DPMPM for missing data imputation: MNAR

Setting:

- K = 3 real 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
- There will be data missing not at random in the last features. The probability of missingness is govern by $logit(w_1^T x_{i1} + w_2^T x_{i2} + w_3^T x_{i3})$ where $w_1 = [-0.75, -1, 0.2]$, $w_2 = [0.6, -2, 0.5]$, and $w3 = [0.2, -0.8, -0.4]$. This results in about 31% of missing data in the last feature.
- For modelling process, we will not rely on Dirichlet distribution for mixing proportion but we will use Dirichlet process and specify the number of possible clusters to be high ($H^* = 10$ in this case). We will use stick breaking process to update all parameters and keep track of the number of clusters being assigned in the sampling process.

Summary on modelling process

$X_{ip} \mid z_i = k, \theta \sim Mult(\theta_{kp})$ for p = 1, 2, 3

$z \mid \pi \sim Mult(\pi_1, \pi_2, ...)$

$\pi_h = V_h \Pi_{g<h}(1 - V_g)$

$V_h \sim Beta(1, \alpha)$

$\alpha \sim Gamma(a_\alpha, b_\alpha)$

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

```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)
```

```r
# Create simulated data
n = 300
class_i = rmultinom(n, size = 1, prob = pi_true)
x1 = c()
x2 = c()
x3_original = 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_original = cbind(x3_original, rmultinom(1, size = 1, prob = theta_p3_true[class_i[, i]==1,]))
}
```

Generate missingness in data

```r
# Define parameter of logistic function
w1 =  c(-0.75, -1, 0.2)
w2 = c(0.6, -2, 0.5)
w3 = c(0.2, -0.8, -0.4)

# Calculate probability of missingness of features 3
prob = apply(w1*x1, MARGIN = 2, FUN = sum) +
  apply(w2*x2, MARGIN = 2, FUN = sum) +
  apply(w3*x3_original, MARGIN = 2, FUN = sum)

prob = 1/(exp(-prob)+1)

# Indicator for X3miss
indicator = rbernoulli(n = 300, p = prob)
x3 = x3_original
x3[, indicator] = c(NA, NA, NA)
```

```r
# Prior Parameter
cluster_num = 10 # H* for stick breaking process
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

# prior for pi
alpha = 1
V = rbeta(cluster_num, 1, alpha)
V[cluster_num] = 1 #truncation
pi = rep(0.1, cluster_num)
for (h in 1:cluster_num) {
  pi[h] = V[h]* base::prod(1 - V[0:(h-1)])
}

# Prior for alpha
a_alp = 0.25
b_alp = 0.25

# Impute x3 using marginal distribution
marginal_3 = apply(x3[,!indicator], MARGIN = 1, mean)
x3[,indicator] = rmultinom(sum(indicator), 1, marginal_3)
```

Blocked Gibbs Sampling here consists of four major steps

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

For each i:

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

where x_{ipj} denotes the indicator of outcome j of features p of sample i

2. Sample V_h from Beta distribution

$V_h \mid - \sim Beta(1 + n_h, \alpha + \Sigma_{k>h} n_k)$

where n_k denotes the number of samples assigned to cluster k and we assign $V_{H^*} = 1$ to truncate the inifinte number of sticks to 10.

Then we update π using stick breaking process

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

4. Sample α from the updated Gamma distribution

$\alpha \mid - \sim Gamma(a_\alpha + H^* - 1, b_\alpha - log(\pi_{H^*}))$

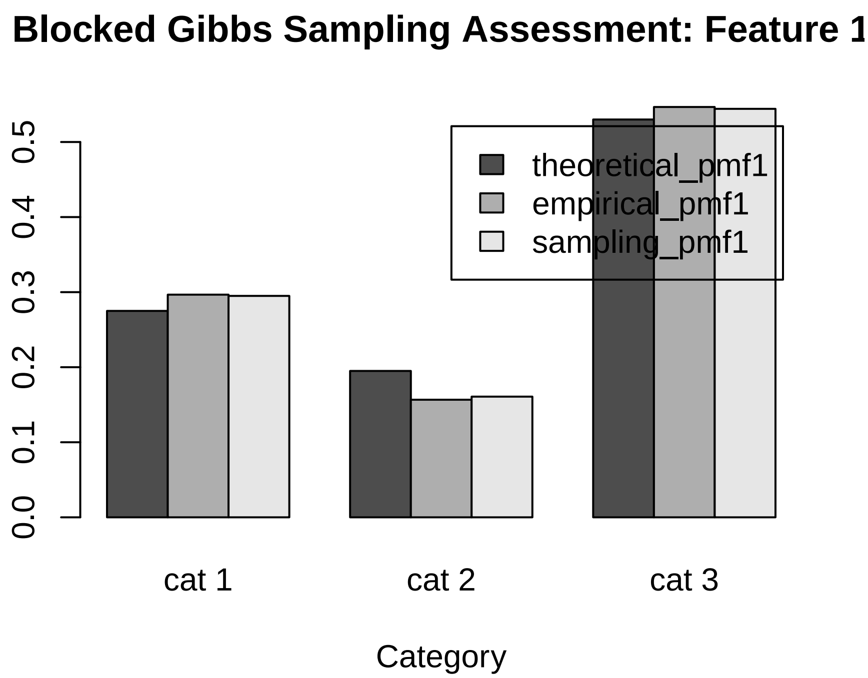
where π_{H^*} is the probability of being assign to the last cluster.

