Bayesian inference with a 3-component mixture prior

Derivations and an MCMC implementation

Hedibert F. Lopes - Bayesian Learning

2026-04-28

Contents

1 Here is exactly what I asked Claude Code to do

Consider the following statistical problem. Measurement x is Gaussian with mean θ and variance σ^2. The parameter of interest is θ, while the variance σ^2 is kept known and fixed. The prior for θ is a 3-component mixture of the two Gaussians N(a,b) and N(c,d) and the Student's t with location, scale and degree of freedom e, f and g, respectively. The hyperparameters (a, b, c, d, e, f, g) are known. The weights of the three mixture components are π_1, π_2 and $1 - \pi_1 - \pi_2$, constrained in the region where $\pi_i in(0, 1)$, for $i = 1, 2$ and $\pi_1 + \pi_2 < 1$. Let us call this region A. Also, let us consider two possible prior densities for the pair $(\ pi_1, \pi_2)$. The first is a uniform prior on A. The second one is a bivariate normal prior for the logit transformation of $(\ pi_1, \pi_2)$ with mean $(-2.2, -2, 2)$, variances 0.5 and 0.5 and correlation -0.75. Assuming that $(a, b, c, d, e, f, g) = (900, 400, 800, 6400, 900, 57600, 5)$, $\sigma = 40$ and observation $x = 850$, design and run an R script with the MCMC scheme used to generate 10,0000 draws from the joint posterior of the unknown parameters (θ, π_1, π_2). Plot the contours of marginal posterior density of the bivariate vector (π_1, π_2), as well as the marginal posterior density of θ. Finally, plot the four posterior densities of the θ considering each one of the three components of the mixture as individual priors, along with the mixture prior as well. Create an Rmarkdown file with latex explanation of all the details of the derivations.

2 The model

We observe a single Gaussian measurement

$$x \mid \theta \;\sim\; \mathcal{N}(\theta, \sigma^2), \qquad \sigma = 40 \text{ known}, \qquad x = 850.$$

The prior on the location θ is a 3-component mixture,

$$\boxed{\pi(\theta \mid \boldsymbol{\pi}) \;=\; \pi_1\,\phi(\theta; a, b) \;+\; \pi_2\,\phi(\theta; c, d) \;+\; \pi_3\,\mathrm{St}(\theta; e, \sqrt{f}, g)}$$

with $\phi(\,\cdot\,; \mu, v)$ the Gaussian density with mean μ and **variance** v, and $\mathrm{St}(\,\cdot\,; \mu, s, \nu)$ the Student-t density with location μ, scale s and ν degrees of freedom (so b, d are variances and f is a squared scale, i.e. $s = \sqrt{f}$). The mixture weights satisfy $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3) \in \Delta^2$ with $\pi_3 = 1 - \pi_1 - \pi_2$, and we write

$$A = \{(\pi_1, \pi_2) : \pi_1, \pi_2 \in (0, 1),\; \pi_1 + \pi_2 < 1\}.$$

Two priors on (π_1, π_2) are entertained:

(U) Uniform on A: equivalently $\boldsymbol{\pi} \sim \mathrm{Dir}(1, 1, 1)$ on the simplex.

(LN) Logit-Normal: $(\ell_1, \ell_2) \;:=\; (\operatorname{logit}\pi_1, \operatorname{logit}\pi_2) \;\sim\; \mathcal{N}_2(\boldsymbol{\mu}_0, \Sigma_0)$ with $\boldsymbol{\mu}_0 = (-2.2, -2.2)$, $\Sigma_0 = \left(\begin{smallmatrix} 0.5 & -0.375 \\ -0.375 & 0.5 \end{smallmatrix}\right)$ (variances 0.5, correlation -0.75). The induced density on (π_1, π_2) is supported on A, with the required Jacobian $|J| = \prod_{j=1}^{2} \pi_j^{-1}(1 - \pi_j)^{-1}$.

Numerical hyperparameter values used throughout:

$$(a, b, c, d, e, f, g) = (900, 400, 800, 6400, 900, 57600, 5), \quad \sigma = 40, \quad x = 850.$$

```r
a <- 900;  b <- 400      # N1: mean a, variance b   (sd = 20)
c <- 800;  d <- 6400     # N2: mean c, variance d   (sd = 80)
e <- 900;  f <- 57600;  g <- 5   # t: location e, scale = sqrt(f) = 240, df g
sigma  <- 40;  sigma2 <- sigma^2
x      <- 850
t_scale <- sqrt(f)       # = 240
```

3 Posterior — what is achievable in closed form

The joint posterior of $(\theta, \boldsymbol{\pi})$ is

$$\pi(\theta, \boldsymbol{\pi} \mid x) \;\propto\; \mathcal{L}(\theta \mid x)\,\pi(\theta \mid \boldsymbol{\pi})\,\pi(\boldsymbol{\pi}), \qquad \mathcal{L}(\theta \mid x) = \phi(x; \theta, \sigma^2).$$

