Multivariate Multinomial Mixture Model

Setting:

- K = 3 possible clusters (classes) (π denotes the mixing proportion)
- For each k, we sample values for three categorical (qualitative) features ($p = 3$).
- For each feature, there are 3 possible outcome ($J = 3$) and the probability of them can be characterized by the multinomial probability parameter θ_{kp} for k = 1, 2, 3 and p = 1, 2, 3
- Use Dirichlet distribution prior for π and θ_{kp} with non-informative prior

Summary on data generating process

$\pi_i \sim Dirichlet(1)$

$\theta_{kp} \sim Dirichlet(1)$

$x_{ip} \mid z_i = k, \theta_{kp} \sim Mult(1, \theta_{kp})$

```r
# Define mixing proportion
set.seed(0)
pi_true = c(0.3, 0.1, 0.6)

# Theta 1 for mixture cluster 1
theta_11_true = c(0.7,0.2,0.1)
theta_12_true = c(0.1,0.8,0.1)
theta_13_true =c(0.2,0.1,0.7)

# Theta 2 for mixture cluster 2
theta_21_true = c(0.05,0.75,0.2)
theta_22_true = c(0.2,0.15,0.65)
theta_23_true =c(0.7,0.2,0.1)

# Theta 3 for mixture cluster 3
theta_31_true = c(0.1,0.1,0.8)
theta_32_true = c(0.7,0.15,0.15)
theta_33_true =c(0.1,0.7,0.2)

# Theta row i is cluster i, column j is category j
theta_p1_true = rbind(theta_11_true, theta_21_true, theta_31_true)
theta_p2_true = rbind(theta_12_true, theta_22_true, theta_32_true)
theta_p3_true = rbind(theta_13_true, theta_23_true, theta_33_true)

# Create simulated data
n = 300
class_i = rmultinom(n, size = 1, prob = pi_true)
x1 = c()
x2 = c()
x3 = c()
for (i in 1:n) {
  x1 = cbind(x1, rmultinom(1, size = 1, prob = theta_p1_true[class_i[, i]==1,]))
  x2 = cbind(x2, rmultinom(1, size = 1, prob = theta_p2_true[class_i[, i]==1,]))
  x3 = cbind(x3, rmultinom(1, size = 1, prob = theta_p3_true[class_i[, i]==1,]))
}
```

```r
# Prior Parameter
cluster_num = 3
category_num = 3
a_pi = rep(1,cluster_num) # For Dirichlet distribution for pi
a_theta = rep(1,category_num) # For Dirichlet distribution for theta
```

Blocked Gibbs Sampling here consists of three major steps (corresponding to 3 parameters of interest z_i, θ_p, π)

1. Sample cluster indicator z_i for each x_i from full conditional multinomial distribution

For each i:

$P(z_i = k \mid x_i, \theta, \pi) = \frac{\pi_k \Pi_p \Pi_j P(x_{ipj}|z_i=k)^{x_{ipj}}}{\sum_k \pi_k \Pi_p \Pi_j P(x_{ipj}|z_i=k)^{x_{ipj}}}$

2. Sample θ_{kp} from the updated (posterior) Dirichlet distribution for each of the 3 clusters

For each k:

$\theta_{kp} \mid x, z \sim Dirichlet(1 + n_{kp1}, 1 + n_{kp2}, 1 + n_{kp3})$

where n_{kpi} is the number of observations found in cluster k in features p that is of category i

