Mixtures

- Basic notation
- Identification
- Density estimation
- Posterior inference
- Data augmentation
- Decomposable prior
- Full conditionals
- MCMC algorithm
- Example: mixture of two normals
- Mixture with unknown number of components

Basic notation

Let the probability density function or the probability distribution of y be

$$p(y|\gamma) = \sum_{j=1}^{k} \pi_j p_j(y|\theta_j)$$

where $\gamma = (\theta_1, \ldots, \theta_k, \pi_1, \ldots, \pi_k)$, $\pi_j > 0$ for $j = 1, \ldots, k$ and $\sum_{j=1}^{k} \pi_j = 1$. The p.d.f. p_j is parameterized by θ_j.

The vectors $\theta_1, \ldots, \theta_k$ may share common component, such as in a mixture of two normal components with common variance, see below.

Identification

If the main inferential goal is identifying/interpreting the mixture components and/or clustering then *label switching* should be treated.

More precisely, it is easy to see that

$$p(y|\gamma) = p(y|\tilde{\gamma})$$

where $\tilde{\gamma} = (\theta_{j_1}, \ldots, \theta_{j_k}, \pi_{j_1}, \ldots, \pi_{j_k})$ and $j_1, \ldots, j_k$ any permutation of $1, \ldots, k$.

Density estimation

If instead the main goal is *posterior prediction* or *density estimation* then label switching is no longer a problem irrelevant.

Conditional on observations $y^n = (y_1, \ldots, y_n)$, the posterior prediction of y_{n+1} is

$$p(y_{n+1}|y^n) = \int p(y_{n+1}|\gamma, y^n)p(\gamma|y^n)d\gamma$$

where $p(\gamma|y^n)$ is the posterior distribution of γ.

The *configuration* of γ is of no interest.

Under conditionally independent observations, the above integral becomes

$$p(y_{n+1}|y^n) = \int p(y_{n+1}|\gamma)p(\gamma|y^n)d\gamma$$

Posterior inference

The posterior distribution of γ is

$$
\begin{aligned}
p(\gamma|y^n) &\propto p(\gamma)\prod_{i=1}^{n} p(y_i|\gamma) \\
&\propto p(\gamma)\prod_{i=1}^{n}\left(\sum_{i=1}^{k}\pi_j p_j(y_i|\theta_j)\right)
\end{aligned}
$$

Data augmentation

Modern Bayesian inference in mixture of distributions is mostly done via a *data augmentation* argument.

Fictitious group classifier/indicator, say z_i, is introduced per observation y_i, observation i is classified in group j when $z_i = j$.

Therefore, the above *product of sums of weighted densities* is broken down into:

$$p(\gamma|y^n, z^n) \propto p(\gamma) \prod_{j=1}^{k} \prod_{i:z_i=j} p_j(y_i|\theta_j)$$

Decomposable prior

If, additionally, $p(\gamma)$ can be decomposed in

$$p(\gamma) = p(\pi) \prod_{j=1}^{n} p(\theta_j)$$

then

$$p(\gamma | y^n, z^n) \propto p(\pi) \prod_{j=1}^{k} \prod_{i:z_i=j} p(\theta_j) p_j(y_i | \theta_j)$$

Full conditional distributions

- Components parameters ($j = 1, \ldots, k$):

$$p(\theta_j|\theta_{-j}, \pi, y^n, z^n) \equiv p(\theta_j|y^n, z^n) \propto \prod_{i:z_i=j} p(\theta_j) p_j(y_i|\theta_j)$$

and $\theta_{-1} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_k)$.

- Components weights ($j = 1, \ldots, k$):

$$p(\pi|z^n, y^n) \equiv p(\pi|z^n) \propto p(\pi) \prod_{j=1}^{n} \pi_j^{n_j}$$

where n_j is the number of observations classified in group j.

- Latent classifier ($i = 1, \ldots, n$):

$$p(z^n|y^n, \gamma) \propto \prod_{i=1}^{n} p(z_i) p(y_i|\theta_i, z_i)$$

Full conditionals (cont.)

Let the prior distribution of π be a Dirichlet(α) (generalization of the Beta). Then

$$p(\pi | z^n, y^n) \propto \prod_{j=1}^{k} \pi_j^{\alpha_j + n_j},$$

which is also a Dirichlet with parameter $\alpha + \mathbf{n}$, for $\mathbf{n} = (n_1, \ldots, n_k)$.

