Finite mixture models

Dr. Jarad Niemi

STAT 615 - Iowa State University

November 9, 2021

Categorical distribution

Let $Z \sim Cat(H, p)$ represent a categorical distribution with

- $P(Z = h) = p_h$ for $h = 1, \ldots, H$ and
- $\sum_{h=1}^{H} p_h = 1$.

Example: discrete choice model

Suppose we have a set of H categories and we label these $1, \ldots, H$. Independently, consumers choose one of the H categories with the same probability. Then a reasonable model is $Z_i \overset{ind}{\sim} Cat(H, p)$.

Multinomial distribution

If we count the number of times the consumer chose each category, i.e.

$$Y_h = \sum_{i=1}^{n} \mathrm{I}(Z_i = h),$$

then the result is the multinomial distribution, i.e. $Y \sim Mult(n, p)$. The multinomial distribution has probability mass function

$$p(y; n, p) = \frac{n!}{y_1! \cdots y_H!} \prod_{h=1}^{H} p_h^{y_h}$$

which has

- $E[Y_i] = np_i$,
- $V[Y_i] = np_i(1 - p_i)$, and
- $Cov[Y_i, Y_j] = -np_ip_j$ for $(i \neq j)$.

A special case is $H = 2$ which is the binomial distribution.

Dirichlet distribution

The Dirichlet distribution (named after Peter Gustav Lejeune Dirichlet), i.e. $P \sim Dir(a)$, is a probability distribution for a probability vector of length H. The probability density function for the Dirichlet distribution is

$$p(P;a) = \frac{\Gamma(a_1 + \cdots + a_H)}{\Gamma(a_1) \cdots \Gamma(a_H)} \prod_{h=1}^{H} p_h^{a_h - 1}$$

where $p_h \geq 0$, $\sum_{h=1}^{H} p_h = 1$, and concentration parameters $a_h > 0$.

Letting $a_0 = \sum_{h=1}^{H} a_h$, then some moments are

- $E[p_h] = \frac{a_h}{a_0}$,
- $V[p_h] = \frac{a_h(a_0 - a_h)}{a_0^2(a_0+1)}$,
- $Cov(p_h, p_k) = -\frac{a_h a_k}{a_0^2(a_0+1)}$, and
- mode$(p_h) = \frac{a_h - 1}{a_0 - H}$ for $a_h > 1$.

A special case is $H = 2$ which is the beta distribution.

Conjugate prior for multinomial distribution

The Dirichlet distribution is the natural conjugate prior for the multinomial distribution. If