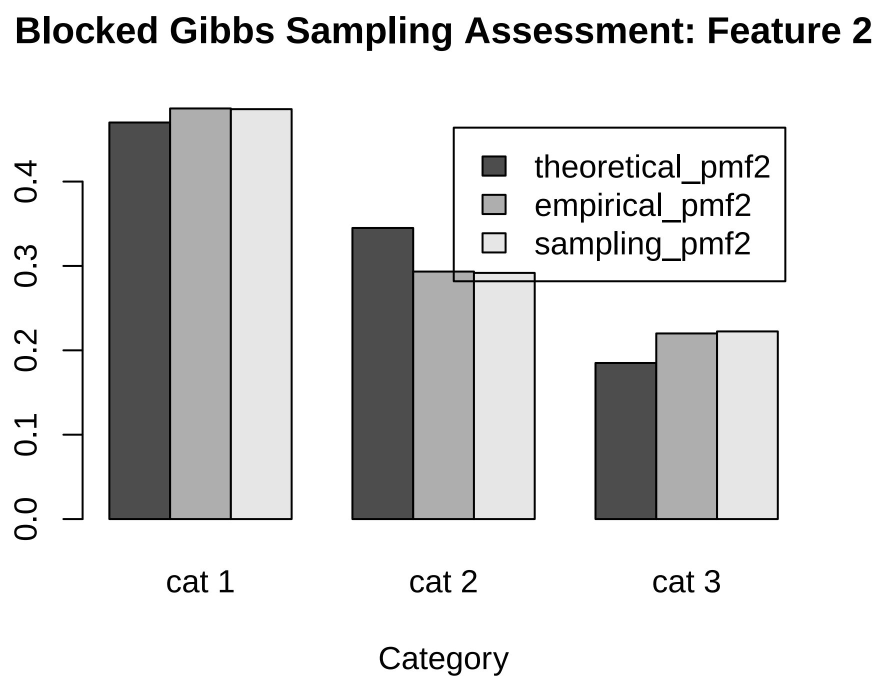
5. Sample $X_{i3} \mid z_i, - \sim Multinomial(\theta_{z_i 3})$

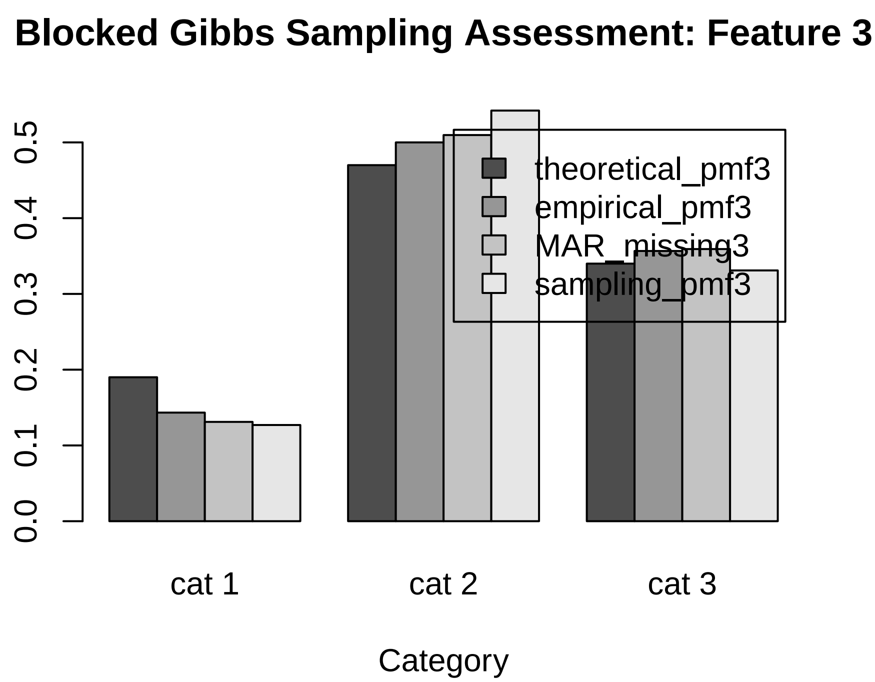
```
# Blocked Gibbs Sampling
set.seed(1)

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()
sample_cluster_used = c()
for (round in 1:15000) {
  # 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
```