Also, for $i = 1, \ldots, n$,

$$(z_i | y_i, \gamma) \sim \{1, \ldots, k\}$$

with

$$Pr(z_i = j | y^n, \gamma) = \frac{\pi_j p(y_i | \theta_j)}{\sum_{l=1}^{k} \pi_l p(y_i | \theta_l)}$$

MCMC algorithm

1. Initial values z^n
2. Iterate
   - 2.1 Compute $\mathbf{n}$
   - 2.2 For $j = 1, \ldots, k$, sample θ_j from

$$p(\theta_j | y^n, z^n) \propto p(\theta_j) \prod_{i: z_i = j} p_j(y_i | \theta_j)$$

   - 2.3 Sample π from Dirichlet$(\alpha + \mathbf{n})$
   - 2.4 For $i = 1, \ldots, n$, sample z_i.

Mixture of two normals

Let $k = 2$ and $(y|\theta_j) \sim N(\mu_j, \sigma_j^2)$, $\theta_j = (\mu_j, \sigma_j^2)$,

$$\mu_j \sim N(\mu_{0j}, \tau_{0j}^2) \quad \text{and} \quad \sigma_j^2 \sim IG(\nu_{0j}/2, \nu_{0j}\sigma_{0j}^2/2)$$

for $j = 1, \ldots, k$, and initial values z^n.

Then, the previous MCMC algorithm becomes:

1. Compute $\mathbf{n}$
2. For $j = 1, \ldots, k$,
   2.1 Sample μ_j from $N(\mu_{1j}, \tau_{1j}^2)$
   2.2 Sample σ_j^2 from $IG(\nu_{1j}/2, \nu_{1j}\sigma_{1j}^2/2)$
3. Sample π from Dirichlet$(\alpha + \mathbf{n})$
4. For $i = 1, \ldots, n$, sample z_i.

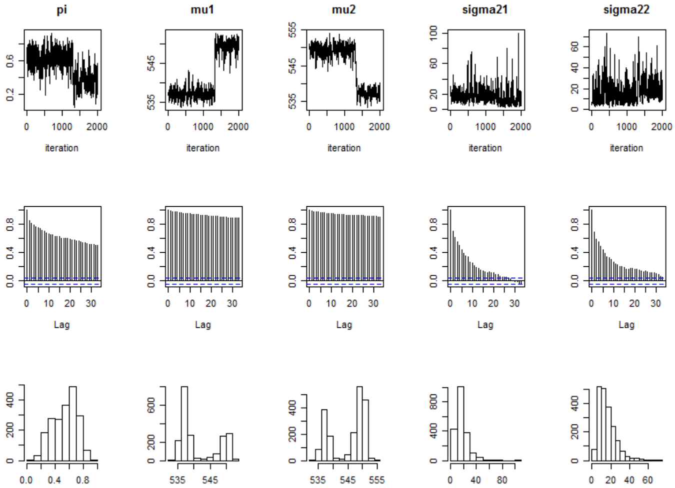
R code

```
mix2normals = function(y,a,b,mu0,tau20,nu0,nu0s02,mu,z,M){
  n=length(y);nn=rep(0,2)
  draws=matrix(0,M,5)
  for (iter in 1:M){
    nn[1]=sum(z==1);nn[2]=n-nn[1]
    # sampling sigma2's
    sy2 = c(sum((y[z==1]-mu[1])^2),sum((y[z==2]-mu[2])^2))
    sigma2 = 1/rgamma(2,(nu0+nn)/2,(nu0s02+sy2)/2)
    # sampling mu's
    var=1/(1/tau20+nn/sigma2)
    sy = c(sum(y[z==1]),sum(y[z==2]))
    mean=var*(mu0/tau20+sy/sigma2)
    mu = rnorm(2,mean,sqrt(var))
    # sampling p
    pi = rbeta(1,a+nn[1],b+nn[2])
    # sampling z's
    pz1=pi*dnorm(y,mu[1],sqrt(sigma2[1]))
    pz2=(1-pi)*dnorm(y,mu[2],sqrt(sigma2[2]))
    pz=pz1/(pz1+pz2)
    for (i in 1:n) z[i]=sample(1:2,size=1,prob=c(pz[i],1-pz[i]))
    draws[iter,]=c(pi,mu,sigma2)
  }
  return(draws)
}
```