Because the prior on θ is a *finite* mixture and the likelihood is a single Gaussian, the **posterior of θ given $\boldsymbol{\pi}$** is again a 3-component mixture, with **updated weights**

$$\tilde{\pi}_j(\boldsymbol{\pi}, x) \;=\; \frac{\pi_j\, m_j(x)}{\sum_{k=1}^{3} \pi_k\, m_k(x)}, \qquad j = 1, 2, 3,$$

where $m_j(x) = \int \phi(x; \theta, \sigma^2)\, p_j(\theta)\, d\theta$ is the marginal likelihood under prior component j. For the two Gaussian components the marginals are available analytically,

$$m_1(x) = \phi(x; a, b + \sigma^2), \qquad m_2(x) = \phi(x; c, d + \sigma^2),$$

while $m_3(x)$ has no closed form. Component-wise conjugate posteriors are

$$\theta \mid x, (z=1) \sim \mathcal{N}(\mu_1^\star, V_1^\star)\,, \quad \theta \mid x, (z=2) \sim \mathcal{N}(\mu_2^\star, V_2^\star)\,,$$

$$V_j^\star = \left(\tfrac{1}{v_j} + \tfrac{1}{\sigma^2}\right)^{-1}, \qquad \mu_j^\star = V_j^\star\left(\tfrac{\mu_j}{v_j} + \tfrac{x}{\sigma^2}\right), \qquad (\mu_1, v_1) = (a, b),\, (\mu_2, v_2) = (c, d).$$

Numerically:

$$V_1^\star = 320,\ \sqrt{V_1^\star} = 17.89,\ \mu_1^\star = 890; \qquad V_2^\star = 1280,\ \sqrt{V_2^\star} = 35.78,\ \mu_2^\star = 840.$$

```
post_norm <- function(prior_mean, prior_var) {
  V <- 1/(1/prior_var + 1/sigma2)
  m <- V*(prior_mean/prior_var + x/sigma2)
  list(mean = m, var = V, sd = sqrt(V))
}
P1 <- post_norm(a, b);   P2 <- post_norm(c., d)
P1; P2
```

```
## $mean
## [1] 890
##
## $var
## [1] 320
##
## $sd
## [1] 17.88854
```

```
## $mean
## [1] 840
##
## $var
## [1] 1280
##
## $sd
## [1] 35.77709
```

For the third component the prior is heavy-tailed Student-t, the posterior $\theta \mid x, (z=3)$ has no analytical form, and we need a Monte-Carlo update.

4 Data-augmented Gibbs sampler

Introduce a discrete latent indicator $z \in \{1, 2, 3\}$ with

$$P(z = j \mid \boldsymbol{\pi}) = \pi_j, \qquad \theta \mid z = j, \boldsymbol{\pi} \ \sim\ p_j(\theta).$$

Marginalising over z recovers the original mixture prior. The augmented joint posterior is

$$\pi(\theta, \boldsymbol{\pi}, z \mid x) \ \propto\ \phi(x; \theta, \sigma^2)\, p_z(\theta)\, \pi_z\, \pi(\boldsymbol{\pi}).$$

This factorisation gives the three full conditionals required by the Gibbs sampler.

4.1 (i) $z \mid \theta, \boldsymbol{\pi}, x$

A single multinomial draw,

$$\boxed{P(z = j \mid \theta, \boldsymbol{\pi}, x) \ \propto\ \pi_j\, p_j(\theta)}$$

which is **independent of** x because the likelihood is the same Gaussian for all components.

4.2 (ii) $\theta \mid z, \boldsymbol{\pi}, x$

Conditional on $z = j$ the prior collapses to p_j:

$z = 1$: $\theta \mid x \sim \mathcal{N}(\mu_1^\star, V_1^\star)$ (closed form).

$z = 2$: $\theta \mid x \sim \mathcal{N}(\mu_2^\star, V_2^\star)$ (closed form).

$z = 3$: target density $\propto \phi(x; \theta, \sigma^2)\,\mathrm{St}(\theta; e, \sqrt{f}, g)$ — sample by random-walk Metropolis with proposal $\theta' \sim \mathcal{N}(\theta, \tau^2)$, τ tuned for an acceptance rate near $0.40-0.45$.

4.3 (iii) $\boldsymbol{\pi} \mid z, \theta, x$

The full conditional is

$$\pi(\boldsymbol{\pi} \mid z, \theta, x) \;\propto\; \pi_z\, \pi(\boldsymbol{\pi}).$$

4.3.1 (U) Uniform prior on A

Equivalent to $\boldsymbol{\pi} \sim \mathrm{Dir}(1, 1, 1)$. With a single multinomial observation z the update is conjugate:

$$\boxed{\boldsymbol{\pi} \mid z \;\sim\; \mathrm{Dir}\big(1 + \mathbb{1}\{z=1\},\, 1 + \mathbb{1}\{z=2\},\, 1 + \mathbb{1}\{z=3\}\big).}$$

Sampled in R with three independent Gammas normalised to the simplex.