$$Y \sim Mult(n, \pi) \qquad \text{and} \qquad \pi \sim Dir(a)$$

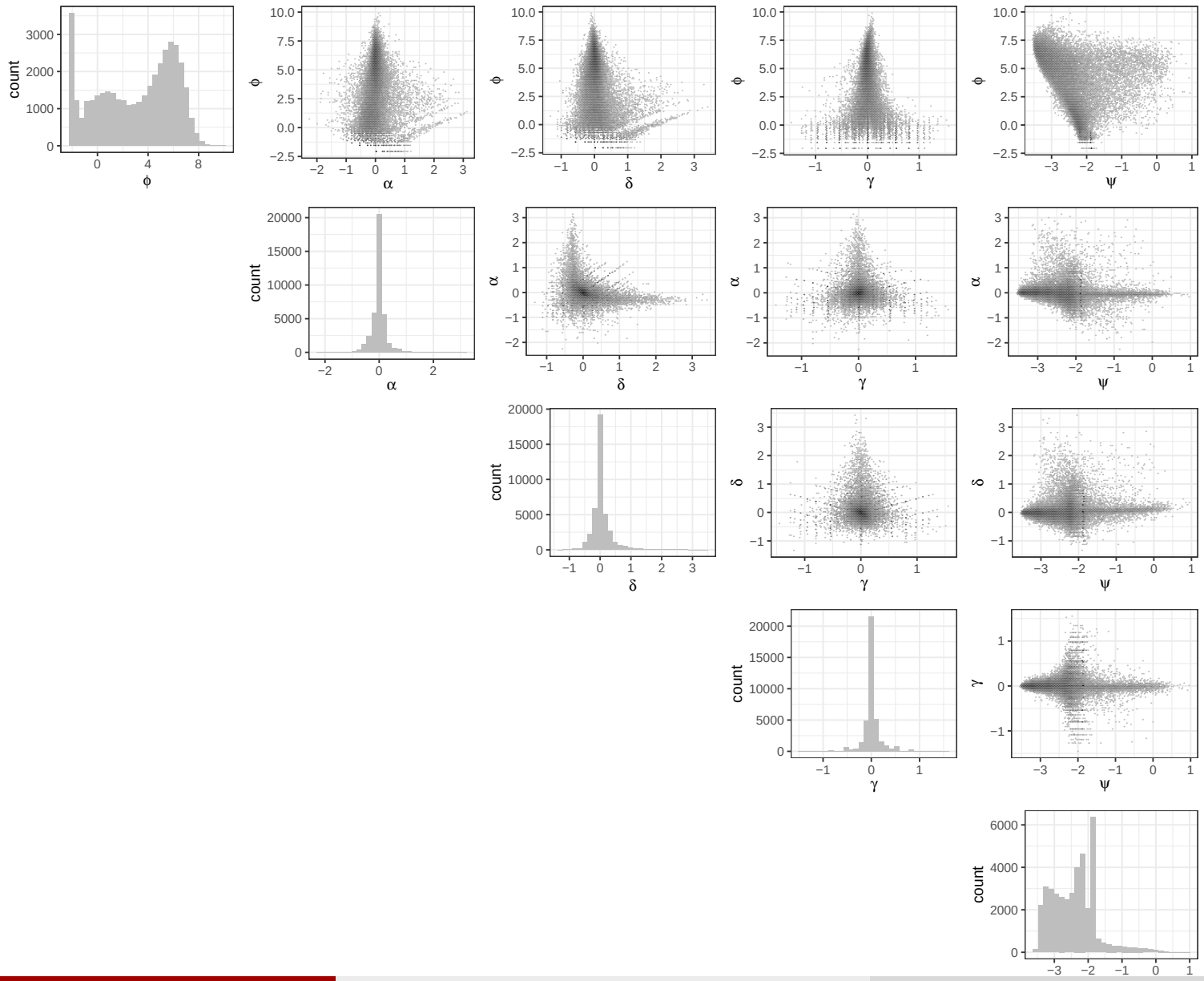
then

$$\pi|y \sim Dir(a + y).$$

Some possible default priors are

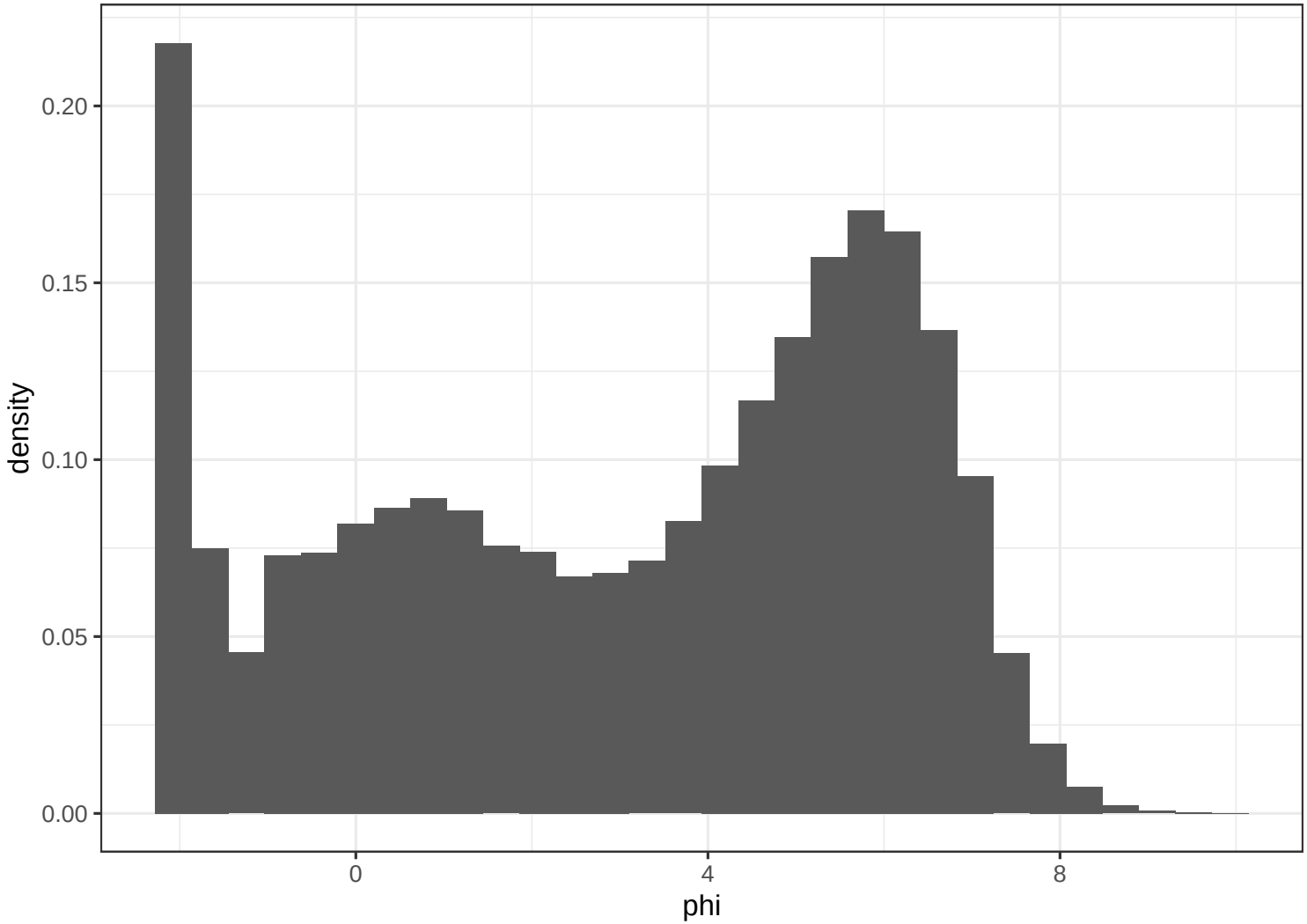
- $a = 1$ which is the uniform density over π,
- $a = 1/2$ is Jeffreys prior for the multinomial,
- $a = 1/H$, and
- $a = 0$, an improper prior that is uniform on $\log(\pi_h)$. The resulting posterior is proper if $y_h > 0$ for all h.

