Two-Component Mixture of Normals

MC SIR, Random-Walk Metropolis, and Gibbs Sampling

Hedibert F. Lopes

May 11, 2026

My commands to Claude

Let $y_1, \ldots, y_n$ iid from a two-component mixture of normals

$$\pi N(y_i|\mu_0, \sigma_0^2) + (1 - \pi)N(y_i|\mu_1, \sigma_1^2),$$

for $\mu_0 < \mu_1$. For now, let us keep $(\pi, \sigma_0, \sigma_1)$ fixed at $(0.75, 2, 1)$.

Assume $p(\mu_0, \mu_1) \propto 1$ (noninformative prior) constrained by $\mu_0 < \mu_1$. This constraint is useful for identification of the mixture components.

Obtain posterior draws for (μ_1, μ_2) by 4 MC schemes:

1. Sampling-importance resampling (SIR);
2. Random-walk Metropolis: sample μ_0 and μ_1 separately;
3. Random-walk Metropolis: sample (μ_0, μ_1) jointly;
4. Gibbs sampler by introducing latent indicators $z_i \in \{0, 1\}$ that label the components at which observation y_i belongs.

Output

Produce posterior kernel density approximations for μ_0 and μ_1 based on the 4 MC schemes;

Compare the ACF of MC schemes 2), 3) and 4).

Finally, organize all the derivations in a Rmarkdown that interacts with LaTeX and produce slides for presentation in my tonight's class.

The Model

Let $y_1, \ldots, y_n$ be a random sample from the two-component mixture

$$y_i \;\sim\; \pi\,\mathcal{N}(\mu_0, \sigma_0^2) \;+\; (1-\pi)\,\mathcal{N}(\mu_1, \sigma_1^2), \qquad \mu_0 < \mu_1.$$

For now, fix $(\pi, \sigma_0, \sigma_1) = (0.75,\ 2,\ 1)$.

Noninformative prior with the identifying order constraint

$$p(\mu_0, \mu_1) \;\propto\; \mathbf{1}\{\mu_0 < \mu_1\}.$$

The constraint $\mu_0 < \mu_1$ removes the label-switching non-identifiability of mixtures.

Likelihood and Posterior

$$L(\mu_0, \mu_1) = \prod_{i=1}^{n} \Big[\pi \, \phi(y_i \,|\, \mu_0, \sigma_0^2) + (1-\pi) \, \phi(y_i \,|\, \mu_1, \sigma_1^2)\Big]$$

$$p(\mu_0, \mu_1 \mid y) \;\propto\; L(\mu_0, \mu_1) \, \mathbf{1}\{\mu_0 < \mu_1\}.$$

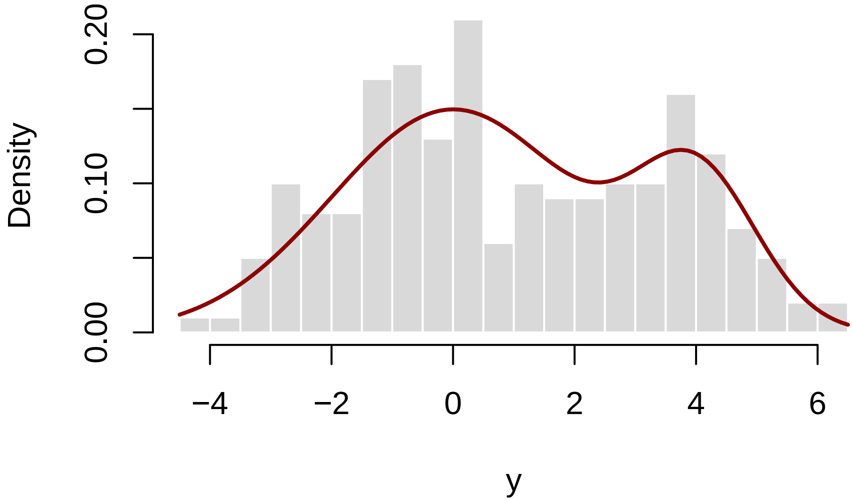
Log-posterior (up to a constant):

$$\ell(\mu_0, \mu_1) = \sum_{i=1}^{n} \log\Big[\pi \, \phi(y_i|\mu_0, \sigma_0^2) + (1-\pi) \, \phi(y_i|\mu_1, \sigma_1^2)\Big] + \log \mathbf{1}\{\mu_0 < \mu_1\}.$$

