{1:k-1}|x_{1:n}, \lambda_{1:k}, z_{1:n}) \propto [\prod_{i=1}^{n} \pi_{z_i}\vartheta(x_i|\lambda_{z_i})][\prod_{j=1}^{k} f(\lambda_j|\alpha,\beta)]p(\pi_{1:k-1})$$

$$\Rightarrow \quad p(\pi_{1:k-1}|x_{1:n}, \lambda_{1:k}, z_{1:n}) \propto \prod_{j=1}^{k-1}\pi_j^{n_j}p(\pi_{1:k-1})$$

$$\Rightarrow \quad p(\pi_{1:k-1}|x_{1:n}, \lambda_{1:k}, z_{1:n}) \propto \prod_{j=1}^{k-1}\pi_j^{n_j+\delta}$$

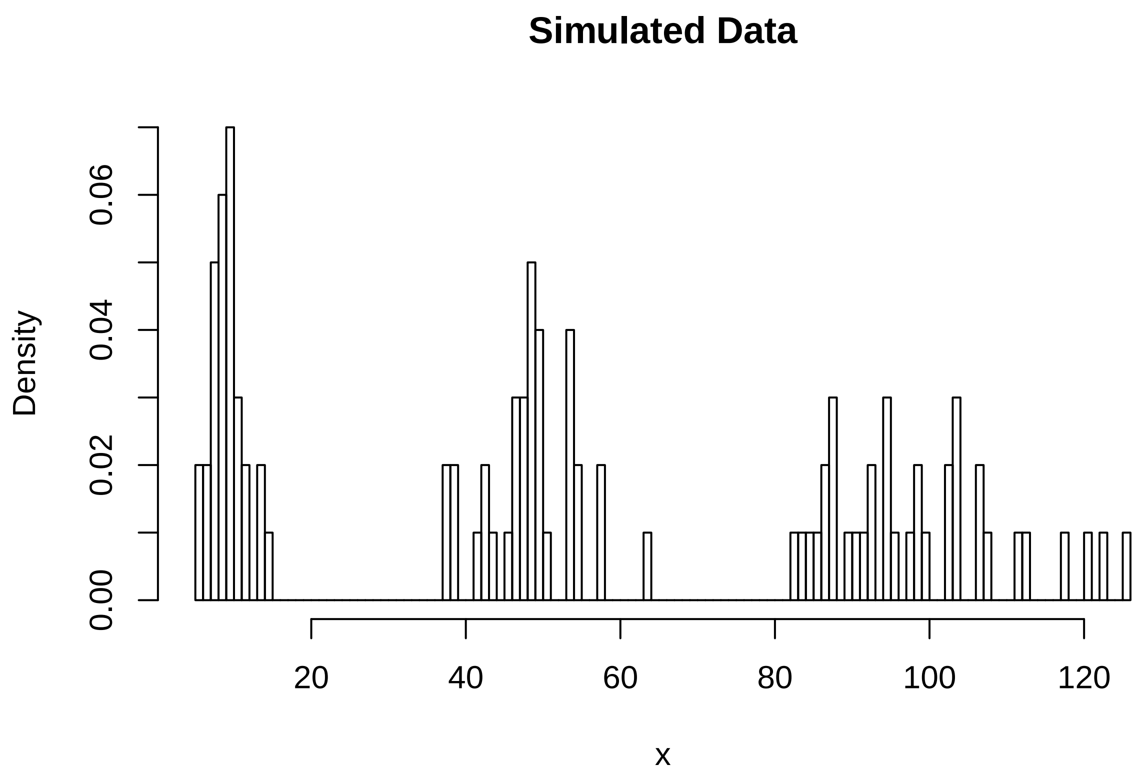
Thus:

$$\pi_{1:k-1}|\cdots \sim D(\delta + n_1^{(t-1)}, \cdots, \delta + n_k^{(t-1)})$$