Complicated distributions



Finite mixtures

Let's focus on modeling the univariate distribution for $\hat{\phi}$



Finite mixture

A model for the marginal distribution for $Y_i = \hat{\phi}_i$ is

$$Y_i \overset{ind}{\sim} \sum_{h=1}^{H} \pi_h N(\mu_h, \sigma_h^2)$$

where $\sum_{h=1}^{H} \pi_h = 1$.

Alternatively, we can introduce a latent variable $\zeta_i = h$ if observation i came from group h. Then

$$\begin{aligned} Y_i|\zeta_i = z &\overset{ind}{\sim} N(\mu_z, \sigma_z^2) \\ \zeta_i &\overset{ind}{\sim} Cat(H, \pi) \end{aligned}$$

where $\zeta \sim Cat(H, \pi)$ is a categorical random variable with $P(\zeta = h) = \pi_h$ for $h = 1, \ldots, H$ and $\pi = (\pi_1, \ldots, \pi_H)$.

A possible prior

Let's assume

$$\begin{aligned} \pi &\sim Dir(a) \\ \mu_h|\sigma_h^2 &\overset{ind}{\sim} N(m_h, v_h^2\sigma_h^2) \\ \sigma_h^2 &\overset{ind}{\sim} IG(c_h, d_h) \end{aligned}$$

and π is independent of $\mu = (\mu_1, \ldots, \mu_H)$ and $\sigma^2 = (\sigma_1^2, \ldots, \sigma_H^2)$.

Commonly, we have $m_h = m$, $v_h = v$, $c_h = c$, and $d_h = d$. If the data have been standardized (scaled and centered), a reasonable default prior is

- $m = 0$,
- $v = 1$,
- $c = 2$,
- $d = 4$, (BDA3 pg 535) and
- a is $1/H$ (BDA3 pg 536).

MCMC

The steps of a Gibbs sampler with stationary distribution

$$p(\pi, \mu, \sigma^2, \zeta|y) \propto p(y|\zeta, \mu, \sigma^2)p(\zeta|\pi)p(\mu|\sigma^2)p(\sigma^2)p(\pi)$$

has steps

1. For $i = 1, \ldots, n$, sample ζ_i from its full conditional (as the ζ are conditionally independent across i):

$$P(\zeta_i = h| \ldots) \propto \pi_h N(y_i; \mu_h, \sigma_h^2)$$

2. Jointly sample π and μ, σ^2 as they are conditionally independent.
   a. Sample $\pi \sim Dir(a + Z)$ where $n = (Z_1, \ldots, Z_H)$ and $Z_h = \sum_{i=1}^n \mathrm{I}(\zeta_i = h)$.
   b. For $h = 1, \ldots, H$, sample μ_h, σ_h^2 from their full conditional (as these are conditionally independent across h):

$$\sigma_h^2 \overset{ind}{\sim} IG(c'_h, d'_h) \quad \mu_h|\sigma_h^2 \overset{ind}{\sim} N(m'_h, v'^2_h \sigma_h^2)$$

where

$$\begin{aligned}
v'^2_h &= (1/v_h^2 + Z_h)^{-1} \\
m'_h &= v'^2_h(m_h/v_h^2 + Z_h\overline{y}_h) \\
c'_h &= c_d + Z_h/2 \\
d'_h &= d_h + \tfrac{1}{2}\left(\textstyle\sum_{i:\zeta_i=h}(y_i - \overline{y}_h)^2 + \left(\frac{Z_h}{1+Z_h/v_h^2}\right)(\overline{y}_h - m_h)^2\right) \\
\overline{y}_h &= \tfrac{1}{Z_h}\textstyle\sum_{i:\zeta_i=h} y_i
\end{aligned}$$

```
library("rjags")
jags_model = "
model {
  for (i in 1:n) {
    y[i] ~ dnorm(mu[zeta[i]], tau[zeta[i]])
    zeta[i] ~ dcat(pi[])
  }

  for (i in 1:H) {
    mu[i] ~ dnorm(0,1e-5)
    tau[i] ~ dgamma(1,1)
    sigma[i] <- 1/sqrt(tau[i])
  }

  pi ~ ddirich(a)
}"
```

```
tmp = hat[sample(nrow(hat), 1000),]
dat = list(n=nrow(tmp), H=3, y=tmp$phi, a=rep(1,3))
```

```
jm = jags.model(textConnection(jags_model), data = dat, n.chains = 3)
r = coda.samples(jm, c('mu','sigma','pi'), 1e3)
```

