Stat-Computing-2 Assignment-6

Anjan Ghosh, MS-1803

5/25/2020

Here we develop an Hierarchical model that gives the optimal number of families in finite mixture model.

Data generation

200 observations are simulated from the mixture distribution:

$$f = 0.35N(10, 5^2) + 0.65N(25, 5^2)$$

This is done in two steps.

- First, a bernoulli observation is drawn $\sim Ber(p = 0.35)$.
- If the simulated value is less than 0.35, we draw one sample from $N(10, 5^2)$ distribution, else from $N(25, 5^2)$ distribution.

Derivation of the MCMC

We use the simulated dataset and estimate the underlying model parameters using a hierarchichal Bayes algorithm. We assume that data comes from the model of the form:

$$f = \sum_{k=1}^{K} p_k N(\mu_k, \sigma^2)$$

First, fix a set of choices for K say, $2 \leq K \leq 5$.

Start with a particular K from this set and consider the hierarchical model:

$$\begin{aligned} p = (p_1, p_2, \ldots, p_K) &\sim Dirichlet(\alpha, \alpha, \ldots, \alpha) \\ y_i|k &\sim N(\mu_k, \sigma^2),\ k = 1, 2, \ldots, K \\ \mu_k &\sim N(0, 10^2),\ \sigma^2 \sim IG(0.01, 0.01) \end{aligned}$$

Latent variable z is introduced which takes value $1, 2, \ldots, K$ with probability $p_1, p_2, \ldots, p_K$ respectively and these are such that $y_i \sim N(\mu_k, \sigma^2)$ when $z_i = k$.

Denote: $y = (y_1, y_2, \ldots, y_{200}), z = (z_1, z_2, \ldots, z_{200}), \theta = (\mu_1, \mu_2, \ldots, \mu_K, \sigma)$

Given hierarchical structure along with latent z forms the base of this MCMC algorithm.

The likelihood can be written as:

$$\pi(y, z, p, \theta) \propto \pi(p) \times \pi(z|p) \times \pi(y|z, \theta) \times \pi(\theta)$$

The posteriors are as follows:

$$p = (p_1, p_2, \ldots, p_K)|z \sim Dirichlet(n_1 + \alpha, n_2 + \alpha, \ldots, n_K + \alpha)$$

$$P[z_i = k|y_i, p, \theta] \propto p_k \times exp(-\frac{1}{2\sigma^2}(y_i - \mu_k)^2), k = 1, 2, \ldots, K,\ i = 1, 2, \ldots, 200\ independently$$

$$\mu_k|y, z, \sigma^2 \sim N(\frac{\frac{n_k}{\sigma^2}\bar{y}_k}{\frac{n_k}{\sigma^2} + \frac{1}{10^2}}, \frac{1}{\frac{n_k}{\sigma^2} + \frac{1}{10^2}}),\ k = 1, 2, \ldots, K$$

$$\sigma^2|y, z, \mu_1, \mu_2, \ldots, \mu_K \sim IG(0.01 + 50, 0.01 + \frac{1}{2}\sum_{k=1}^{K}\sum i : z_i = k(y_i - \mu_k)^2)$$

Where $n_k = \sum_{i:z_i=k} 1$, $\bar{y}_k = \frac{1}{n_k}\sum_{i:z_i=k} y_i,\ k = 1, 2, \ldots, K.$

This is repeated for $K = 2, 3, 4, 5$ separately. Bayesian information criterion for each K is calculated and the model with smallest BIC is selected. Estimates of parameters of this chosen model are provided.

BIC is calculated as:

$$BIC = k\ln(n) - 2\ln(\hat{L})$$

Where

- $\hat{L}$ is the maximized likelihood value for the selected model.
- k is the number of parameters in the model.
- n is the number of observations, here 200.