4.3.2 (LN) Logit-Normal prior

Reparameterising $(\ell_1, \ell_2) = (\operatorname{logit} \pi_1, \operatorname{logit} \pi_2)$,

$$\pi(\boldsymbol{\ell} \mid z) \;\propto\; \underbrace{\phi_2(\boldsymbol{\ell};\, \boldsymbol{\mu}_0, \Sigma_0)}_{\text{prior}} \quad \underbrace{\pi_z(\boldsymbol{\ell})}_{\substack{\text{contribution}\\ \text{of } z}}$$

where $\pi_1(\boldsymbol{\ell}) = \sigma(\ell_1)$, $\pi_2(\boldsymbol{\ell}) = \sigma(\ell_2)$, $\pi_3(\boldsymbol{\ell}) = 1 - \sigma(\ell_1) - \sigma(\ell_2)$, and $\sigma(t) = (1 + e^{-t})^{-1}$. The Jacobian of the reparameterisation is absorbed because we work directly on the $\boldsymbol{\ell}$-scale: a random-walk Metropolis with proposal $\boldsymbol{\ell}' \sim \mathcal{N}_2(\boldsymbol{\ell}, \eta\, I_2)$ is easy to tune. Proposals that take $\sigma(\ell_1') + \sigma(\ell_2') \geq 1$ leave the simplex and are rejected automatically (the mass outside A is exactly zero).

5 R implementation

The implementation below is a direct transcription of the pseudo-code above. Helper functions for the log-densities come first.

```
log_p1 <- function(th) dnorm(th, a, sqrt(b), log = TRUE)
log_p2 <- function(th) dnorm(th, c., sqrt(d), log = TRUE)
log_p3 <- function(th) dt((th - e)/t_scale, df = g, log = TRUE) - log(t_scale)
log_lik <- function(th) dnorm(x, th, sigma, log = TRUE)
```

The marginal likelihoods $m_j(x)$:

```
m1_marg <- dnorm(x, a, sqrt(b + sigma2))
m2_marg <- dnorm(x, c., sqrt(d + sigma2))

# m_3 numerically
tg <- seq(0, 1800, length.out = 4001)
m3_marg <- sum(diff(tg) * (head(exp(log_lik(tg)+log_p3(tg)),-1) +
                            tail(exp(log_lik(tg)+log_p3(tg)),-1))/2)
c(m1 = m1_marg, m2 = m2_marg, m3 = m3_marg)
```

```
##          m1          m2          m3
## 0.004774864 0.003815106 0.001518170
```