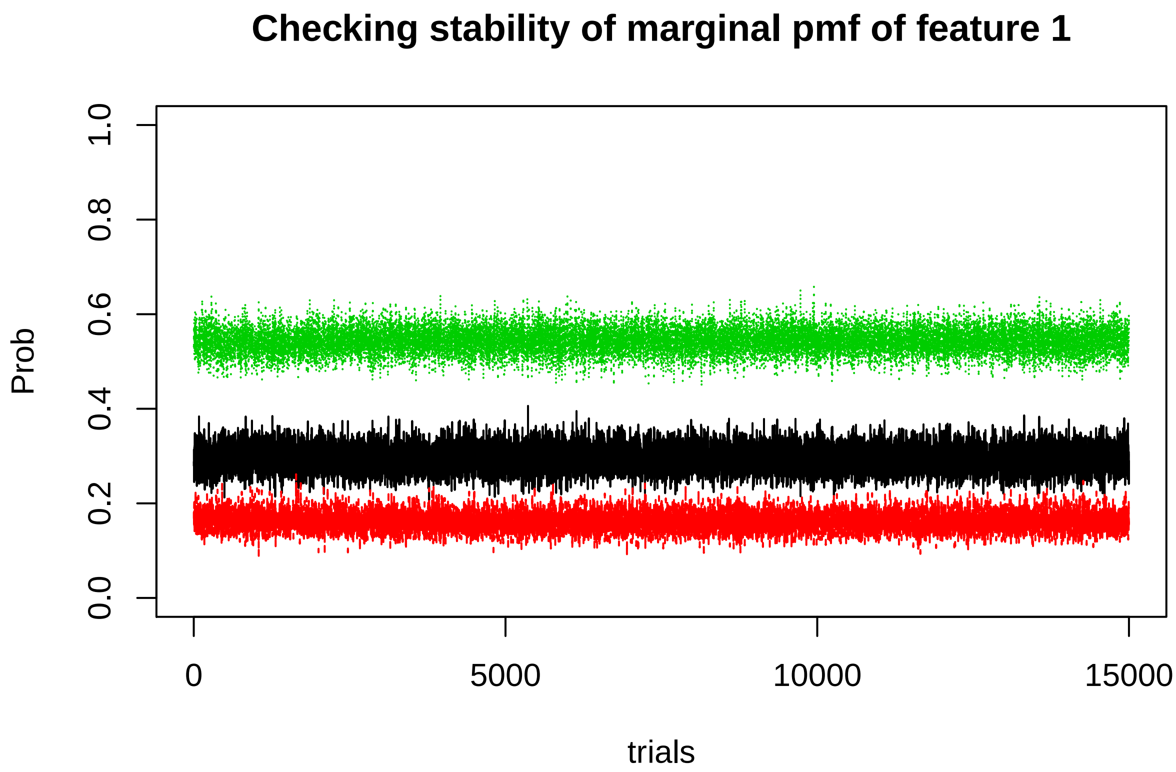
```r
  fullcon_zi = fullcon_zi*rowProds(t(t(theta_2)^x2[,i]))
  # Third feature
  fullcon_zi = fullcon_zi*rowProds(t(t(theta_3)^x3[,i]))
  # Scale conditional pmf
  fullcon_zi = fullcon_zi/sum(fullcon_zi)

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

# Step 2.1: Sampling Vh
n = apply(z, MARGIN = 1, sum) #number of samples in each cluster

for (k in 1:(cluster_num-1)) {
  V[k] = rbeta(1, 1 + n[k], alpha + sum(n[(k+1):cluster_num]))
}

# Step 2.2: Update Pi
for (h in 1:cluster_num) {
pi[h] = V[h]* base::prod(1 - V[0:(h-1)])
}
sample_pi = rbind(sample_pi, pi)
sample_cluster_used = c(sample_cluster_used, sum(pi>0))

# Step 3: Sampling 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 4: Sampling alpha
alpha = rgamma(1, shape = a_alp + cluster_num - 1, b_alp - log(pi[cluster_num]))
```

```
  # Step 5: Sample x3 for missing entries
  for (ind in 1:dim(x3)[2]) {
    if (indicator[ind]) {
      # This entry of x3 is missing
      x3[, ind] = rmultinom(1,1,prob = theta_3[z[,3]==1,])
    }
  }

  # 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))
}
```