The following R code will simulate the Gibbs sampler:

```
library(gtools)

k <- 3

Mixture <- function(lambda, n){

  # Weights
  set.seed(1)

  Pi <- as.numeric(rep(1/k,k))

  # simulate data from model

  z <- numeric(n)   # missing data

  x <- numeric(n)

  # sample an z according to Pi

  for(i in 1:n){

    #z[i] represents that data i belongs to z[i]=j component
    z[i] <- sample(c(1:k),1,replace = TRUE, prob = Pi)

    x[i] <- rpois(1, lambda[z[i]])
  }

  return(list(x, lambda, k, n, Pi))
}

Return_value <- Mixture(c(10,50,100), 100)

x <- Return_value[[1]]
```

```r
#plot data
hist(x, freq = F, nclass = 100, main="Simulated Data")
```



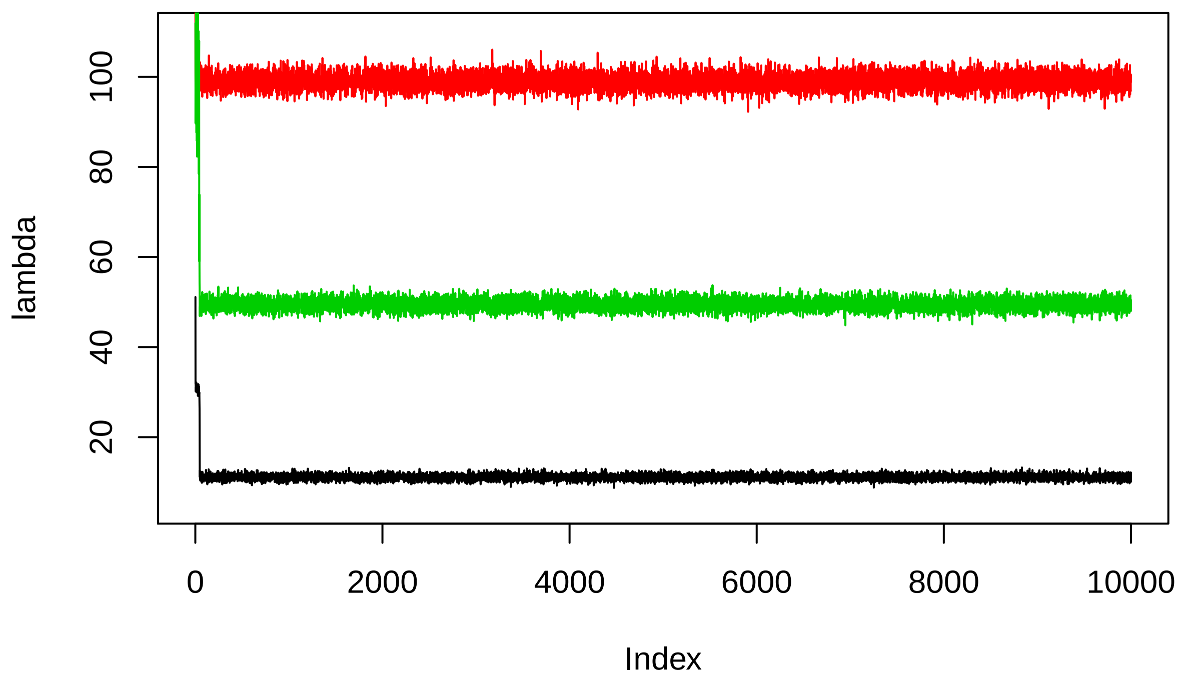
```r
Gibbs <- function(Return_value, N){

  x <- Return_value[[1]]

  k <- Return_value[[3]]

  n <- Return_value[[4]]

  # initiate lambda
  alpha <- 50; beta <- 0.5; set.seed(1)

  lambda <- as.numeric(rgamma(k, alpha, beta))

  lambda_0 <- lambda

  # initiate Pi(weights)
  set.seed(2)

  Pi <- as.numeric(rdirichlet(1, c(rep(1,k))))

  Pi_0 <- Pi

  # initiate z (missing data)
  updata_z <- function(seed){

    prob_zij <- matrix(rep(0,n*k),n,k)

    z <- numeric(n)

    for(i in c(1:n)){

      for(j in c(1:k)){

        #Pi[j]*exp(-lambda[j])*(lambda[j])^(x[i])
        prob_zij[i,j] <- Pi[j]*dpois(x[i],lambda[j])

      }

      for(j in c(1:k)){

        prob_zij[i,j] <- prob_zij[i,j]/sum(prob_zij[i,])
```

```
    }

    z[i] <- sample(c(1:k),1,replace=F,prob = prob_zij[i,])

  }

  return(z)
}

z <- updata_z(1)

# the number of data in each component
sumI_and_sumxI <- function(z){

  # the number of samples belong to class j
  sumI <- numeric(k)

  # sum all samples from class j
  sumxI <- numeric(k)

 for(j in 1:k){

    sumI[j] <- length(which(z==j))

    sumxI[j] <- sum(x[which(z==j)])
 }

  return(list(sumI,sumxI))
}

Result1 <- sumI_and_sumxI(z)

sumI <- Result1[[1]]

sumxI <- Result1[[2]]

# record simulation

lambdas <- matrix(rep(0,N*k),N,k)

Pis <- matrix(rep(0,N*k),N,k)
```