Simulated Data

```r
set.seed(123)
n       <- 200
pi_w    <- 0.75; s0 <- 2; s1 <- 1
mu0_T   <- 0;    mu1_T <- 4
z_T     <- rbinom(n, 1, 1 - pi_w)              # 0 with prob pi_w
y       <- ifelse(z_T == 0,
                  rnorm(n, mu0_T, s0),
                  rnorm(n, mu1_T, s1))

logpost <- function(mu0, mu1, y, pi_w, s0, s1) {
  if (mu0 >= mu1) return(-Inf)
  sum(log(pi_w * dnorm(y, mu0, s0) +
          (1 - pi_w) * dnorm(y, mu1, s1)))
}
```

Simulated data



Method 1: Sampling-Importance Resampling (SIR)

Idea. Draw $(\mu_0^{(j)}, \mu_1^{(j)}) \sim g$ for $j = 1, \ldots, M$, compute weights

$$w_j \;\propto\; \frac{p(\mu_0^{(j)}, \mu_1^{(j)} \mid y)}{g(\mu_0^{(j)}, \mu_1^{(j)})}, \qquad \widetilde{w}_j = \frac{w_j}{\sum_k w_k},$$

then **resample** N values (μ_0, μ_1) with probabilities $\widetilde{w}_j$.

Proposal: $\mu_0, \mu_1 \overset{\text{iid}}{\sim} \mathcal{N}(\bar{y}, 4^2)$.

The indicator $\mathbf{1}\{\mu_0 < \mu_1\}$ in the posterior sets $w_j = 0$ whenever $\mu_0^{(j)} \geq \mu_1^{(j)}$.

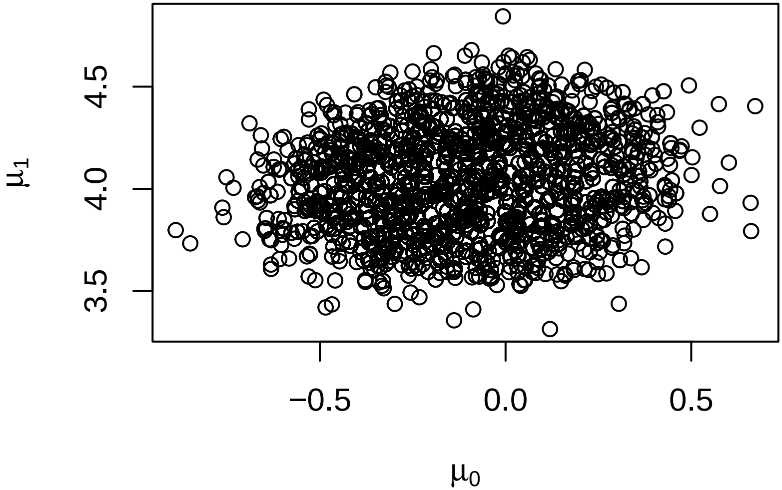
Method 1: SIR Code

```
M <- 200000; N <- 20000
ybar <- mean(y); sdp <- 4
mu0p <- rnorm(M, ybar, sdp)
mu1p <- rnorm(M, ybar, sdp)

logw <- mapply(function(a,b) logpost(a,b,y,pi_w,s0,s1),
               mu0p, mu1p) -
        dnorm(mu0p, ybar, sdp, log = TRUE) -
        dnorm(mu1p, ybar, sdp, log = TRUE)
logw <- logw - max(logw)
w    <- exp(logw); w <- w / sum(w)
idx  <- sample.int(M, N, replace = TRUE, prob = w)
sir  <- cbind(mu0 = mu0p[idx], mu1 = mu1p[idx])

ess <- 1 / sum(w^2)                # effective sample size
```

Effective sample size for SIR: 589 out of 2×10^5.