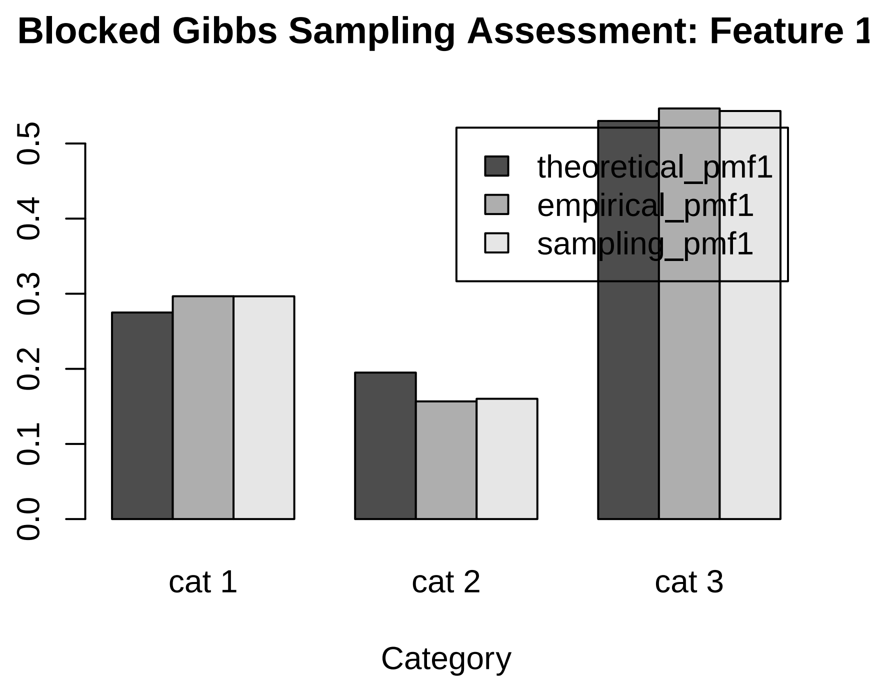
3. Sample π from the updated (posterior) Dirichlet distribution to obtain the new mixing proportion

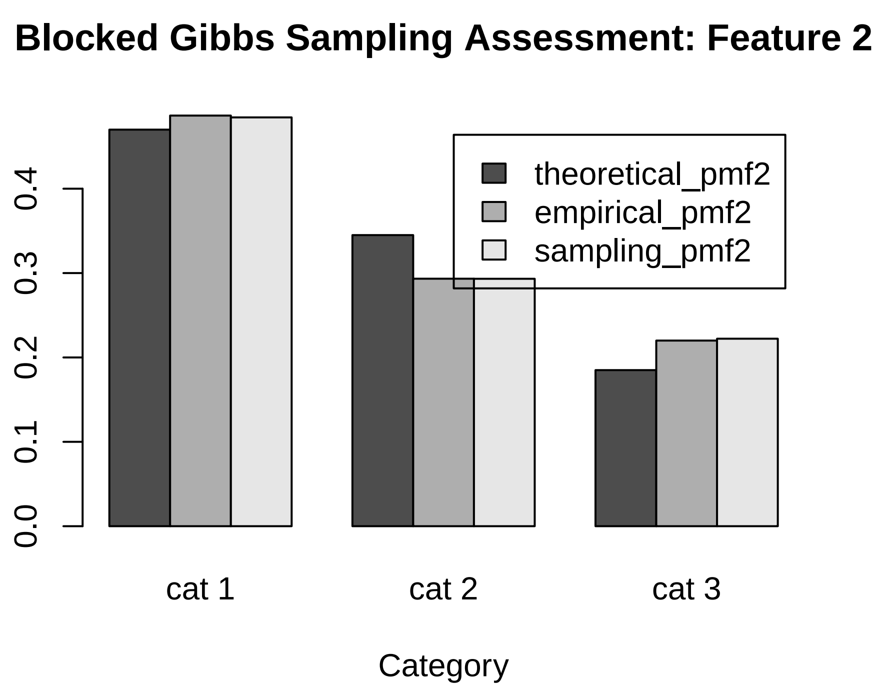
$\pi \mid x, z \sim Dirichlet(1 + n_1, 1 + n_2, 1 + n_3, 1 + n_4)$