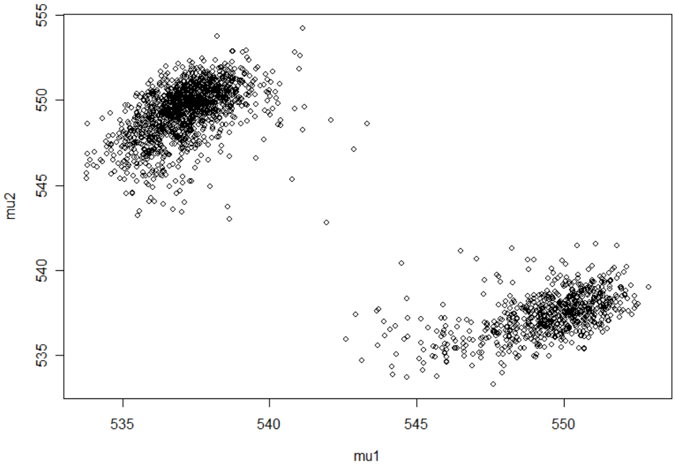
Example

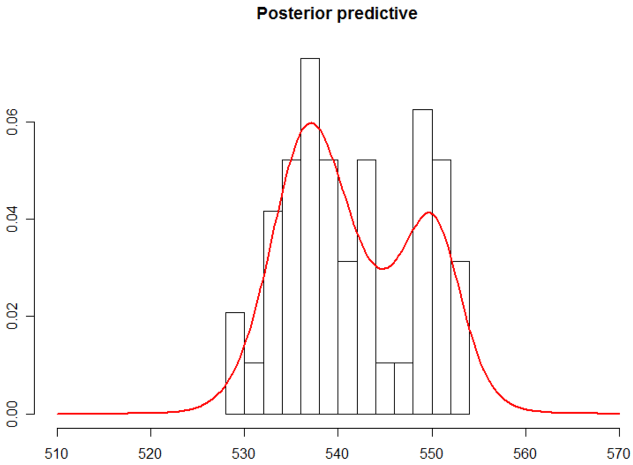
Bowmaker et al (1985) analyse data on the peak sensitivity wavelengths for individual microspectrophotometric records on a small set of monkey's eyes. Data for one monkey are given below.

```
y  = c(529.0, 530.0, 532.0, 533.1, 533.4, 533.6, 533.7, 534.1, 534.8, 535.3, 535.4, 535.9,
536.1, 536.3, 536.4, 536.6, 537.0, 537.4, 537.5, 538.3, 538.5, 538.6, 539.4, 539.6, 540.4,
540.8, 542.0, 542.8, 543.0, 543.5, 543.8, 543.9, 545.3, 546.2, 548.8, 548.7, 548.9, 549.0,
549.4, 549.9, 550.6, 551.2, 551.4, 551.5, 551.6, 552.8, 552.9,553.2)
n = length(y)
# Hyperparameters
a=1;b=1;nu0=rep(3,2);nu0s02=rep(3*20,2);mu0=rep(535,550);tau20=rep(1000,2)
# Initial values
z=rep(1,n);z[y>545]=2;mu=c(mean(y[z==1]),mean(y[z==2]))
# MCMC
set.seed(13579)
run = mix2normals(y,a,b,mu0,tau20,nu0,nu0s02,mu,z,2000)
# Posterior parameter summary
names = c("pi","mu1","mu2","sigma21","sigma22")
png(file="posterior.png",height=600,width=800)
par(mfrow=c(3,5))
for (i in 1:5) ts.plot(run[,i],xlab="iteration",ylab="",main=names[i])
for (i in 1:5) acf(run[,i],ylab="",main="")
for (i in 1:5) hist(run[,i],xlab="",ylab="",main="")
dev.off()
png(file="scatterplot.png",height=600,width=800)
par(mfrow=c(1,1))
plot(run[,2:3],xlab="mu1",ylab="mu2",main="")
dev.off()
# Posterior predictive
N=200;yy=seq(510,570,length=N)
pred = rep(0,N);probs=cbind(run[,1],1-run[,1])
for (i in 1:N) pred[i]=mean(apply(probs*dnorm(yy[i],run[,2:3],sqrt(run[,4:5])),1,sum))
png(file="postpred.png",height=600,width=800)
hist(y,nclass=12,xlab="",ylab="",main="Posterior predictive",prob=T,xlim=range(yy))
lines(yy,pred,col=2,lwd=2)
dev.off()
```



pi
mu1
mu2
sigma21
sigma22
iteration
Lag


555
550
545
540
535
mu2
535
540
545
550
mu1



Posterior predictive
0.00
0.02
0.04
0.06
510
520
530
540
550
560
570