Matt Dickenson `mcd31`
STA561/CS571 — Fall 2013 **Homework 9** Due: 25 November, 2013

Homework Notes: I did not work with anyone else on this homework or refer to resources other than the course notes, textbook, and course Piazza page.

Problem 1

A The generative model for a continuous η, a base distribution G_0, concentration parameter α, and the $\{B_1, ...B_K\}$ partitions ($K = \inf$), is:

$$
\begin{aligned}
(G(\eta \in B_1), ..., G(\eta \in B)K) &\sim \text{Dirich}(\alpha G_0(B_1), ...\alpha G_0(B_K)) \\
p(\eta_i \in B_j) &= \int p(\eta_i \in B_j|G)p(G|G_0)dG \\
&= \frac{\alpha G_0(B_j)}{\sum_K \alpha G_0(B_k)} \\
&\propto \alpha G_0(B_j)
\end{aligned}
$$

The posterior is

$$G|\eta_{1:n}, \alpha, G_0 \sim DP(\alpha, G_0 + \sum_{i=1}^{n} \delta_{\eta_i}(\eta))$$

A simple choice for the base distribution is G_0 is the Gamma distribution, due to the conjugacy of the Gamma distribution with the Gaussian distribution.

B For the cluster assignment step in the Gibbs sampler, we can exploit exchangeability.

```
cluster = function(x, centroids, alpha){
  # do the restaurant process
  table_counts = restaurant(x, alpha)
  table_props = table_counts/sum(table_counts)
  num_tables = length(table_counts)

  # then exploit permutation
  permuted_x = sample(x)
  n = length(x)
  table_assignments = rep(NA, n)

  # pretend each x is last to arrive
  for(i in 1:length(permuted_x)){
    table_i = sample(c(1:num_tables), 1, prob=table_props)
    table_assignments[i] = table_i
  }
}
```

```r
restaurant = function(x, alpha){
  table_counts = c(1) # number of 'customers' at each 'table'
                      # first customer at first table
  for(m in 2:n){
    tmp = c(table_counts, alpha)
    table_props = tmp/sum(tmp)

    # assign each 'customer' to a 'table' according to crp
    table_m = sample(c(1:length(tmp)), 1, prob=table_props)
    if(table_m==length(tmp)){ table_counts[table_m] = 1}
    else{ table_counts[table_m] = table_counts[table_m] + 1}
  }
  return(table_counts) # sufficient statistic
}
```

C This algorithm does not discard empty clusters. Because the number of clusters is potentially infinite (in theory), it is possible for an observation to be assigned to a previously empty cluster at any iteration of the Gibbs sampler.

D Rather than discarding clusters with fewer than γ points assigned to them, we allow the number of potential clusters to be infinite. Thus, this model better addresses the issue of not knowing the number of clusters *a priori*. If we examine the number of clusters at each iteration of the Gibbs sampler (or across multiple runs), we can even get a posterior distribution over the number of clusters.

Problem 2