Convergence diagnostics

```
gelman.diag(r)
```

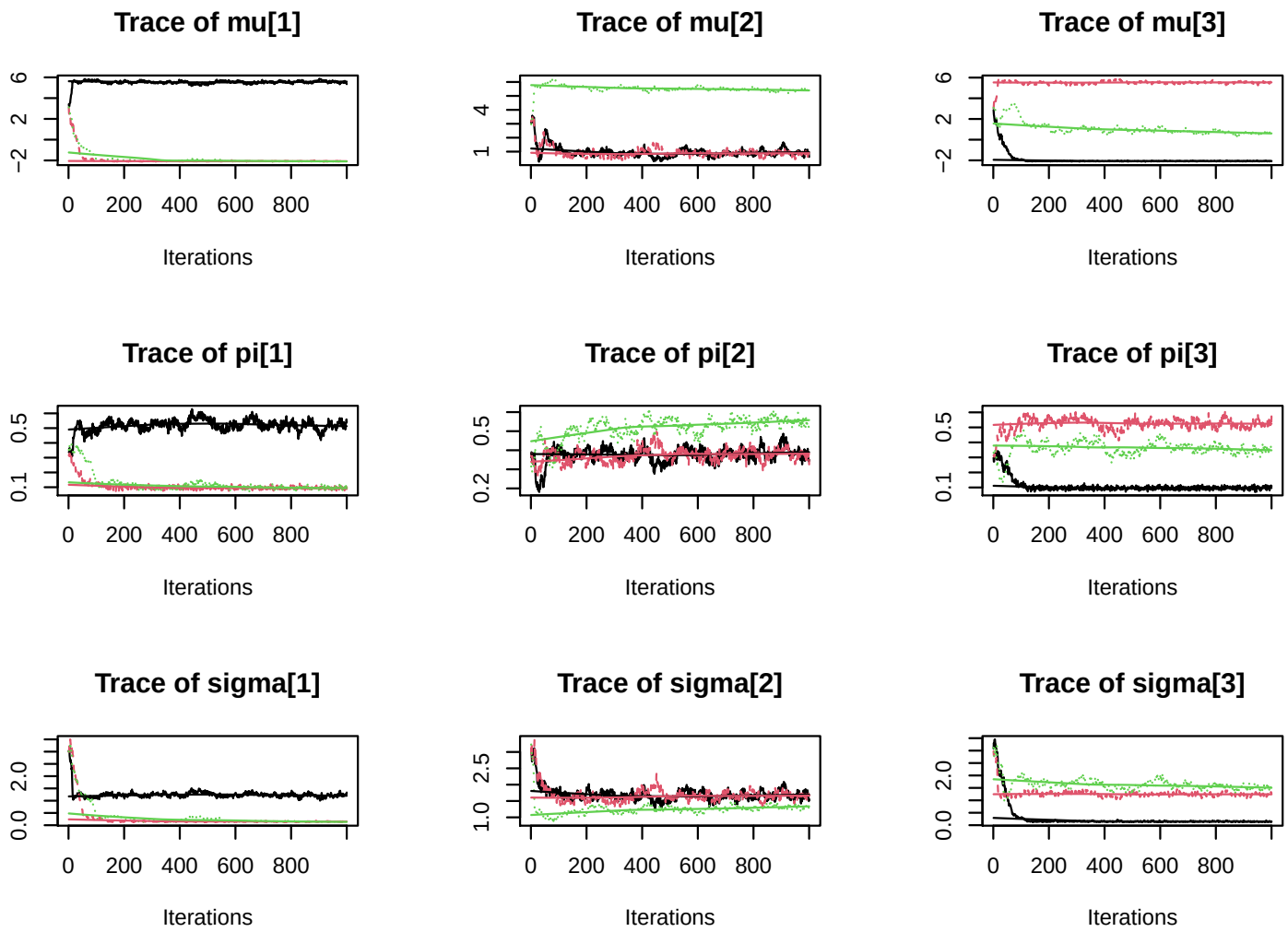
```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## mu[1]         99.02     358.12
## mu[2]         23.57      49.18
## mu[3]         42.49     150.84
## pi[1]         18.64      58.20
## pi[2]          4.70       8.83
## pi[3]         13.38      31.09
## sigma[1]      18.46      50.98
## sigma[2]       3.08       5.96
## sigma[3]      12.75      41.99
##
## Multivariate psrf
##
## 102
```

```
gelman.diag(r, multivariate=FALSE)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## mu[1]         99.02     358.12
## mu[2]         23.57      49.18
## mu[3]         42.49     150.84
## pi[1]         18.64      58.20
## pi[2]          4.70       8.83
```

Convergence diagnostics (2)

```
plot(r, density=FALSE)
```



Prior distributions

The parameters of the model are unidentified due to label-switching, i.e.

$$Y_i \overset{ind}{\sim} \sum_{h=1}^{H} \pi_h N(\mu_h, \sigma_h^2) \overset{d}{=} \sum_{h'=1}^{H} \pi_{h'} N(\mu_{h'}, \sigma_{h'}^2)$$

for some permutation h'.