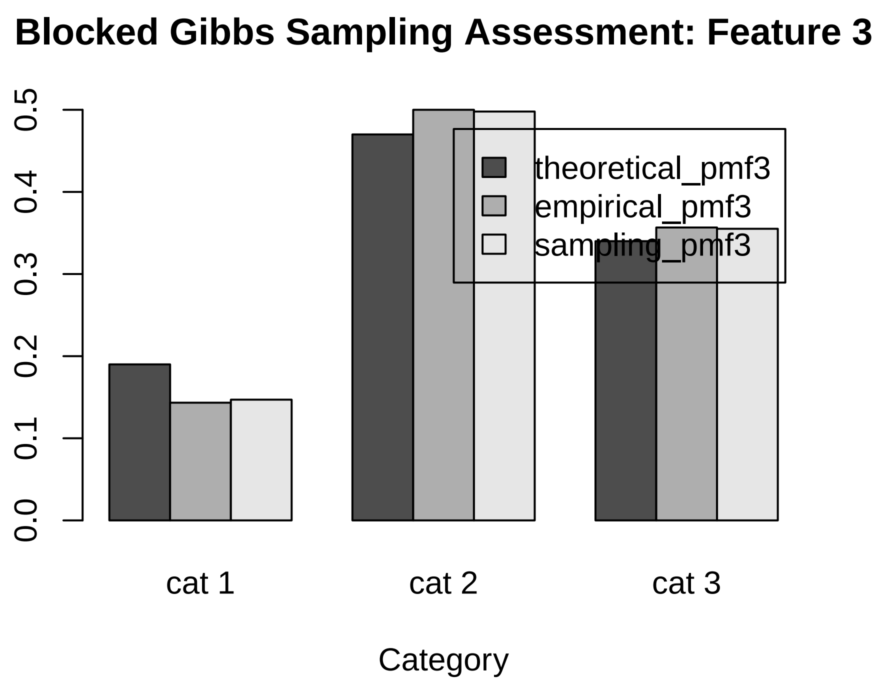
where n_k is the number of observations found in cluster k

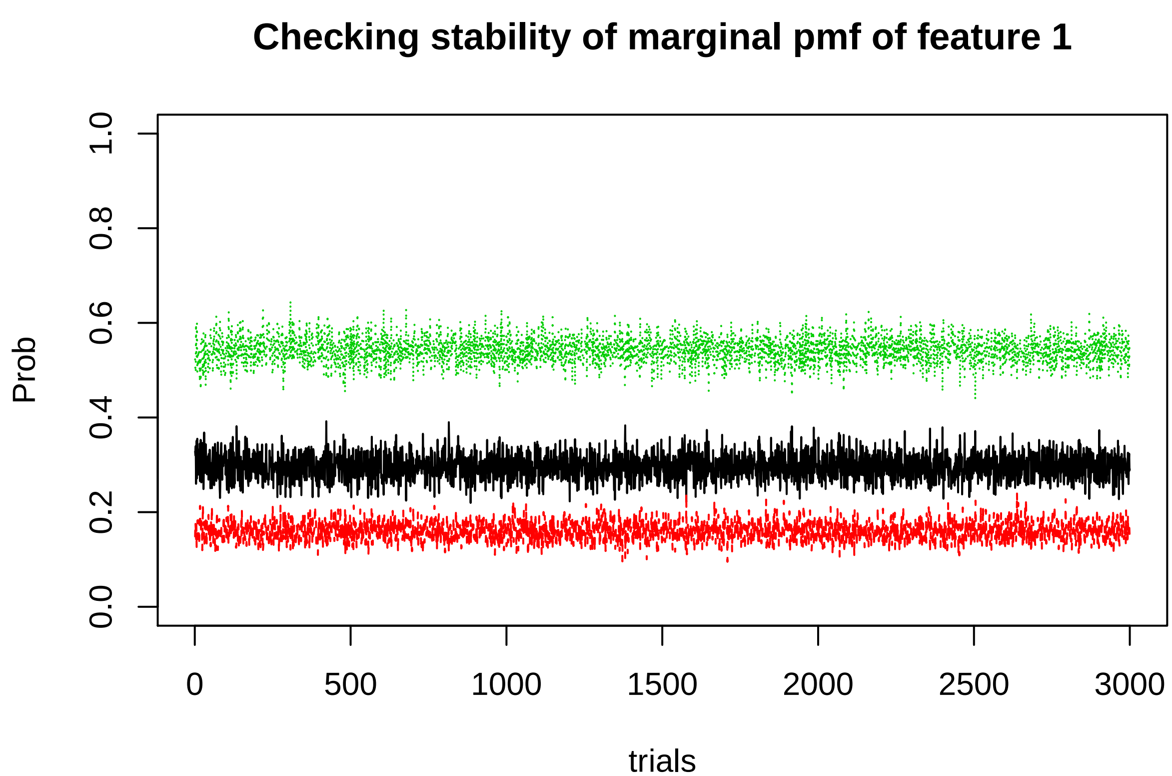
```r
# Blocked Gibbs Sampling
set.seed(1)
# Initialize parameters
pi = c(0.33, 0.33, 0.34)

theta_1 = c()
theta_2 = c()
theta_3 = c()
for (i in 1:cluster_num) {
  theta_1 = rbind(theta_1, rdirichlet(1, a_theta))
  theta_2 = rbind(theta_2, rdirichlet(1, a_theta))
  theta_3 = rbind(theta_3, rdirichlet(1, a_theta))
}

# Initialize the sampling matrix
sample_pi = c()
sample_pmf1 = c()
sample_pmf2 = c()
sample_pmf3 = c()
for (round in 1:3000) {

  # Step 1: Sampling cluster indicator
  z = c()
  for (i in 1:dim(x1)[2]) {
    # Calculate the full conditional probability of belonging to cluster k
    fullcon_zi = pi
    # First feature
    fullcon_zi = fullcon_zi*rowProds(t(t(theta_1)^x1[,i]))
    # Second feature
    fullcon_zi = fullcon_zi*rowProds(t(t(theta_2)^x2[,i]))
    # Third feature
```