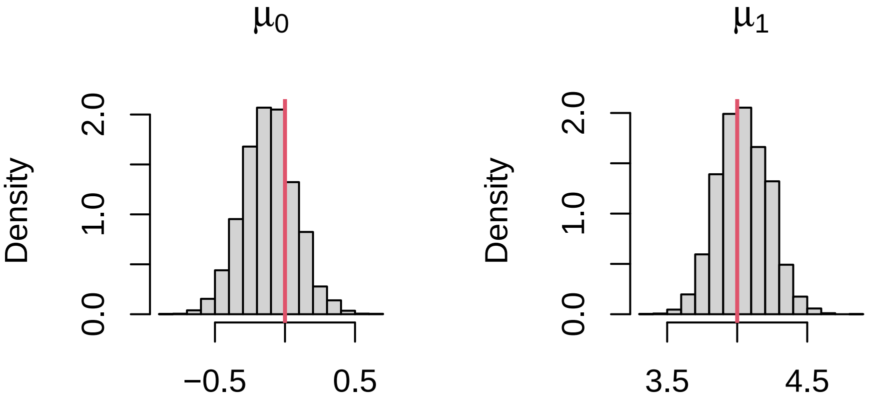
Posterior draws

```
plot(sir[,1],sir[,2],xlab=expression(mu[0]),
     ylab=expression(mu[1]))
```



Marginal densities

```
par(mfrow=c(1,2))
hist(sir[,1],main=expression(mu[0]),xlab="",prob=TRUE)
abline(v=mu0_T,col=2,lwd=2)
hist(sir[,2],main=expression(mu[1]),xlab="",prob=TRUE)
abline(v=mu1_T,col=2,lwd=2)
```



Method 2: Random-Walk Metropolis (single move)

Update μ_0 and μ_1 from their **full conditionals**:

$$p(\mu_0 \mid \mu_1, y) \propto L(\mu_0, \mu_1)\, \mathbf{1}\{\mu_0 < \mu_1\},$$

$$p(\mu_1 \mid \mu_0, y) \propto L(\mu_0, \mu_1)\, \mathbf{1}\{\mu_0 < \mu_1\}.$$

RW step for μ_0: propose $\mu_0^\star = \mu_0^{(t)} + \epsilon,\ \epsilon \sim \mathcal{N}(0, \tau^2)$, accept with

$$\alpha = \min\left\{1,\ \frac{p(\mu_0^\star \mid \mu_1^{(t)}, y)}{p(\mu_0^{(t)} \mid \mu_1^{(t)}, y)}\right\}.$$

The constraint is enforced because $\log p = -\infty$ when $\mu_0^\star \geq \mu_1^{(t)}$.

Method 2 Code

```
N <- 20000; burn <- 2000
rw1 <- matrix(NA, N + burn, 2)
mu0 <- ybar - 1; mu1 <- ybar + 1; tau <- 0.30
acc <- c(0,0)
for (t in 1:(N + burn)) {
  p0 <- mu0 + rnorm(1, 0, tau)
  if (log(runif(1)) < logpost(p0, mu1, y, pi_w, s0, s1) -
                      logpost(mu0, mu1, y, pi_w, s0, s1)) {
    mu0 <- p0; acc[1] <- acc[1] + 1
  }
  p1 <- mu1 + rnorm(1, 0, tau)
  if (log(runif(1)) < logpost(mu0, p1, y, pi_w, s0, s1) -
                      logpost(mu0, mu1, y, pi_w, s0, s1)) {
    mu1 <- p1; acc[2] <- acc[2] + 1
  }
  rw1[t, ] <- c(mu0, mu1)
}
rw1     <- rw1[-(1:burn), ]
acc_rw1 <- acc / (N + burn)
```

Method 3: Random-Walk Metropolis (joint move)

Update (μ_0, μ_1) **jointly**: propose

$$\begin{pmatrix} \mu_0^\star \\ \mu_1^\star \end{pmatrix} = \begin{pmatrix} \mu_0^{(t)} \\ \mu_1^{(t)} \end{pmatrix} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}_2(\mathbf{0},\, \tau^2 I_2),$$

accept with

$$\alpha = \min\left\{1,\ \frac{p(\mu_0^\star, \mu_1^\star \mid y)}{p(\mu_0^{(t)}, \mu_1^{(t)} \mid y)}\right\}.$$

A single accept/reject step per iteration; tends to be more correlated than componentwise when the posterior is anisotropic, but easier to tune when components are correlated.