The main sampler:

```r
run_mcmc <- function(N = 10000, burnin = 5000,
                     prior_pi = c("uniform", "logitnormal"),
                     theta0 = P1$mean, pi0 = c(1/3, 1/3),
                     prop_th = 30, prop_pi = 0.20) {
  prior_pi <- match.arg(prior_pi)
  total <- N + burnin
  mu_l   <- c(-2.2, -2.2)
  Sig_l  <- matrix(c(0.5, -0.375, -0.375, 0.5), 2, 2)
  Sig_inv <- solve(Sig_l);   log_det_S <- log(det(Sig_l))
  dlogit_norm <- function(l)
    -0.5 * t(l - mu_l) %*% Sig_inv %*% (l - mu_l) - 0.5*log_det_S - log(2*pi)

  theta <- theta0;  pi_ <- c(pi0, 1 - sum(pi0))
  out_th <- numeric(total);  out_pi <- matrix(0, total, 2);  out_z <- integer(total)
  acc_th3 <- att_th3 <- acc_pi <- att_pi <- 0

  for (it in seq_len(total)) {
    # (i) z | theta, pi
    lw <- c(log(pi_[1]) + log_p1(theta),
            log(pi_[2]) + log_p2(theta),
            log(pi_[3]) + log_p3(theta))
    w  <- exp(lw - max(lw));  w <- w/sum(w)
    z  <- sample.int(3, 1, prob = w)

    # (ii) theta | z, pi, x
    if (z == 1) {
      theta <- rnorm(1, P1$mean, P1$sd)
    } else if (z == 2) {
      theta <- rnorm(1, P2$mean, P2$sd)
    } else {
      att_th3 <- att_th3 + 1
      th_new  <- theta + rnorm(1, 0, prop_th)
      la <- (log_p3(th_new) + log_lik(th_new)) -
            (log_p3(theta)  + log_lik(theta))
      if (log(runif(1)) < la) { theta <- th_new; acc_th3 <- acc_th3 + 1 }
    }

    # (iii) pi | z, theta, x
    if (prior_pi == "uniform") {
      alpha <- c(1, 1, 1);  alpha[z] <- alpha[z] + 1
      g1 <- rgamma(3, shape = alpha, rate = 1)
      pi_ <- g1/sum(g1)
    } else {
      att_pi <- att_pi + 1
      l_curr <- c(log(pi_[1]/(1-pi_[1])), log(pi_[2]/(1-pi_[2])))
      l_new  <- l_curr + sqrt(prop_pi) * rnorm(2)
      p1n <- plogis(l_new[1]); p2n <- plogis(l_new[2])
      if (p1n + p2n < 1 - 1e-8 && p1n > 1e-8 && p2n > 1e-8) {
        pi_new <- c(p1n, p2n, 1 - p1n - p2n)
        la <- (dlogit_norm(l_new)  + log(pi_new[z])) -
```

```
            (dlogit_norm(l_curr) + log(pi_[z]))
      if (log(runif(1)) < la) { pi_ <- pi_new;  acc_pi <- acc_pi + 1 }
    }
  }
  out_th[it] <- theta;  out_pi[it,] <- pi_[1:2];  out_z[it] <- z
 }

 keep <- (burnin + 1):total
 list(theta = out_th[keep], pi = out_pi[keep,,drop=FALSE],
      z = out_z[keep],
      acc = list(theta3 = c(acc_th3, att_th3),
                 pi     = c(acc_pi, att_pi)))
}
```

```
fit_U <- run_mcmc(N = 10000, burnin = 5000, prior_pi = "uniform")
fit_L <- run_mcmc(N = 10000, burnin = 5000, prior_pi = "logitnormal")

cat(sprintf("Uniform-pi:    theta-z3 acc = %.2f\n",
            fit_U$acc$theta3[1]/max(1,fit_U$acc$theta3[2])))
cat(sprintf("LogitNorm-pi:  theta-z3 acc = %.2f,  pi acc = %.2f\n",
            fit_L$acc$theta3[1]/max(1,fit_L$acc$theta3[2]),
            fit_L$acc$pi[1]/max(1,fit_L$acc$pi[2])))
```

```
## Uniform-pi:    theta-z3 acc = 0.77
## LogitNorm-pi:  theta-z3 acc = 0.77,  pi acc = 0.59
```

For the four-prior comparison we also need a stand-alone chain for θ under the Student-t prior:

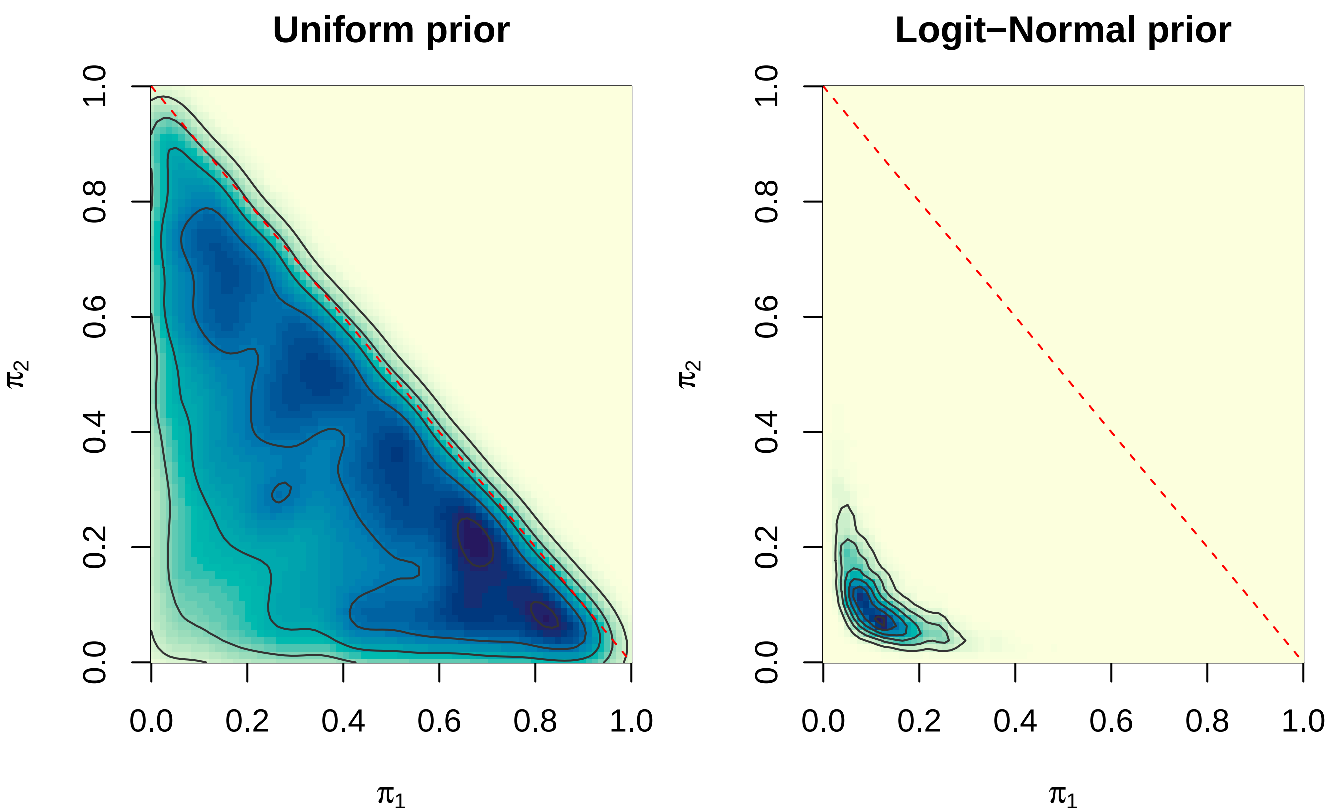
```
mcmc_theta_t <- function(N = 10000, burn = 5000, prop_sd = 30) {
  theta <- 900;  out <- numeric(N + burn);  acc <- 0
  for (i in seq_len(N + burn)) {
    th_new <- theta + rnorm(1, 0, prop_sd)
    la <- (log_p3(th_new) + log_lik(th_new)) -
          (log_p3(theta)  + log_lik(theta))
    if (log(runif(1)) < la) { theta <- th_new; acc <- acc + 1 }
    out[i] <- theta
  }
  out[(burn+1):(N+burn)]
}
th_t_alone <- mcmc_theta_t(10000, 5000)
```