R code

```
#install.packages(c("LaplacesDemon","invgamma"))
library(LaplacesDemon)
library(invgamma)
```

```
##
## Attaching package: 'invgamma'
```

```
## The following objects are masked from 'package:LaplacesDemon':
##
##     dinvchisq, dinvgamma, rinvchisq, rinvgamma
```

```
generate_data<-function(n,mu1=10,mu2=25,sig=5,p=0.35)
{
  y<-rnorm(n,ifelse(runif(n)<p,mu1,mu2),sig)
  return(y)
}
count_k<-function(k,z)
{
  count<-numeric(k)
  for(i in 1:k)
    count[i]<-sum(z==i)
  return(count)
}
get_prob<-function(y,p,sig,mu)
{
```

```
  spl<-numeric(length(y))
  for(i in 1:length(y))
  {
    prob<-p*(dnorm(y[i],mu,sig))
    prob<-prob/sum(prob)
    spl[i]<-sample(1:length(p),1,prob = prob)
  }
  return(spl)
}
simulate_mu<-function(y,z,sig,k)
{
  mu<-numeric(k)
  for(i in 1:k)
  {
    mu[i]<-rnorm(1,sum(y[z==i])*100/(sig^2+100*sum(z==i)),
                 sqrt(100*sig^2/(sig^2+100*sum(z==i))))
  }
  return(mu)
}
find_max<-function(z)
{
  return(which(table(z)==max(table(z)))[1])
}
ret_bic<-function(n,k,y)
{
  ### VECTOR/MATRIX INITIALISATIONS
  mu<-matrix(, nrow = 20000, ncol = k)
  sig<-numeric(20000)
  p<-matrix(,nrow = 20000,ncol = k)
  z<-matrix(,nrow = 20000,ncol = n)
  ### PRIORS
  mu_init<-rnorm(k,0,10)
  sig_init<-sqrt(1/rgamma(1,0.01,0.01))
  p_init<-rdirichlet(1,rep(1,k))
  z_init<-sample(1:k,n,replace = T,prob = p_init)
  ### ITERATIONS
  for(i in 1:20000)
  {
    mu[i,]<-simulate_mu(y=y,z=z_init,sig=sig_init,k=k)
    sig[i]<-sqrt(1/rgamma(1,n/2+0.01,0.01+sum((y-mu[i,z_init])^2)/2))
    p[i,]<-rdirichlet(1,count_k(k=k,z=z_init)+1)
    z[i,]<-get_prob(y=y,p=p[i,],sig=sig[i],mu=mu[i,])

    mu_init<-mu[i,]
    z_init<-z[i,]
    p_init<-p[i,]
    sig_init<-sig[i]
  }
  ### BURN-IN AND THINNING
  index<-seq(from=2001,to=20000,by=10)
  mu1<-apply(mu[index,], 2, mean)
  p1<-apply(p[index,], 2, mean)
  sig1<-mean(sig[index])
```

```
  z1<-apply(z[index,],2,find_max)
  ### LOG-LIKELIHOOD AND BIC CALCULATION
  lik<-sum(dnorm(y,mu1[z1],sig1,log = T))+sum(dnorm(mu1,0,10,log = T))+
    dinvgamma(sig1^2,0.01,0.01,log = T)+sum(count_k(k=k,z=z1)*log(p1))
  BIC<-k*log(n)-2*lik

  return(list(mu=mu[index,],p=p[index,],sigma=sig[index],z=z,BIC=BIC))
}
### DATA GENERATION
n<-200
k<-2:5
y<-generate_data(n=n)
### FINDING OPTIMUM VALUE OF K
a<-numeric(length(k))
for(i in 1:length(k))
  a[i]<-ret_bic(n=n,k=k[i],y=y)$BIC
k_opt<-k[which(a==min(a))]
### USING THE OPTIMUM VALUE OF K TO FIND ESTIMATES OF PARAMETERS OF OPTIMAL MODEL
output<-ret_bic(n=n,k=k_opt,y=y)

apply(output$p,2,mean)
apply(output$mu, 2, mean)
mean(output$sigma)
apply(output$p,2,quantile,c(0.025,0.975))
apply(output$mu,2,quantile,c(0.025,0.975))
quantile(output$sigma,c(0.025,0.975))
```

Results

Table containing BIC values

K	BIC
2	1502.8039538
3	2121.1890448
4	1825.6648347
5	2556.8414141

Optimum value of K is: 2

Table containing posterior estimates and 95% credible interval of the parameters

Parameter	Posterior Mean	Posterior Median	95% Credible Interval
p	0.2750489	0.273947	(0.2030262, 0.3490414)
μ_1	9.4865899	9.4946354	(7.7315255, 11.2400904)
μ_2	25.432117	25.4498479	(24.3899535, 26.353366)
$sigma$	5.0873425	5.0633687	(4.4686791, 5.8528933)

Conclusion

- This algorithm is able to correctly find out the actual number of distributions in the mixture.
- Posterior estimates are close to the actual values of parameters.