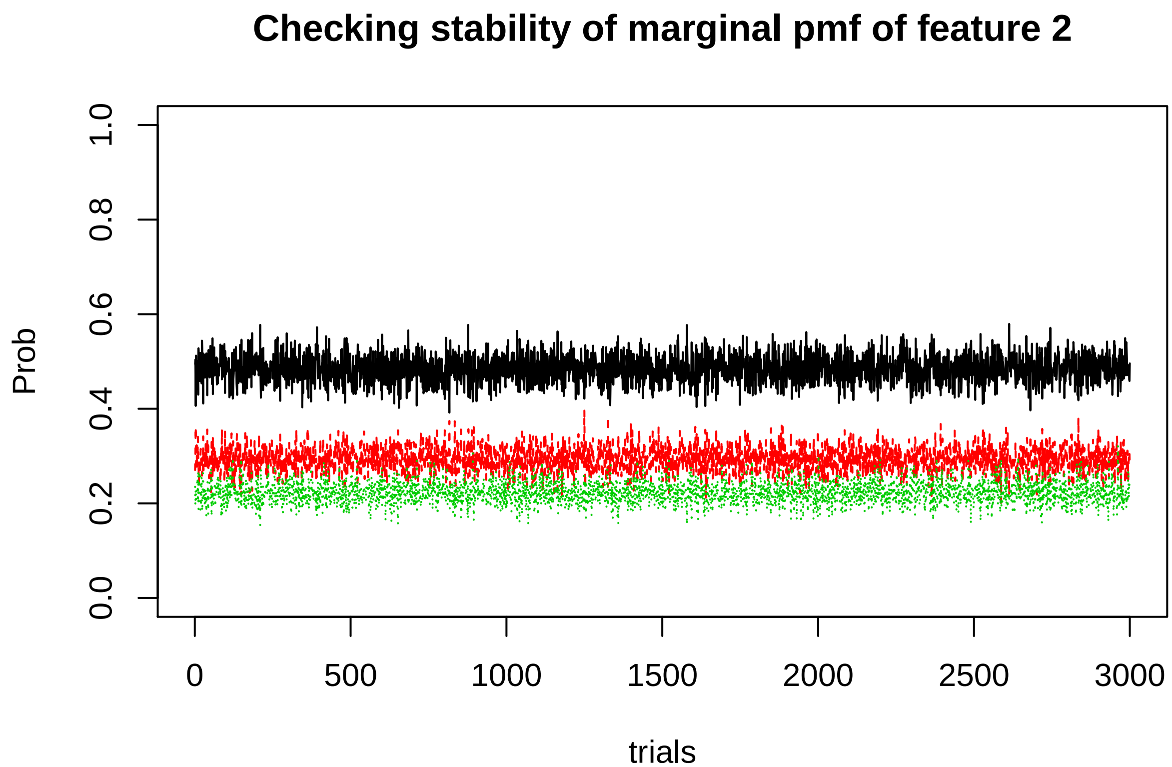
```
    fullcon_zi = fullcon_zi*rowProds(t(t(theta_3)^x3[,i]))
    # Scale conditional pmf
    fullcon_zi = fullcon_zi/sum(fullcon_zi)

    z = cbind(z, rmultinom(1,1,fullcon_zi))
  }

  # Step 2: Update theta
  for (k in 1:length(pi)) {
    if (is.null(dim(x1[, z[k,]==1]))) {
      # only one member of no member
      if (length(x1[, z[k,]==1] == 0)) {
        # No member
        nk1 = rep(0, category_num)
        nk2 = rep(0, category_num)
        nk3 = rep(0, category_num)
      }else{
        # One member
        nk1 = x1[, z[k,]==1]
        nk2 = x2[, z[k,]==1]
        nk3 = x3[, z[k,]==1]
      }
    }else{
      # More than one member
      nk1 = apply(x1[, z[k,]==1], MARGIN = 1, FUN = sum)
      nk2 = apply(x2[, z[k,]==1], MARGIN = 1, FUN = sum)
      nk3 = apply(x3[, z[k,]==1], MARGIN = 1, FUN = sum)
    }
    # Sample theta using full conditional distribution
    theta_1[k,] = rdirichlet(1, a_theta + nk1)
    theta_2[k,] = rdirichlet(1, a_theta + nk2)
    theta_3[k,] = rdirichlet(1, a_theta + nk3)
  }

  # Step 3: Update pi
  n = apply(z, MARGIN = 1, sum)
  pi = rdirichlet(1, a_pi + n)
  sample_pi = rbind(sample_pi, pi)

  # Record pmf
  sample_pmf1 = rbind(sample_pmf1, apply(as.vector(pi)*theta_1, MARGIN = 2, sum))
  sample_pmf2 = rbind(sample_pmf2, apply(as.vector(pi)*theta_2, MARGIN = 2, sum))
  sample_pmf3 = rbind(sample_pmf3, apply(as.vector(pi)*theta_3, MARGIN = 2, sum))
}
```