Method 3 Code

```r
rw2 <- matrix(NA, N + burn, 2)
mu0 <- ybar - 1; mu1 <- ybar + 1; tau <- 0.25
acc <- 0
for (t in 1:(N + burn)) {
  p0 <- mu0 + rnorm(1, 0, tau)
  p1 <- mu1 + rnorm(1, 0, tau)
  if (log(runif(1)) < logpost(p0, p1, y, pi_w, s0, s1) -
                          logpost(mu0, mu1, y, pi_w, s0, s1)) {
    mu0 <- p0; mu1 <- p1; acc <- acc + 1
  }
  rw2[t, ] <- c(mu0, mu1)
}
rw2     <- rw2[-(1:burn), ]
acc_rw2 <- acc / (N + burn)
```

Method 4: Gibbs Sampler with Latent Indicators

Augment with $z_i \in \{0,1\}$, where $z_i = 0$ means observation i came from component 0:

$$P(z_i = 0 \mid \mu_0, \mu_1, y_i) \;=\; \frac{\pi\,\phi(y_i \mid \mu_0, \sigma_0^2)}{\pi\,\phi(y_i \mid \mu_0, \sigma_0^2) + (1-\pi)\,\phi(y_i \mid \mu_1, \sigma_1^2)}.$$

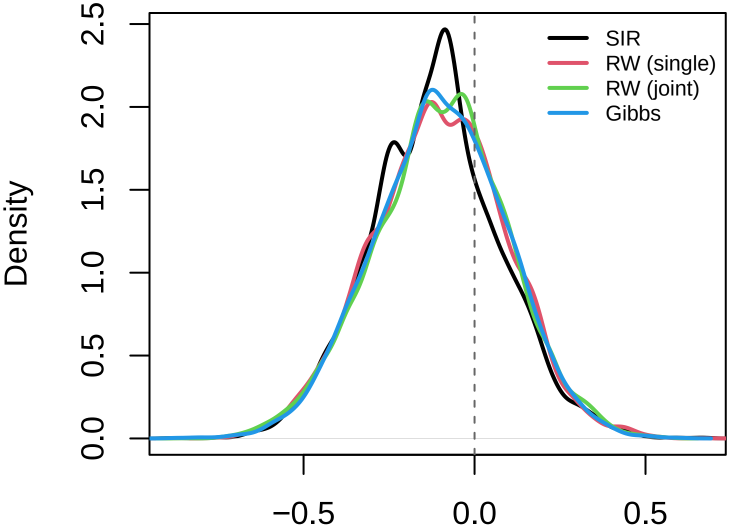
Given $\boldsymbol{z}$, let $n_k = \sum_i \mathbf{1}\{z_i = k\}$ and $\bar{y}_k = \frac{1}{n_k}\sum_{z_i=k} y_i$. Because the prior is flat,

$$\mu_k \mid \boldsymbol{z}, y \;\sim\; \mathcal{N}\Big(\bar{y}_k,\; \sigma_k^2/n_k\Big) \quad \text{truncated to } \mu_0 < \mu_1.$$