One way to resolve this issue is to enforce identifiability in the prior. For example, in one-dimension, we can order the component means: $\mu_1 < \mu_2 < \cdots < \mu_H$.

To ensure the posterior is proper

- Maintain proper prior for π
- Ensure proper prior for ratios of variances
  (perhaps by ensuring prior is proper for variances themselves)

Two conditionally conjugate prior options

Option 1:

$$Dir(\pi; a)\mathrm{I}(\mu_1 < \cdots < \mu_H) \prod_{h=1}^{H} N(\mu_h; m_h, v_h^2 \quad) IG(\sigma_h^2; c_h, d_h)$$

Option 2:

$$Dir(\pi; a)\mathrm{I}(\mu_1 < \cdots < \mu_H) \prod_{h=1}^{H} N(\mu_h; m_h, v_h^2\sigma_h^2) IG(\sigma_h^2; c_h, d_h)$$

```
library("rjags")
jags_model = "
model {
  for (i in 1:n) {
    y[i] ~ dnorm(mu[zeta[i]], tau[zeta[i]])
    zeta[i] ~ dcat(pi[])
  }

  for (i in 1:H) {
    mu0[i] ~ dnorm(0,1e-5)
    tau[i] ~ dgamma(1,1)
    sigma[i] <- 1/sqrt(tau[i])
  }

  mu[1:H] <- sort(mu0)
  pi ~ ddirich(a)
}"
```

```
jm = jags.model(textConnection(jags_model), data = dat, n.chains = 3)
r = coda.samples(jm, c('mu','sigma','pi'), 1e3)
```

Convergence diagnostics

```
gelman.diag(r)
```

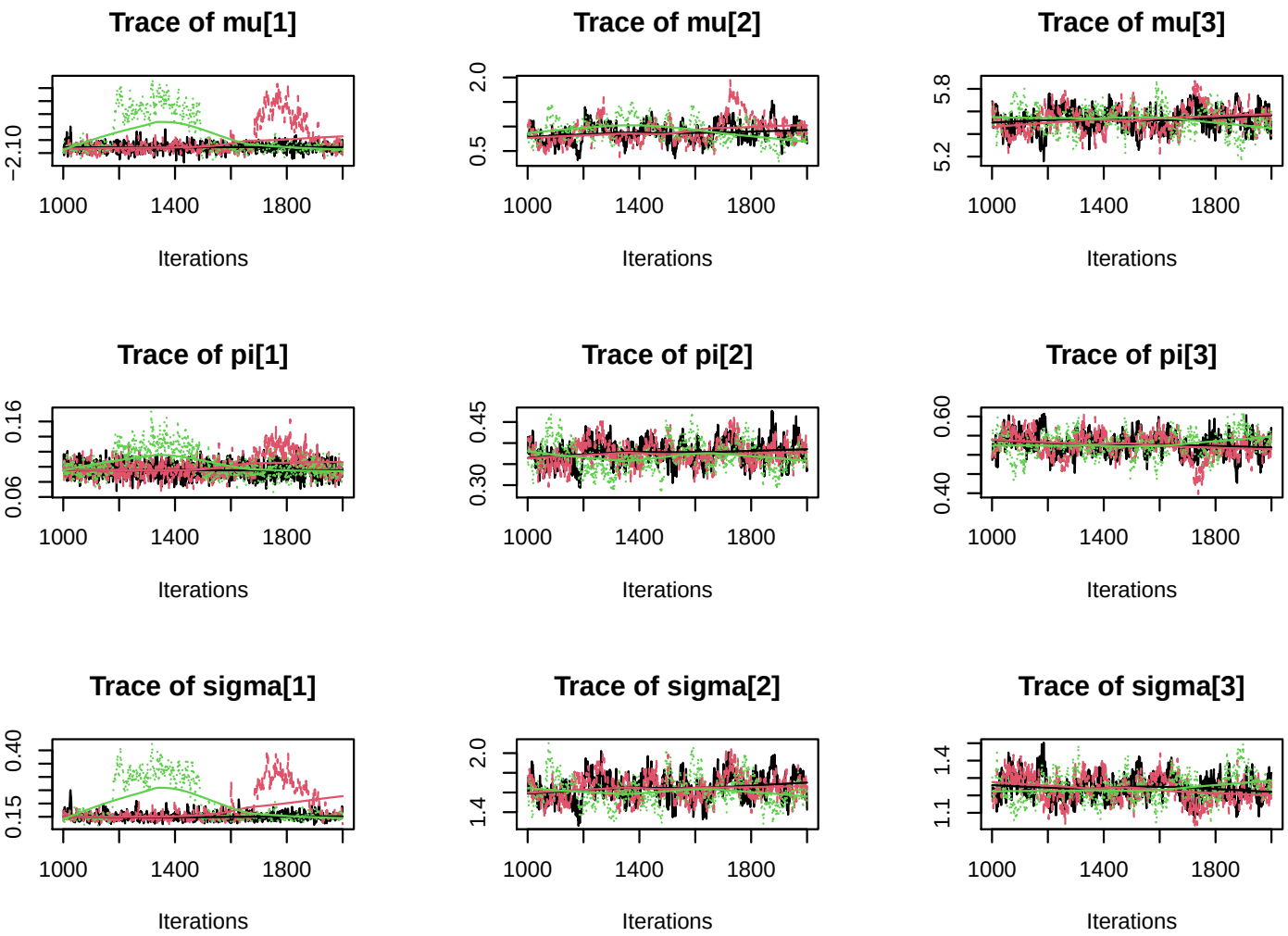
```
## Error in chol.default(W): the leading minor of order 6 is not positive definite
```

```
gelman.diag(r, multivariate=FALSE)
```