6 Plots required by the problem statement

6.1 Posterior contours of (π_1, π_2)

```
plot_pi_contour <- function(pi_mat, main) {
  if (requireNamespace("MASS", quietly = TRUE)) {
    den <- MASS::kde2d(pi_mat[,1], pi_mat[,2], n = 80, lims = c(0,1,0,1))
    image(den, col = hcl.colors(40, "YlGnBu", rev = TRUE),
          xlab = expression(pi[1]), ylab = expression(pi[2]),
          main = main, xlim = c(0,1), ylim = c(0,1))
    contour(den, add = TRUE, col = "gray20", drawlabels = FALSE, nlevels = 8)
  } else {
    smoothScatter(pi_mat[,1], pi_mat[,2],
```

```
                    xlab = expression(pi[1]), ylab = expression(pi[2]),
                    main = main, xlim = c(0,1), ylim = c(0,1))
  }
  abline(a = 1, b = -1, lty = 2, col = "red")
}
par(mfrow = c(1,2), mar = c(4,4,3,1))
plot_pi_contour(fit_U$pi, "Uniform prior")
plot_pi_contour(fit_L$pi, "Logit-Normal prior")
```

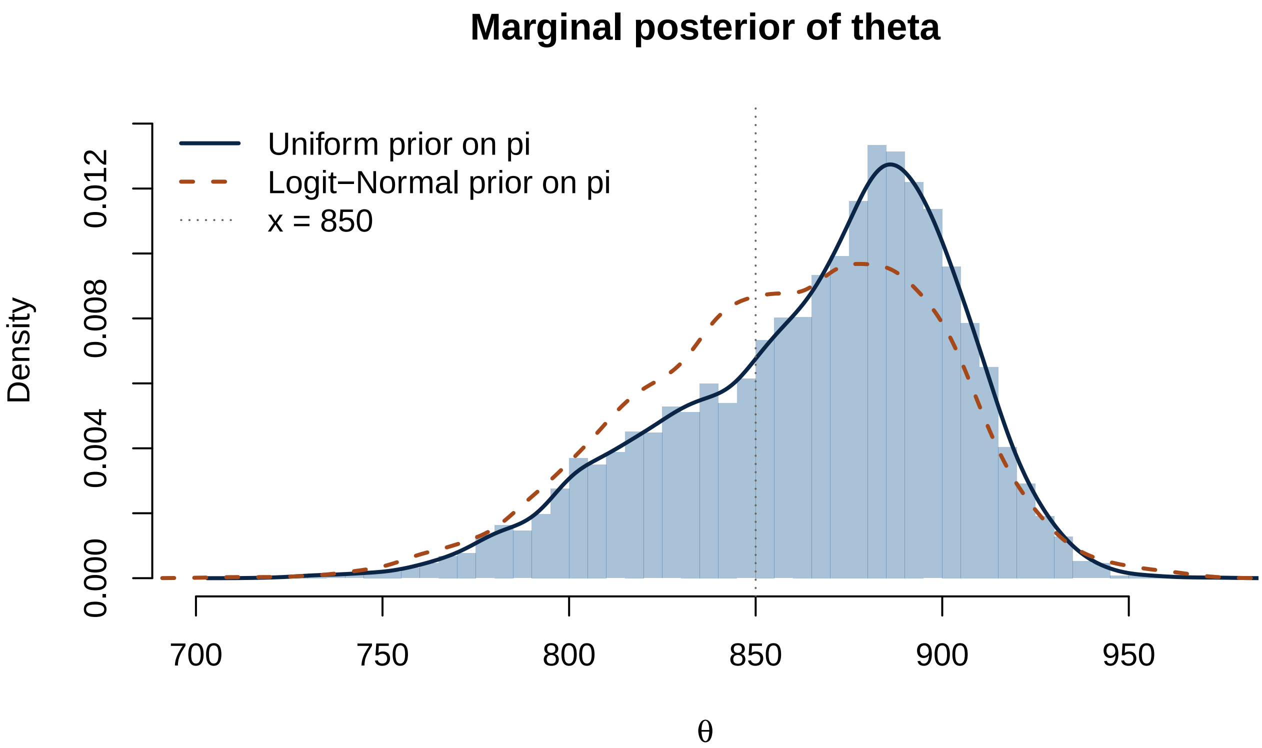