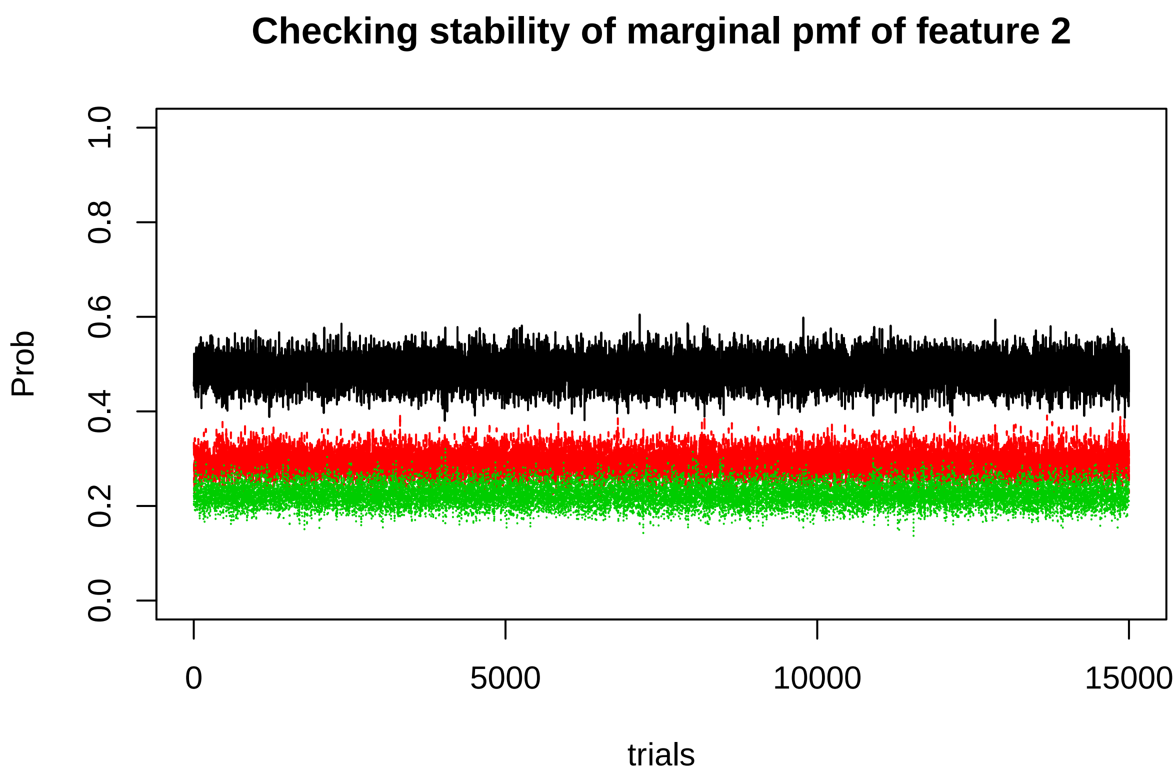



Blocked Gibbs Sampling Assessment: Feature 2
theoretical_pmf2
empirical_pmf2
sampling_pmf2
0.0
0.1
0.2
0.3
0.4
cat 1
cat 2
cat 3
Category