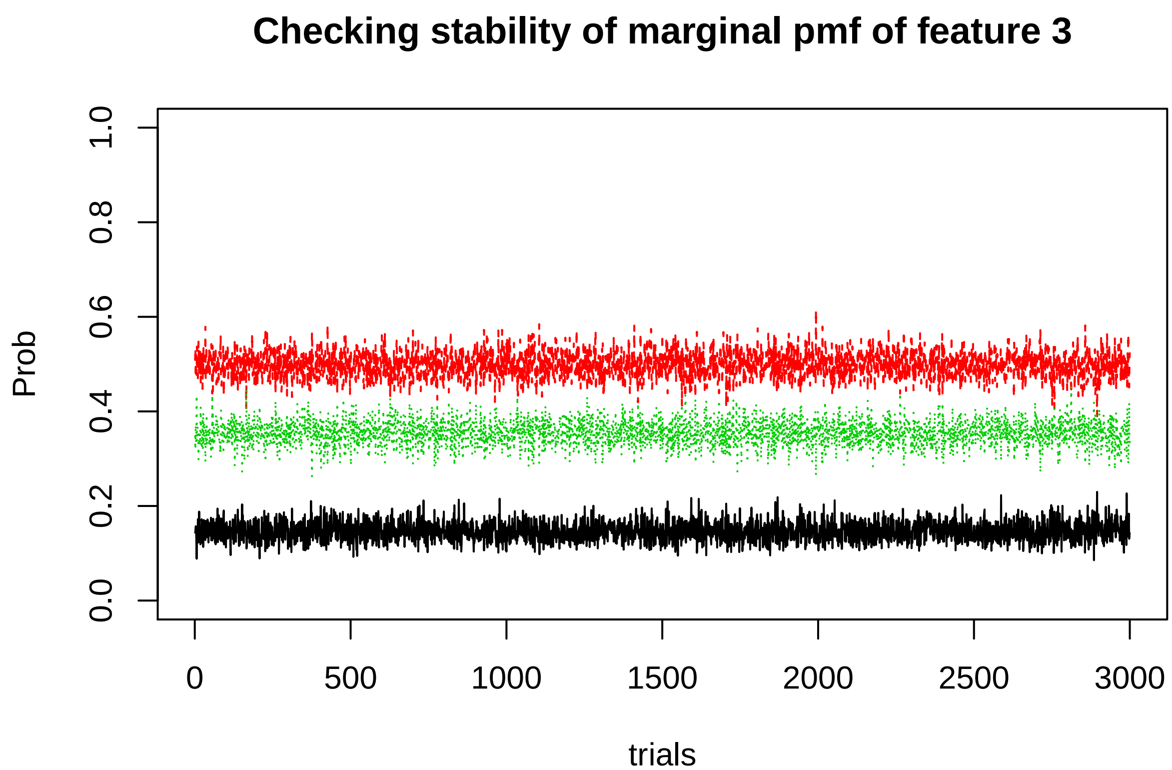

Blocked Gibbs Sampling Assessment: Feature 1
0.0
0.1
0.2
0.3
0.4
0.5
theoretical_pmf1
empirical_pmf1
sampling_pmf1
cat 1
cat 2
cat 3
Category

