$$L(\theta, \mu, \tau^2; y)\, p(\quad)$$

$$\left(\prod_j p(y_j \mid \theta_j)\, p(\theta_j \mid \mu, \tau)\right]\, p(\mu, \tau)\Big)$$

$$\mu \mid \theta_1, \ldots, \theta_J \sim N\left(\bar{\theta}, \frac{\tau^2}{n}\right)$$

$$\theta_j \mid y_1, \ldots, y_J, \theta_{-j}, \mu, \tau$$

$$\theta_j \sim N\left(w y_j + (1-w)\mu, \frac{1}{1/\sigma_j^2 + 1/\tau^2}\right)$$

Gibbs Sampler

- Pro: proposals, p($\theta^{(t+1)} \mid \theta^{(t)}$), are **never** rejected.
- Pro: don't need to choose the proposal density distribution or tune parameters of the density
- Con: it can be difficult to derive the full-conditional distributions (unless, for example, the prior distributions are chosen to be conjugate)
- Con: when the parameters of the posterior distribution are highly correlated:
  - Autocorrelation of the samples is high
  - Effective sample size is low

Gibbs: a special case of MH

Gibbs Sampler: Finite Mixture Models

Reminder: mixture of binomials example

```r
set.seed(123)
n <- 25
phi <- 0.4
size <- 10
z <- ifelse(rbinom(100, 1, phi), "Experience", "No Experience")
y <- rbinom(n, size=size, prob=ifelse(z=="Experience", 0.7, 0.3))
```

$P(\phi, \theta_0, \theta_1 \mid y_1, \ldots, y_{25})$

$$P(\phi, \theta_0, \theta_1, z_1, \ldots z_{25} \mid y_1, \ldots y_{25})$$

$z_i \sim \mathrm{Bern}(\phi)$ (experience)

$$y_i \mid z_i \sim \begin{cases} \mathrm{Bin}(10, \theta_1) & \text{if } z_i = 1 \\ \mathrm{Bin}(10, \theta_0) & \text{if } z_i = 0 \end{cases}$$

(uniform priors)

$$L(\phi, \theta_0, \theta_1, z) \propto P(\phi, \theta_0, \theta_1, z) \propto$$

$$\theta_1^{\sum z_i y_i} (1-\theta_1)^{\sum z_i (10 - y_i)} \; \theta_0^{\sum (1-z_i) y_i} (1-\theta_0)^{\sum (1-z_i)(10-y_i)}$$

$$\times \; \phi^{\sum z_i} (1-\phi)^{25 - \sum z_i}$$

1. $\theta_1 \sim \mathrm{Beta}\left(\sum z_i y_i + 1, \sum z_i (10 - y_i) + 1\right)$
2. $\theta_0 \sim \mathrm{Beta}\left(\sum (1-z_i) y_i + 1, \sum (1-z_i)(10-y_i) + 1\right)$
3. $\phi \sim \mathrm{Beta}\left(\sum z_i + 1, 25 - \sum z_i + 1\right)$
4.

$$\underline{z_i \mid \theta_1, \theta_0, \phi \text{ !}}$$

$$z_i : \quad \theta_1^{z_i y_i} (1-\theta_1)^{z_i(10-y_i)} \theta_0^{(1-z_i)y_i} \leftarrow$$

$$(1-\theta_0)^{(1-z_i)(10-y_i)} \phi^{z_i} (1-\phi)^{1-z_i}$$

$$z_i = 1 : \quad \theta_1^{y_i} (1-\theta_1)^{10-y_i} \phi$$

$$z_i = 0 : \quad \theta_0^{y_i} (1-\theta_0)^{10-y_i} (1-\phi)$$

$$z_i \sim \text{Bern}\left(\frac{\theta_1^{y_i} (1-\theta_1)^{10-y_i} \phi}{\left(\theta_1^{y_i} (1-\theta_1)^{10-y_i} \phi + \theta_0^{y_i} (1-\theta_0)^{10-y_i} (1-\phi)\right)} \right)$$

$$p(\tilde{y} \mid y_1, \ldots, y_n)$$

Gibbs Sampler: Binomial Mixture

```r
## Initialize all
run_binomial_gibbs <- function() {
experience_vec <- rbinom(n, 1, 0.5)
theta_cur <- c(0.1, 0.9)
phi_cur <- 0.5
nsamps <- 10000

theta_samps <- matrix(0, nrow=nsamps, ncol=2)
colnames(theta_samps) <- c("theta1", "theta2")
phi_samps <- numeric(nsamps)
experience_samps <- matrix(0, nrow=nsamps, ncol=n)

for(i in 1:nsamps) {

  ## Update experience indicators
  experience_prob <- (phi_cur*dbinom(y, 10, prob=theta_cur[2])) / ((1-phi_cur)*dbinom(y, 10, prob=theta_cur[1
  experience_vec <- 1*(runif(n) < experience_prob)
    ## Update mixture weights
  phi_cur <- rbeta(1, sum(experience_vec)+1, n-sum(experience_vec)+1)
  phi_samps[i] <- phi_cur

  y0 <- y[experience_vec == 0]
  n0 <- sum(experience_vec == 0)
  y1 <- y[experience_vec == 1]
  n1 <- sum(experience_vec)

```

Z

$\theta_0 = .1, \quad \theta_1 = .9$

Quantile intervals

$\theta_0 = .3, \quad \theta_1 = .7$

$\phi = .4$

```
1  quantile(gibbs_results[[1]][, "theta1"], c(0.025, 0.5, .975))
```

```
     2.5%       50%     97.5%
0.2399146 0.3738561 0.7823448
```

```
1  quantile(gibbs_results[[1]][, "theta2"], c(0.025, 0.5, .975))
```

```
     2.5%       50%     97.5%
0.3040817 0.7194903 0.8339195
```

```
1  quantile(gibbs_results[[1]][, "phi_samps"], c(0.025, 0.5, .975))
```

```
     2.5%       50%     97.5%
0.2355142 0.5119565 0.7879399
```

