STA601: Homework 9

Akbota Anuarbek

*****Halloween Costumes*****

$i = 1, ..., n$=type of costume
$j = 1, ..., n_i$=person wearing ith type
y_{ij}=count of the #pieces of candy fo ij
t_{ij}=time spent trick and trenting

Hierchical Model allowing variablity across types:

$y_{ij} \sim Poisson(\theta\lambda_i t_{ij})$
$\theta \sim Gamma(a, b)$
$\lambda_i \sim Gamma(\phi, \phi)$
$\phi \sim Gamma(c, d)$

Here:

$\{\lambda_i\}_{i=1}^n$ are random effect/latent variables
θ and ϕ are population parameters

$E(y_{ij}) = E(\theta\lambda_i t_{ij}) = t_{ij}\theta\frac{\phi}{\phi} = \theta t_{ij}$, i.e. expected # of candies collected by child ij is the time he/she spent trick&trenting multiplied by population mean

Gibbs sampler/Derive full conditionals:

$\pi(\theta|\lambda_{i:n}, y, t) \propto \prod_{i=1}^n \prod_{j=1}^{n_i} \frac{(\theta\lambda_i t_{ij})^{y_{ij}}}{y_{ij}!} \exp\left(-\theta\lambda_i t_{ij}\right)\theta^{a-1}\exp(-b\theta) \propto \theta^{a+\sum_{i=1}^n \sum_{j=1}^{n_i} y_{ij}-1} \exp\left[-\theta(b+\sum_{i=1}^n \sum_{j=1}^{n_i} \lambda_i t_{ij})\right] \propto Gamma(a + \sum_{i=1}^n \sum_{j=1}^{n_i} y_{ij}, b + \sum_{i=1}^n \sum_{j=1}^{n_i} \lambda_i t_{ij})$

$\pi(\lambda_i|\theta, y, t) \propto \prod_{j=1}^{n_i} (\lambda_i)^{y_{ij}} \exp\left(\lambda_i(\phi + \theta \sum_{j=1}^{n_i} t_{ij})\right) \propto Gamma(\phi + \sum_{j=1}^{n_i} y_{ij}, \phi + \theta \sum_{j=1}^{n_i} t_{ij})$

So will draw θ and λ_i's using Gibbs.

Metropolis algorithm

However, conditional for ϕ is not conjugate and we can't use Gibbs sampling to draw ϕ's. Instead we will use Metropolis Algorithm for drawing ϕ.

- Draw candidate $\phi^*|\phi^{(t-1)} \sim g(\phi^{(t-1)}) = Gamma(\phi^{(t-1)}\epsilon, \epsilon)$, where we let $\epsilon = 1$

- Accept the candidate, i.e. set $\phi^{(t)} = \phi^*$ with probability $= \min\left\{1, \frac{\pi(\theta,\lambda_{1:n},\phi^*|Y,t)}{\pi(\theta,\lambda_{1:n},\phi^{(t-1)}|Y,t)} \times \frac{g(\phi^{(t-1)}|\phi^*)}{g(\phi^*|\phi^{(t-1)})}\right\} = \min\left\{1, \frac{\sum_{i=1}^{n} Gamma(\lambda_i;\phi^*,\phi^*)\times Gamma(\phi^*;c,d)}{\sum_{i=1}^{n} Gamma(\lambda_i;\phi^{(t-1)},\phi^{(t-1)})\times Gamma(\phi^{(t-1)};c,d)} \times \frac{g(\phi^{(t-1)}|\phi^*)}{g(\phi^*|\phi^{(t-1)})}\right\}$

Otherwise, set $\phi^{(t)} = \phi^{(t-1)}$

Below using R we assume some parameter values, generate data, and use Gibbs and Metropolis algorithms to estimate parameters:

```
#assume hyperparameters have the following values

a=0.5
b=0.5
c=0.5
d=0.5


#randomly pick #of costumes and #of children wearing each type

n<-sample(20:50,1)
n_i<-sample(20:50,n, replace=TRUE)


#assume true parameter values

theta=10
phi=10
lambda<-rgamma(n,shape=f,rate=f)


#generate data, letting t_ij be measured in hours

t<-matrix(NA,n, max(n_i))
y<-matrix(NA,n, max(n_i))
for(i in 1:n){
  for(j in 1:n_i[i]){
    t[i,j]<-sample(1:5,1, replace=TRUE)
    y[i,j]<-rpois(1,theta*lambda[i]*t[i,j])
  }
}


#Sampling part
```