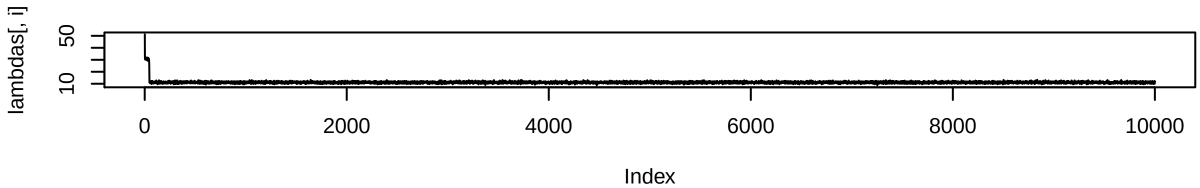
```r
# main loop
for(loop in 1:N){

  # update lambda

  for(j in 1:k){

    lambda[j] <- rgamma(1, alpha+sumxI[j], beta+sumI[j])
  }

  # update z

  z <- updata_z(1)

  Result1 <- sumI_and_sumxI(z)

  sumI <- Result1[[1]]

  sumxI <- Result1[[2]]

  # update weight Pi

  Pi <- as.numeric(rdirichlet(1, c(rep(1,k))+sumI))

  # record updated parameters

  lambdas[loop, ] <- lambda

  Pis[loop, ] <- Pi

  # break condition

  sigma <- 0

  # break loop or not

  if(loop != 1){

    J <- matrix(rep(0,2*k),2,k)

    for(j in 1:k){
```

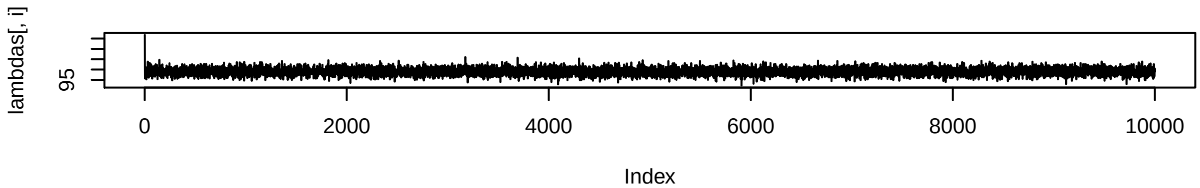
```r
        J[1,j] <- abs(lambdas[loop,j]-lambdas[(loop-1),j])<=sigma

        J[2,j] <- abs(Pis[loop,j]-Pis[(loop-1),j])<=sigma
      }

      len <- length(which(J == FALSE))

      if(len==0){break}
    }
  }

  Converged_Pi <- Pis[loop,]

  Converged_lambda <- lambdas[loop,]

  Pis <- Pis[c(1:loop),]

  lambdas <- lambdas[c(1:loop),]

  return(list(Converged_Pi, Converged_lambda, Pis, lambdas))

}

# steps of iteration
N <- 10000

Result <- Gibbs(Return_value, N)

Converge_Pi <- Result[[1]]

Converge_lambda <- Result[[2]]

Pis <- Result[[3]]

lambdas <- Result[[4]]
```