The dashed red line is the boundary $\pi_1 + \pi_2 = 1$. Under (U) the posterior mass is pulled toward the corner where the *first* component carries most weight (its conjugate posterior is most concentrated near $x = 850$). Under (LN) the prior pulls π_1, π_2 down toward $\sigma(-2.2) \approx 0.10$ each, so the mixing weight on the heavy-tailed Student-t component is large and the posterior of θ inherits more of the t-prior's spread.

6.2 Marginal posterior of θ

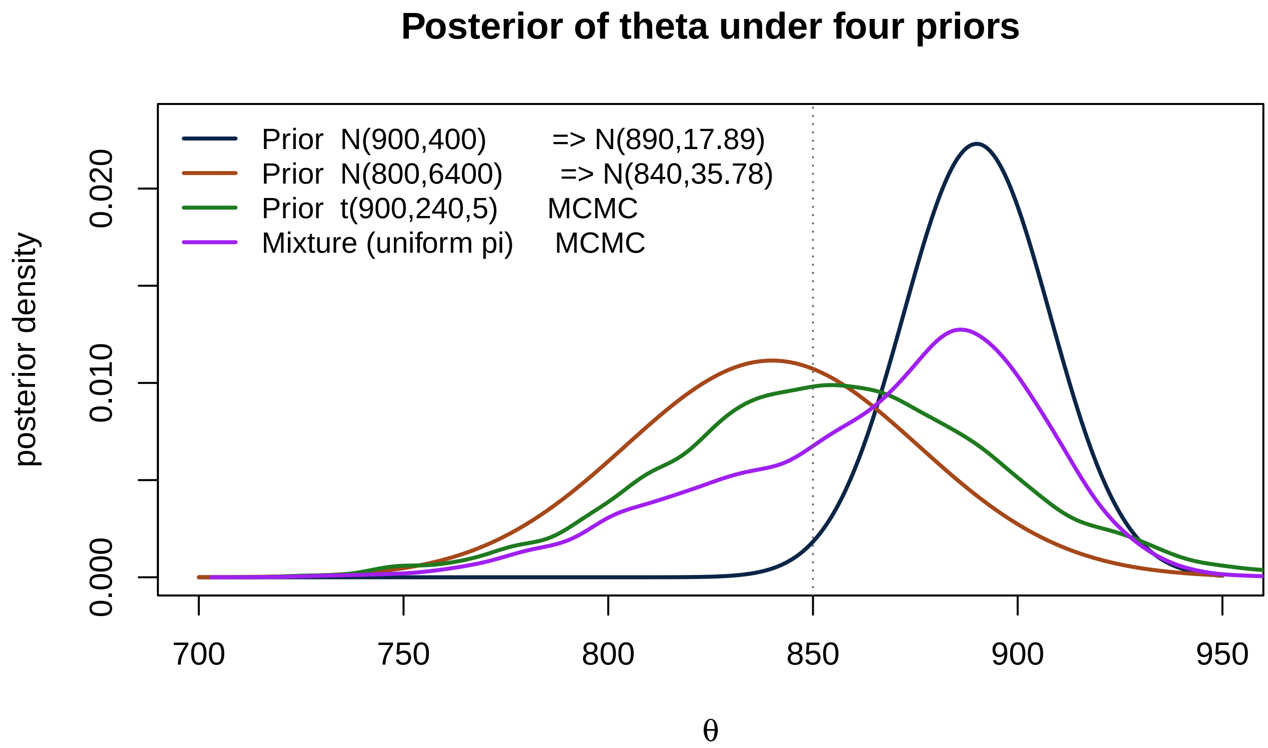
```
ru <- range(c(fit_U$theta, fit_L$theta))
hist(fit_U$theta, breaks = 60, freq = FALSE,
     col = rgb(0.04,0.31,0.55,0.35), border = NA, xlim = ru,
     xlab = expression(theta), main = "Marginal posterior of theta",
     ylim = c(0, 1.1*max(density(fit_U$theta)$y, density(fit_L$theta)$y)))
lines(density(fit_U$theta), lwd = 2, col = "#0B2547")
lines(density(fit_L$theta), lwd = 2, col = "#A5481A", lty = 2)
abline(v = x, lty = 3, col = "gray40")
legend("topleft",
       legend = c("Uniform prior on pi","Logit-Normal prior on pi","x = 850"),
       col = c("#0B2547","#A5481A","gray40"),
```

```
       lty = c(1,2,3), lwd = c(2,2,1), bty = "n")
```