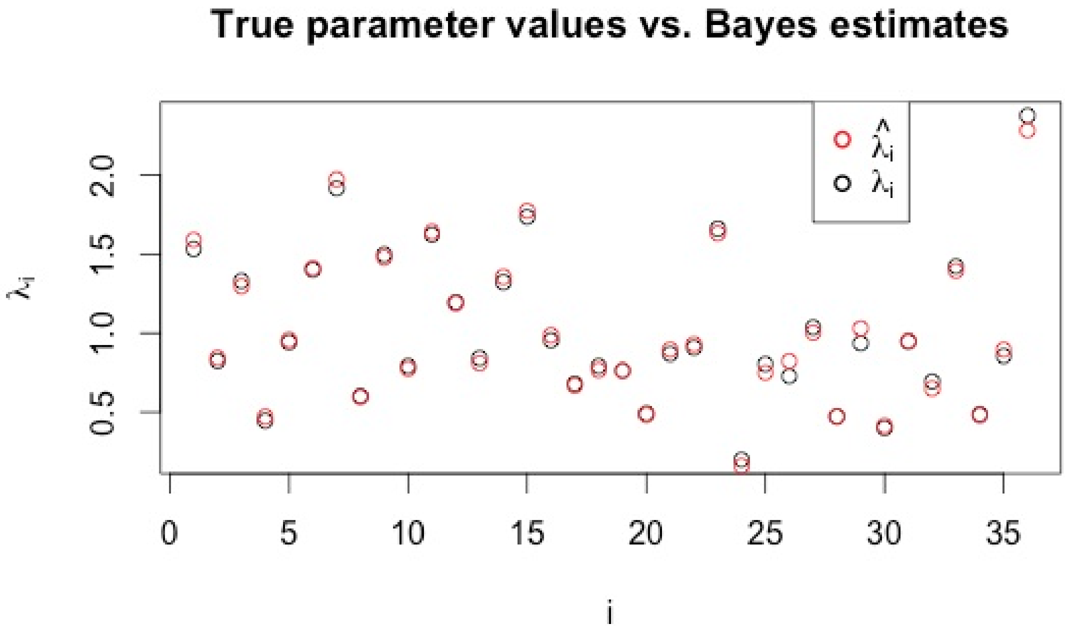
```
N<-10000
th<-c(1)
f<-c(5)
l<-matrix(NA,N,n)
l[1,]<-1
epsilon=1

for(k in 2:N){

  #update theta

  sum_y<-0
  sum_t<-0
  for(i in 1:n){
    for(j in 1:n_i[i]){
      sum_y=sum_y+y[i,j]
      sum_t=sum_t+t[i,j]*l[k-1,i]
    }
  }
  th[k]<-rgamma(1,shape=a+sum_y, rate=b+sum_t)



  #update all lambdas

  for(i in 1:n){
   l[k,i]<-rgamma(1, shape=f[k-1]+sum(y[i,1:n_i[i]]), rate=f[k-1]+th[k]*sum(t[i,1:n_i[i]]))
  }



  #update phi Metropolis Algo

  ff<-rgamma(1,shape=f[k-1]*epsilon, rate=epsilon)

 product.num=dgamma(x=ff, shape=c, rate=d, log=TRUE)+dgamma(x=f[k-1], shape=ff*epsilon, rate=epsilon,
 product.denom=dgamma(x=f[k-1], shape=c, rate=d, log=TRUE)+dgamma(x=ff, shape=f[k-1]*epsilon, rate=ep
  for(i in 1:n){
    product.num=product.num+dgamma(x=l[k,i],shape=ff, rate=ff, log=TRUE)
   product.denom=product.denom+dgamma(x=l[k,i], shape=f[k-1], rate=f[k-1], log=TRUE)
  }

  alpha<-min(1, exp(product.num-product.denom))
  r<-runif(1)
```

```
  if(r<=alpha){
    f[k]=ff
  }else{
    f[k]=f[k-1]
  }


  #counter

  if(k%%2000==0){
    print(k)
  }
}


#derive posterior estimates

theta.hat=mean(th[2001:N])
phi.hat=mean(f[2001:N])
lambda.hat<-c()
for(s in 1:n){
  lambda.hat[s]=mean(l[2001:N,s])
}

plot(lambda, main="True parameter values vs. Bayes estimates", xlab=i, ylab=expression(lambda[i]))
points(lambda.hat, col="red")
legend(27, 2.5, c(expression(hat(lambda[i])), expression(lambda[i])), pt.lwd=c(2,2), pch=c(1,1), col=c
```

Finally, using post burn-in draws parameters were estimated. True parameter values and Bayes estimates for θ and ϕ are given in the table below:

θ	10
$\hat{\theta}$	9.74
ϕ	10
$\hat{\phi}$	9.53

Bayes estimation results for random effect variables $\{\lambda_i\}$ are summarized in the plot below *(true parameter values are in black, Bayes posterior estimates are in red)*:



To conclude, we can see that on average everybody gets 10 pieces of candy, but average number of candies also varies accross costume types, see λ_i's. By comparing posterior estimates and true parameter values we can see that estimates are quite precise.