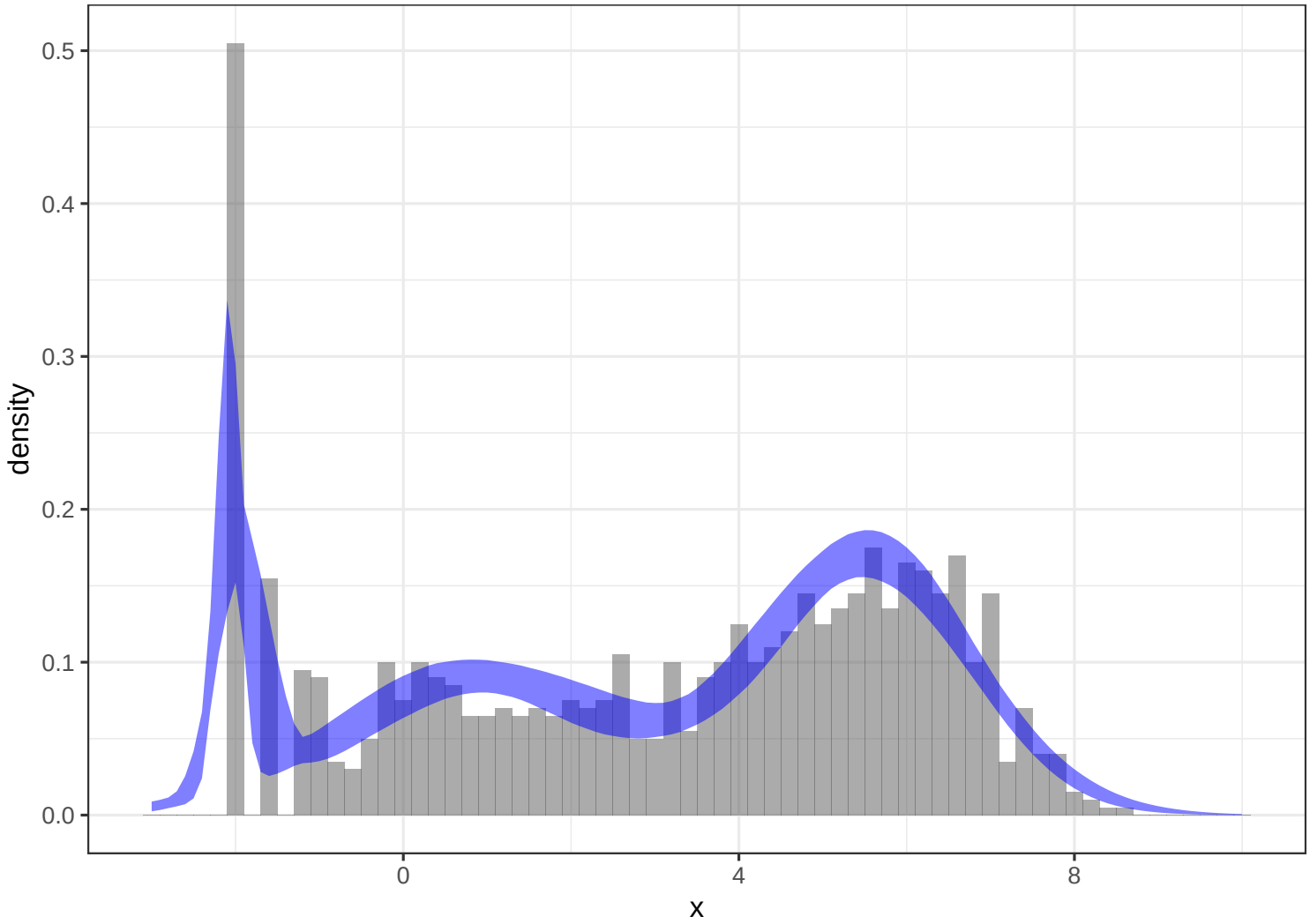
```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## mu[1]          1.20       1.68
## mu[2]          1.02       1.05
## mu[3]          1.00       1.01
## pi[1]          1.10       1.30
## pi[2]          1.03       1.08
## pi[3]          1.01       1.01
## sigma[1]       1.22       1.75
## sigma[2]       1.02       1.06
## sigma[3]       1.00       1.01
```