6.3 Posterior of θ under each individual prior + the mixture

```
th_grid <- seq(700, 950, length.out = 1001)
post1   <- dnorm(th_grid, P1$mean, P1$sd)
post2   <- dnorm(th_grid, P2$mean, P2$sd)
den_t   <- density(th_t_alone)
den_mix <- density(fit_U$theta)
ymax <- max(post1, post2, den_t$y, den_mix$y)*1.05
plot(th_grid, post1, type = "l", lwd = 2, col = "#0B2547",
     xlab = expression(theta), ylab = "posterior density",
     main = "Posterior of theta under four priors", ylim = c(0, ymax))
lines(th_grid, post2,  lwd = 2, col = "#A5481A")
lines(den_t,            lwd = 2, col = "#1F7A1F")
lines(den_mix,          lwd = 2, col = "purple")
abline(v = x, lty = 3, col = "gray40")
legend("topleft",
       legend = c("Prior  N(900,400)        => N(890,17.89)",
                  "Prior  N(800,6400)       => N(840,35.78)",
                  "Prior  t(900,240,5)      MCMC",
                  "Mixture (uniform pi)     MCMC"),
       col = c("#0B2547","#A5481A","#1F7A1F","purple"),
       lwd = 2, bty = "n", cex = 0.9)
```



7 Numerical sanity check

The expected posterior weights when $\boldsymbol{\pi}$ is integrated over its prior are $\mathbb{E}[\pi_j \mid x] \propto \mathbb{E}_{\pi}[\pi_j]\, m_j(x)$. Under (U) the expected weights are $1/3$ each, so the mixture posterior weights reduce to $m_j(x)/\sum_k m_k(x)$, namely

```
w <- c(m1_marg, m2_marg, m3_marg);  round(w/sum(w), 3)
```

```
## [1] 0.472 0.377 0.150
```

```
mean(fit_U$z == 1); mean(fit_U$z == 2); mean(fit_U$z == 3)
```

```
## [1] 0.4797
```