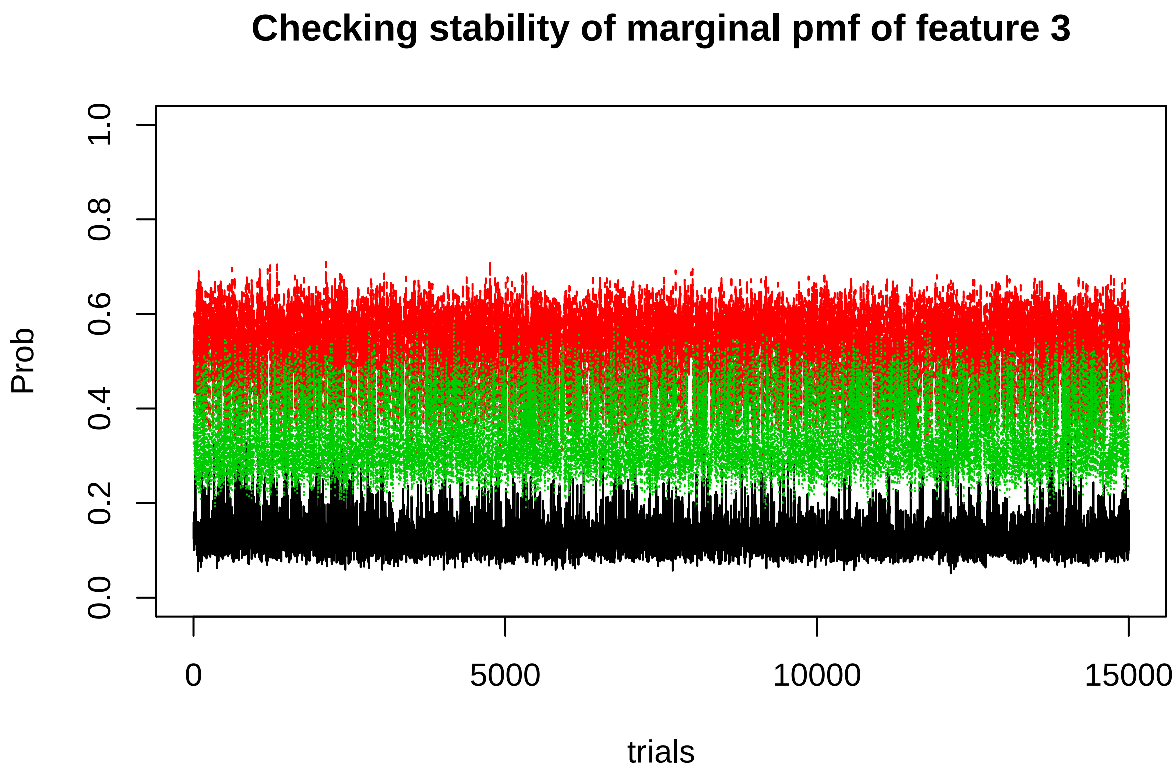

Blocked Gibbs Sampling Assessment: Feature 3
theoretical_pmf3
empirical_pmf3
MAR_missing3
sampling_pmf3
0.0
0.1
0.2
0.3
0.4
0.5
cat 1
cat 2
cat 3
Category