Convergence diagnostics (2)

```
plot(r, density=FALSE)
```



Posterior on data density



Group membership

Group membership can be obtained using the ζ_i, e.g.

$$P(\text{gene } i \text{ in cluster } h) = P(\zeta_i = h|y) \approx \sum_{m=1}^{M} \mathrm{I}\left(\zeta_i^{(m)} = h\right).$$

```
##    parameter p1          p2        p3
## 1    zeta[1]  0 0.001000000 0.9990000
## 2   zeta[10]  0 0.001333333 0.9986667
## 3  zeta[100]  0 0.003666667 0.9963333
## 4 zeta[1000]  0 0.007333333 0.9926667
## 5  zeta[101]  0 0.030333333 0.9696667
## 6  zeta[102]  0 0.785000000 0.2150000
```

Clustering

Genes can then be clustered by assigning them to a group based on their posterior probabilities of group membership, i.e. for gene i, we assign the group according to

$$\text{argmax}_h P(\zeta_i = h|y).$$

Unfortunately clustering is extremely sensitive to the parametric model chosen, e.g. normal in this example, and the cluster could change dramatically with a different choice, e.g. t.

Choosing H

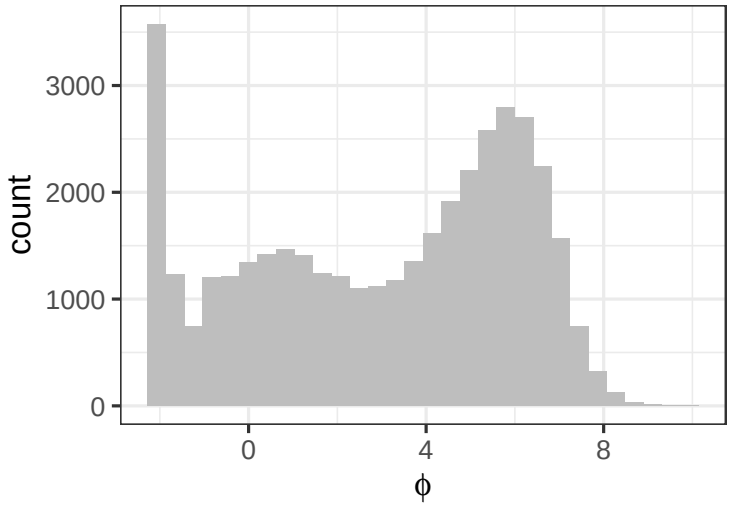
When using finite mixture models one of the key choices is to choose H, the number of clusters.

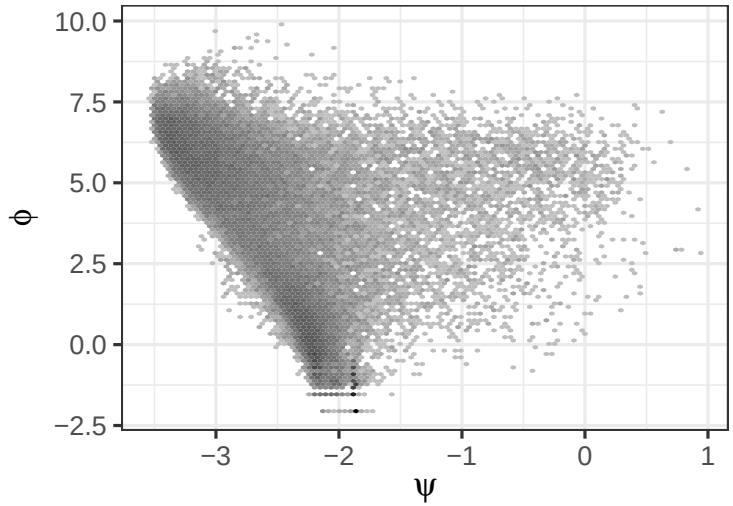
- A Bayesian approach would place a prior on H, e.g. a Poisson or truncated Poisson, and then use reversible jump MCMC to estimate it.
- A more pragmatic approach is to start with a small H and then determine whether there is some feature of the data that is not being adequately addressed, e.g. via posterior predictive pvalues.
- An empirical Bayes finds an MLE (or MAP) via

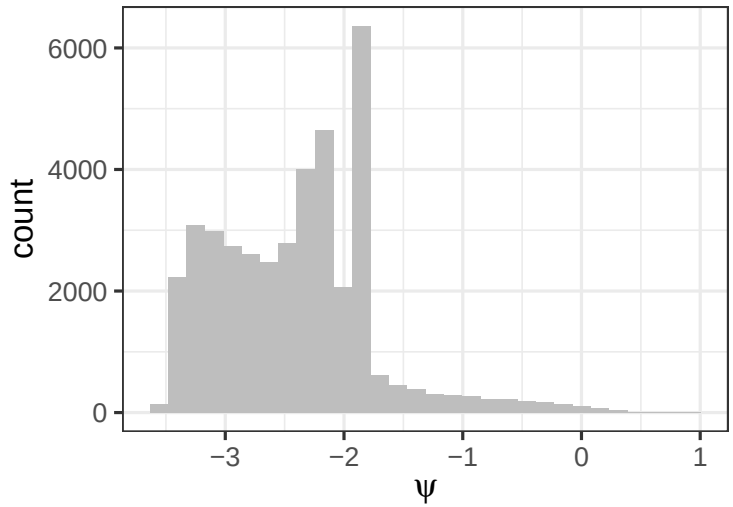
$$\hat{H} = \text{argmax}_H p(y|H) = \int p(y|\pi, \mu, \sigma^2, H) p(\pi, \mu, \sigma^2|H) d\pi d\mu d\sigma^2$$