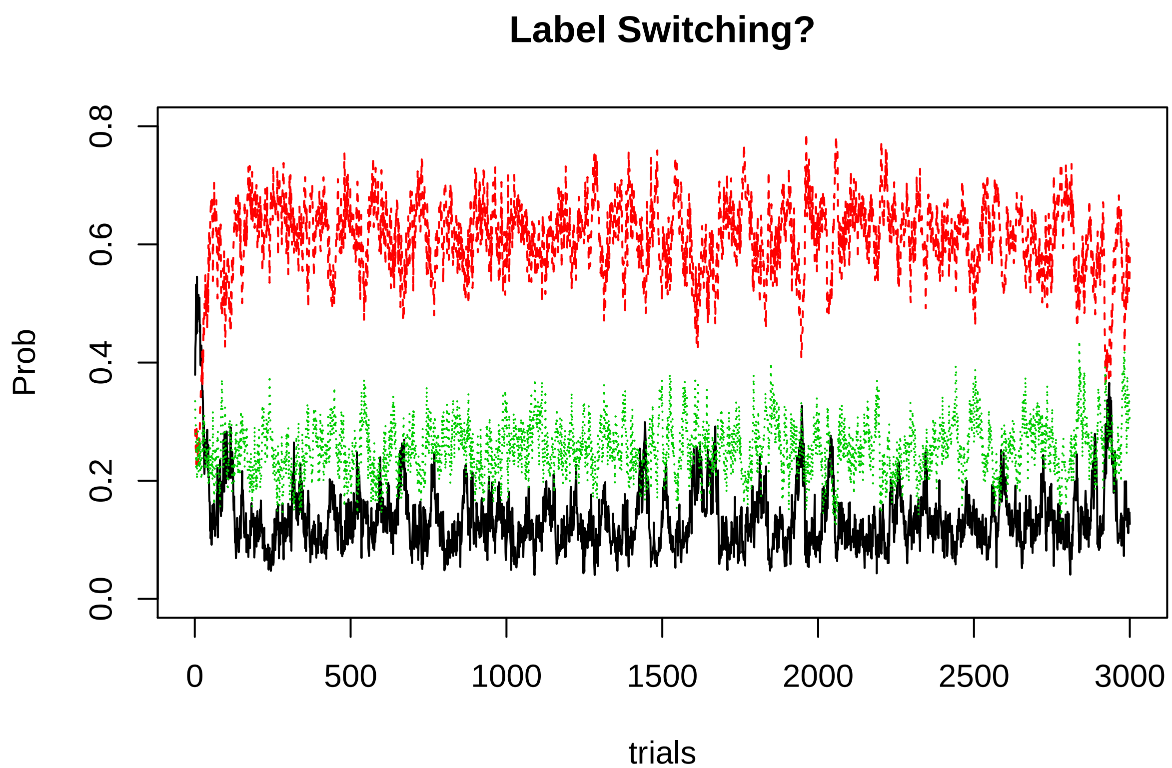

Blocked Gibbs Sampling Assessment: Feature 2
0.0
0.1
0.2
0.3
0.4
theoretical_pmf2
empirical_pmf2
sampling_pmf2
cat 1
cat 2
cat 3
Category


Blocked Gibbs Sampling Assessment: Feature 3
0.0
0.1
0.2
0.3
0.4
0.5
theoretical_pmf3
empirical_pmf3
sampling_pmf3
cat 1
cat 2
cat 3
Category


Checking stability of marginal pmf of feature 1
Prob
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000
trials


Checking stability of marginal pmf of feature 2
Prob
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000
trials


Checking stability of marginal pmf of feature 3
Prob
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000
trials


Label Switching?
Prob
0.0
0.2
0.4
0.6
0.8
0
500
1000
1500
2000
2500
3000
trials