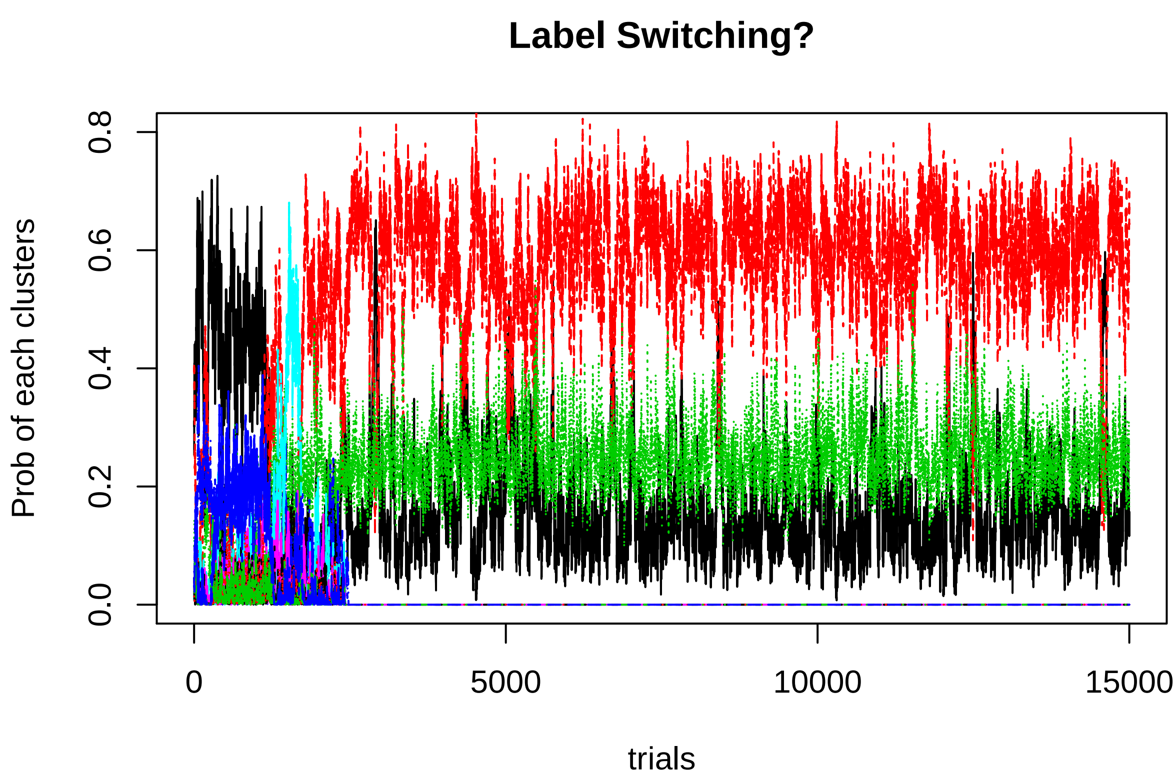


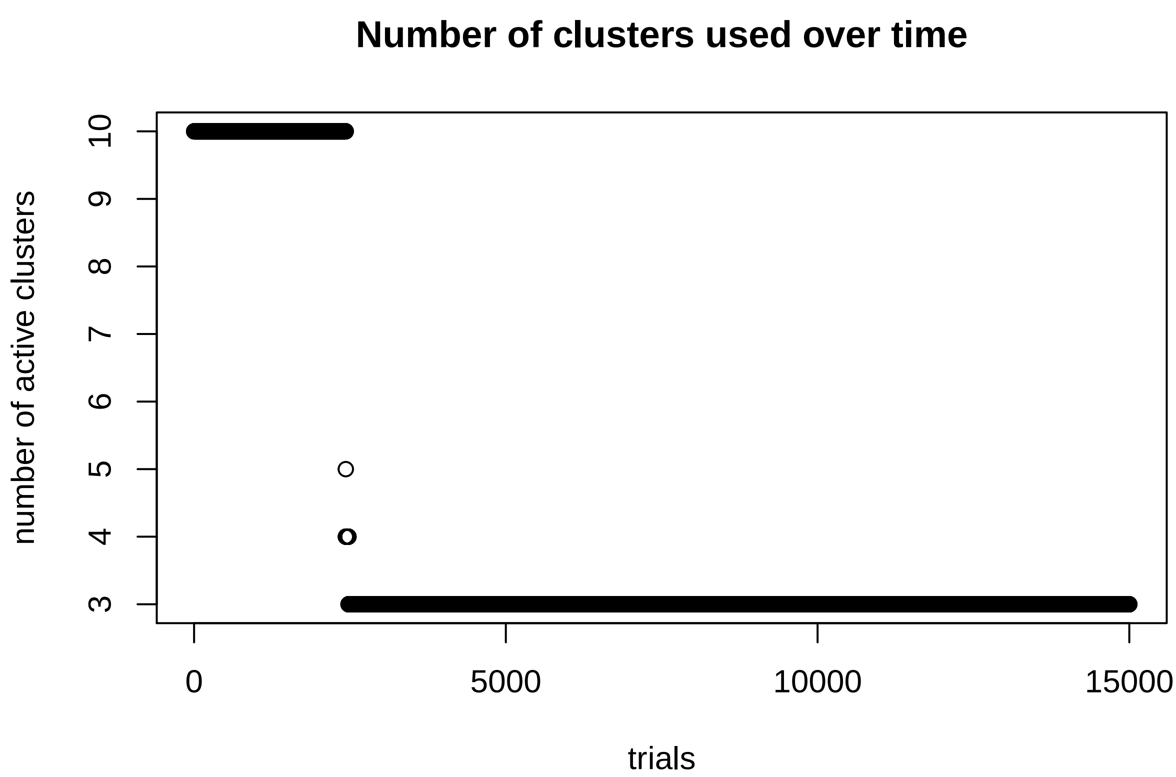
Checking stability of marginal pmf of feature 1
Prob
trials
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000



Checking stability of marginal pmf of feature 2
Prob
trials
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000


Checking stability of marginal pmf of feature 3
Prob
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000
trials


Label Switching?
Prob of each clusters
0.0
0.2
0.4
0.6
0.8
0
5000
10000
15000
trials


Number of clusters used over time
number of active clusters
10
9
8
7
6
5
4
3
0
5000
10000
15000
trials