Rhat

```r
rstan::Rhat(cbind(gibbs_results[[1]][, "theta1"],gibbs_results[[2]][, "thet
```

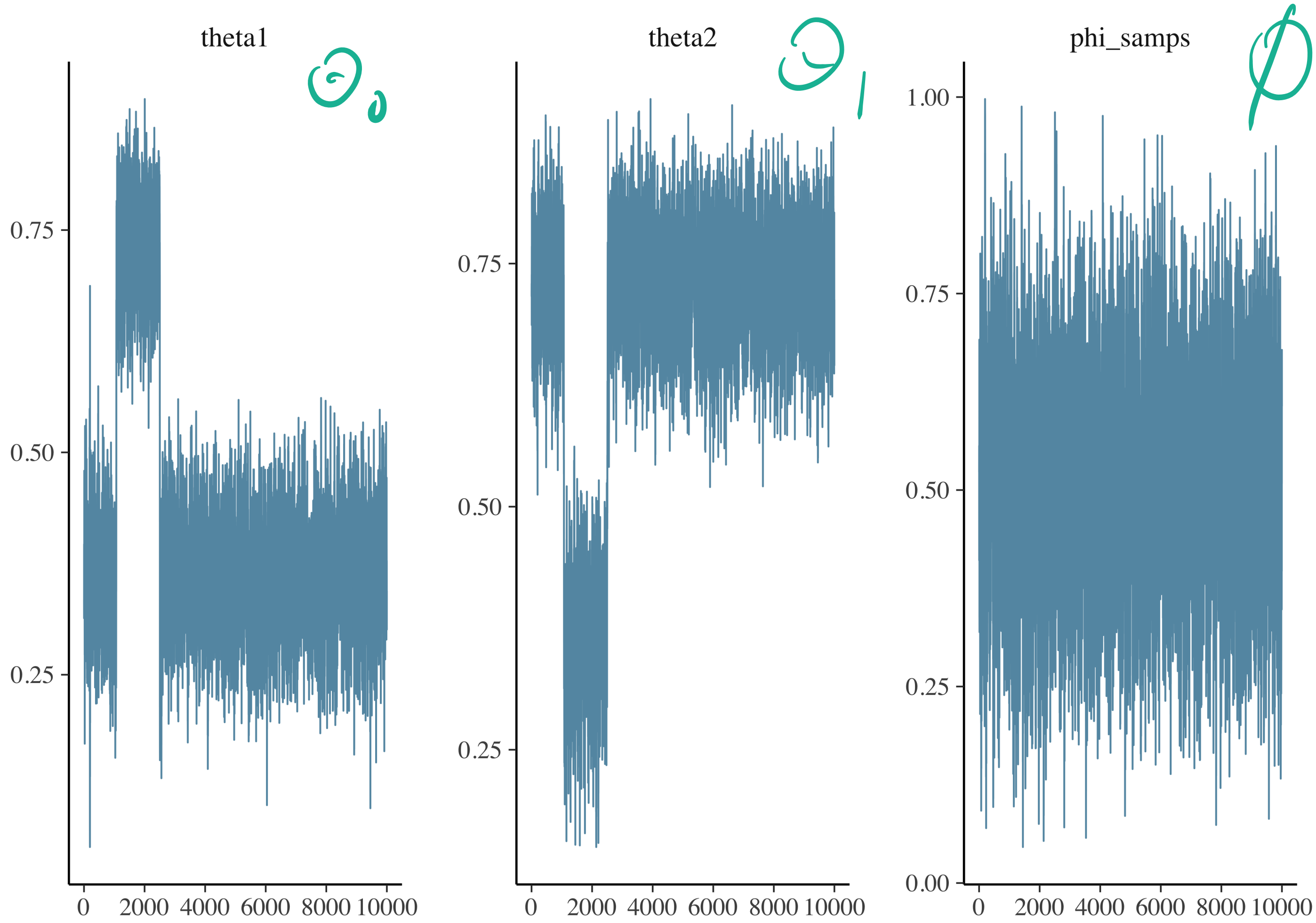
```
[1] 1.175579
```

```r
rstan::Rhat(cbind(gibbs_results[[1]][, "theta2"],gibbs_results[[2]][, "thet
```

```
[1] 1.174507
```

Traceplots

```
bayesplot::mcmc_trace(gibbs_results[[1]])
```



Correcting label switching

```r
quantile(apply(gibbs_results[[1]][, 1:2], 1, max), c(0.05, 0.5, 0.95))
```

```
       5%       50%       95% 
0.6328924 0.7315352 0.8209664 
```

```r
quantile(apply(gibbs_results[[1]][, 1:2], 1, min), c(0.05, 0.5, 0.95))
```

```
       5%       50%       95% 
0.2570862 0.3606355 0.4630786 
```

Post-processing

$$P(\theta_0, \theta_1, \phi \mid y_1 \ldots y_n) =$$

$$\prod_i^n \Big[\underbrace{\phi P(y \mid \theta_1)}_{\log} + \underbrace{(1-\phi) P(y \mid \theta_0)} \Big]$$

$$\underbrace{\log \phi + \log P(y \mid \theta_1)}_{U} \qquad \underbrace{\log(1-\phi) + \log P(y \mid \theta_0)}_{V}$$

$$\log(\exp(U) + \exp(V)) =$$

$$\log\left(\frac{\exp(U) + \exp(V)}{m} \times m \right) =$$

$$\log\left(\frac{\exp(U) + \exp(V)}{m} \right) + \log(m)$$