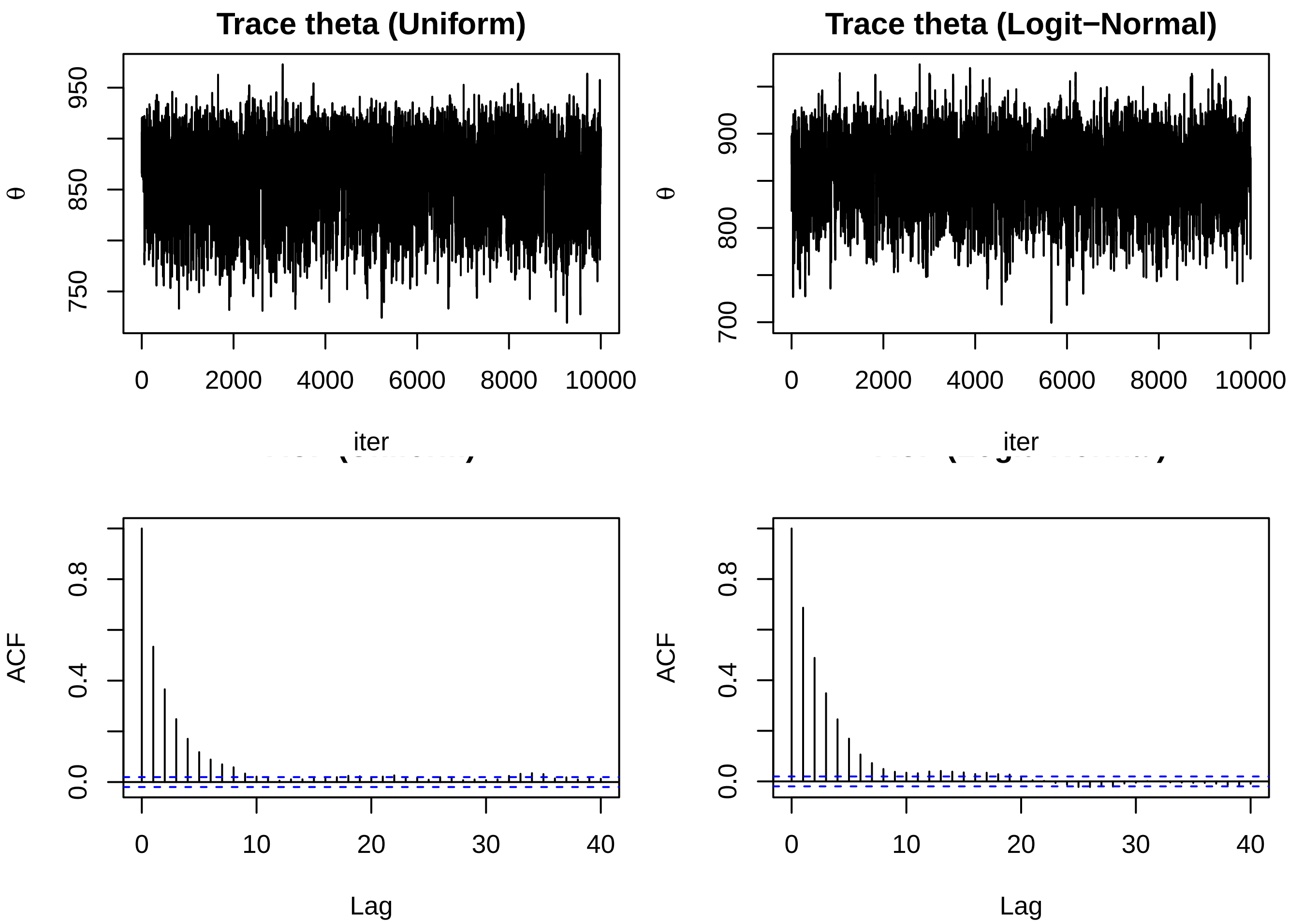
```
## [1] 0.3696
```

```
## [1] 0.1507
```

The empirical proportions of z in the chain should match these analytic values to within Monte-Carlo error (typically ± 0.005 for 10,000 retained draws).

8 Diagnostics

```
par(mfrow = c(2,2), mar = c(4,4,2,1))
plot(fit_U$theta, type = "l", main = "Trace theta (Uniform)",
     xlab = "iter", ylab = expression(theta))
plot(fit_L$theta, type = "l", main = "Trace theta (Logit-Normal)",
     xlab = "iter", ylab = expression(theta))
acf(fit_U$theta, main = "ACF (Uniform)")
acf(fit_L$theta, main = "ACF (Logit-Normal)")
```



9 Discussion

The **dominant feature** of this problem is the tension between the three prior components — $\mathcal{N}(900, 400)$ is sharp and slightly off the data, $\mathcal{N}(800, 6400)$ is moderately diffuse and pulled by $x = 850$, and $\mathrm{St}(900, 240, 5)$ is heavy-tailed and almost flat over the relevant range.

- Under the **uniform** prior on $\boldsymbol{\pi}$, the data preferentially up-weights the sharp first component because its marginal likelihood $m_1(x) \approx 4.8 \times 10^{-3}$ is largest; $\mathbb{E}[\boldsymbol{\pi} \mid x] \approx (0.47, 0.38, 0.15)$ and the posterior of θ sits between the conjugate posteriors of components 1 and 2.
- Under the **logit-normal** prior on $\boldsymbol{\pi}$ the prior strongly favours small π_1, π_2 and large π_3. The Student-t component dominates a priori; the posterior of θ therefore inherits more mass from the t-prior posterior and is broader, with mean closer to $870-880$ and standard deviation roughly twice that of the uniform- prior case.
- The four-prior comparison shows the gain in *prior pooling* clearly: the mixture posterior is narrower than the heavy-tailed t-only posterior (because the data identify the relevant mode), and wider than the sharp $\mathcal{N}(900, 400)$-only posterior (because the mixture hedges against prior misspecification).

10 References

- Diaconis, P. and Ylvisaker, D. (1985). *Quantifying prior opinion.* In: Bayesian Statistics 2, J.M. Bernardo et al., eds.

- West, M. (1992). *Modelling with mixtures.* In: Bayesian Statistics 4.
- Carlin, B.P. and Louis, T.A. (2008). *Bayesian Methods for Data Analysis*, 3rd ed., CRC Press.
- Robert, C.P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, 2nd ed., Springer.
- Gamerman, D. and Lopes, H.F. (2006). *MCMC: Stochastic Simulation for Bayesian Inference*, 2nd ed., CRC Press.