and then condition on $\hat{H}$ in the analysis. Typically this MLE (or MAP) is found via the EM algorithm.

Multivariate density estimation







Finite mixture

A model for the joint distribution for $Y_i = (\phi_i, \psi)^\top$ is

$$Y_i \overset{ind}{\sim} \sum_{h=1}^{H} \pi_h N(\mu_h, \Sigma_h)$$

where $\sum_{h=1}^{H} \pi_h = 1$.

Alternatively, we can introduce a latent variable $\zeta_i = h$ if observation i came from group h. Then

$$\begin{aligned} Y_i|\zeta_i = z &\overset{ind}{\sim} N(\mu_z, \Sigma_z) \\ \zeta_i &\overset{ind}{\sim} Cat(H, \pi) \end{aligned}$$

where $\zeta \sim Cat(H, \pi)$ is a categorical random variable with $P(\zeta = h) = \pi_h$ for $h = 1, \ldots, H$ and $\pi = (\pi_1, \ldots, \pi_H)$.

A possible prior

Let's assume

$$\begin{aligned} \pi &\sim Dir(a) \\ \mu_h|\Sigma_h &\overset{ind}{\sim} N_p(m_h, v_h^2\Sigma_h) \\ \Sigma_h &\overset{ind}{\sim} IW(D_h, c_h) \end{aligned}$$

where $c_h > p-1$ is the degrees of freedom and D is the scale matrix. The mean of this distribution is $D_h/(c_h - p - 1)$ for $c_h > p+1$.

MCMC

The steps of a Gibbs sampler with stationary distribution

$$p(\pi, \mu, \Sigma, \zeta|y) \propto p(y|\zeta, \mu, \Sigma)p(\zeta|\pi)p(\mu|\Sigma)p(\Sigma)p(\pi)$$

has steps

1. For $i = 1, \ldots, n$, sample ζ_i from its full conditional

$$P(\zeta_i = h| \ldots) \propto \pi_h N(y_i; \mu_h, \Sigma_h)$$

2. Jointly sample π and μ, σ^2 as they are conditionally independent.
   a. Sample $\pi \sim Dir(a + Z)$ where $Z = (Z_1, \ldots, Z_H)$ and $Z_h = \sum_{i=1}^n \mathrm{I}(\zeta_i = h)$.
   b. For $h = 1, \ldots, H$, sample μ_h, Σ_h from their full conditional

$$\Sigma_h \overset{ind}{\sim} IW(D_h', c_h') \quad \mu_h|\Sigma_h \overset{ind}{\sim} N(m_h', v_h'^2\Sigma_h)$$

where

$$\begin{aligned}
v_h'^2 &= (1/v_h^2 + Z_h)^{-1} \\
m_h' &= v_h'^2(m_h/v_h^2 + Z_h\overline{y}_h) \\
c_h' &= c_d + Z_h \\
D_h' &= D_h + \textstyle\sum_{i:\zeta_i=h}(y_i - \overline{y}_h)(y_i - \overline{y}_h)^\top + \left(\frac{Z_h}{1+Z_h/v_h^2}\right)(\overline{y}_h - \mu_h)(\overline{y}_h - \mu_h)^\top \\
\overline{y}_h &= \tfrac{1}{Z_h}\textstyle\sum_{i:\zeta_i=h} y_i
\end{aligned}$$

```
library("rjags")
joint_mixture_model = "
model {
  for (i in 1:n) {
    y[i,1:p] ~ dmnorm(mu[,zeta[i]], Tau[,,zeta[i]])
    zeta[i] ~ dcat(pi[])
  }

  for (h in 1:H) {
    mu[1:p,h] ~ dmnorm(mu0,Tau[,,h])
    Tau[1:p,1:p,h] ~ dwish(D[,],c)
    Sigma[1:p,1:p,h] <- inverse(Tau[,,h])
  }

  pi ~ ddirich(a)
}"
```

```
tmp = hat[sample(nrow(hat), 1000),]
dat = list(n=nrow(tmp), y = tmp[,c('phi','psi')], p=2, H=10)
dat$a = rep(1/dat$H, dat$H)
dat$D = diag(1, dat$p)
dat$c = dat$p+1
dat$mu0 = c(3,0)
```

```
jm = jags.model(textConnection(joint_mixture_model),
                data = dat,
                n.chains = 3)
r = coda.samples(jm, c('pi','mu','Sigma'), 1e3)
```