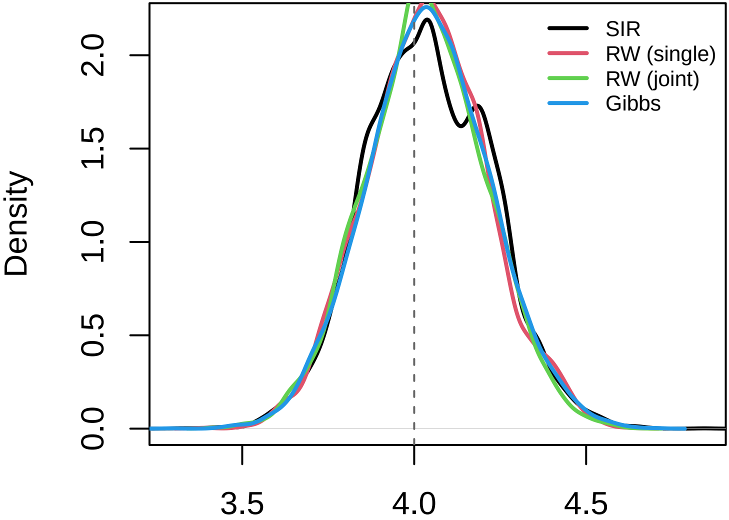
Method 4 Code

```
library(truncnorm)
gibbs <- matrix(NA, N + burn, 2)
mu0 <- ybar - 1; mu1 <- ybar + 1
for (t in 1:(N + burn)) {
  d0 <- pi_w     * dnorm(y, mu0, s0)
  d1 <- (1-pi_w) * dnorm(y, mu1, s1)
  pr0 <- d0 / (d0 + d1)
  z   <- rbinom(n, 1, 1 - pr0)              # z=0 w.p. pr0
  n0 <- sum(z == 0); n1 <- sum(z == 1)
  yb0 <- if (n0 > 0) mean(y[z == 0]) else ybar
  yb1 <- if (n1 > 0) mean(y[z == 1]) else ybar
  sd0 <- if (n0 > 0) s0/sqrt(n0) else 10
  sd1 <- if (n1 > 0) s1/sqrt(n1) else 10
  mu0 <- rtruncnorm(1, a = -Inf, b = mu1, mean = yb0, sd = sd0)
  mu1 <- rtruncnorm(1, a = mu0,  b = Inf, mean = yb1, sd = sd1)
  gibbs[t, ] <- c(mu0, mu1)
}
gibbs <- gibbs[-(1:burn), ]
```

Posterior Kernel Densities: μ_0



Posterior Kernel Densities: μ_1

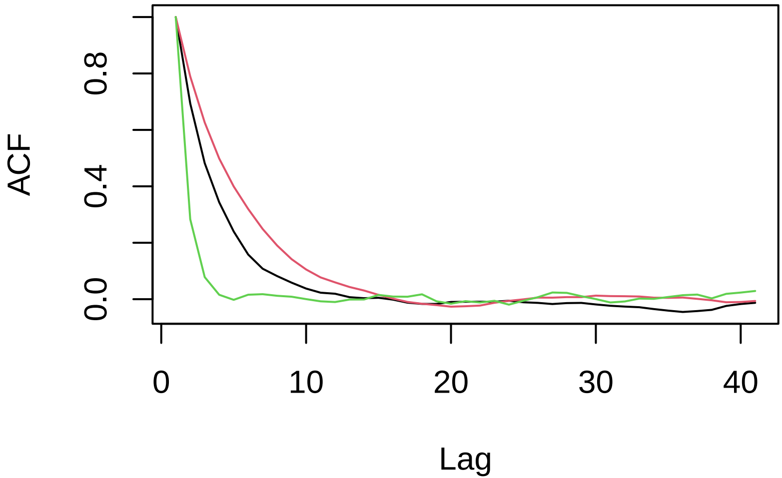


Posterior Summaries

Table 1: Posterior means and standard deviations

	mean(mu1)	mean(mu1)	sd(mu0)	sd(mu1)
SIR	-0.114	4.043	0.189	0.181
RWS	-0.103	4.038	0.197	0.176
RWJ	-0.099	4.033	0.196	0.175
Gibbs	-0.100	4.042	0.194	0.179

ACF for chains of μ_0



RW Single, {RW joint} and {Gibbs}.

Effective Sample Sizes (MCMC)

Table 2: Approximate effective sample sizes (out of 20,000)

	ESS(mu0)	ESS(mu1)
RWS	3822	4092
RWJ	1250	1344
Gibbs	5700	4580

Acceptance rates: RW single ≈ 0.58, 0.54; RW joint ≈ 0.43.

Take-aways

- **SIR** is simple and embarrassingly parallel, but quality depends entirely on the proposal g; weight degeneracy worsens in higher dimensions.
- **RW Metropolis (singe move)** is the easiest MCMC; needs careful step-size tuning and exhibits stronger autocorrelation when μ_0, μ_1 are correlated under the posterior.
- **RW Metropolis (joint move)** can adapt to the joint geometry but reduces acceptance with dimension.
- **Gibbs with latent indicators** exploits the natural conditional conjugacy of mixtures and typically delivers the **lowest autocorrelation** and the **highest ESS** per iteration.

The same machinery scales to $(\pi, \sigma_0, \sigma_1)$ being random as well, using priors $\pi \sim$ Beta, $\sigma_k^2 \sim$ Inv-Gamma.