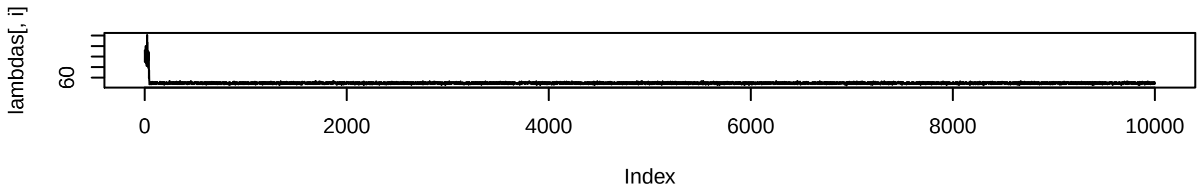
```
#plot samples
plot(lambdas[,1],type="l",ylim=c(5,110),ylab="lambda")

lines(lambdas[,2],col=2)

lines(lambdas[,3],col=3)
```



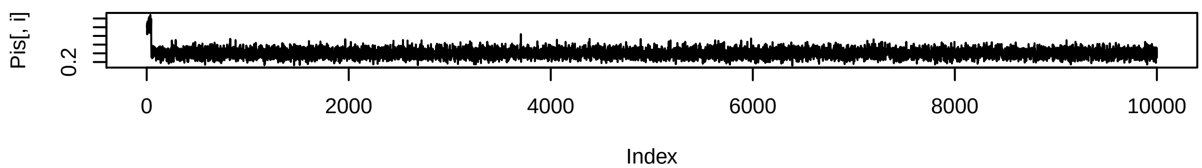
```
par(mfrow=c(k,1))

for(i in 1:k)
{
    plot(lambdas[,i],type="l")
}
```

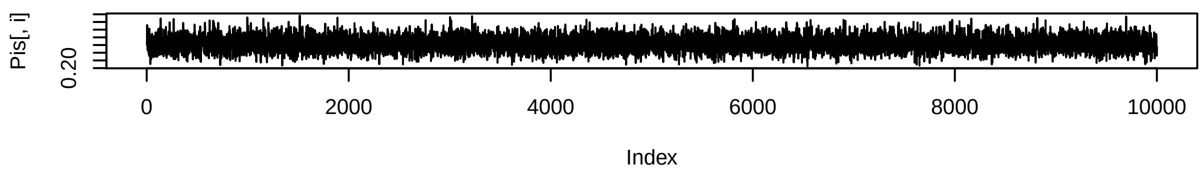


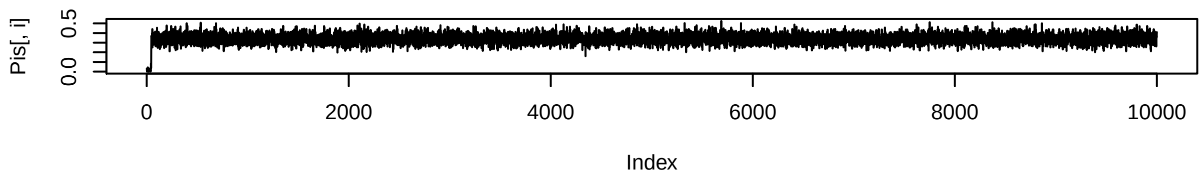




```
par(mfrow=c(k,1))

for(i in 1:k)
{
 plot(Pis[,i],type="l")
}
```







Conclusion:

We ran the Gibbs sampler on 100 simulated data, for 10000 iterations.From the output, we can see our observed data, which appear to correspond to 3 separated modes, including the posterior samples of the parameter λ (which should be located close to these modes), actually the output indicates that modes are located correctly.