Matt Dickenson `mcd31`
STA561/CS571 — Fall 2013 **Homework 3** Due: 7 October, 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

$$\begin{aligned}\hat{\mu}_k &= \sum_{i=1}^{n} p(z_i = k|\hat{\pi}, \hat{\Sigma}_k) * x_i \\ \hat{\pi}_k &= \sum_{i=1}^{n} p(x_i|\hat{\mu}_k \hat{\Sigma}_k)\end{aligned}$$

B These estimates differ slightly from those in the Murphy textbook. In that text, $\hat{\pi}_k = 1/K$ for all k, whereas here we use information in the data to estimate $\hat{\pi}$. Similarly, in the Murphy book $\mu_k = \frac{1}{N_k}\sum_{i:z_i=k} x_i$ relies on the indicator funciton, while in the version above $\hat{\mu}$ is a weighted average of the x_i's based on the probability that observation i is in category k. Thus, the version above relies more fully on information in the data rather than the `argmin` of the deviance function.

Problem 2

A Let $n_X = \sum_{i=1}^{n} X_i$. The data likelihood can be written

$$\begin{aligned}P(\mathcal{D}|\theta) &= \prod_{i=1}^{n} P(X_i|\theta) \\ &= \prod_{i=1}^{n} \mu_{Z_t} \times X_t + (1-\mu_{Z_t}) \times (1 - X_t) \\ &= \mu_{Z_t}^{n_X} \times (1-\mu_{Z_t})^{n-n_X}.\end{aligned}$$

B The complete log likelihood can be written

$$\begin{aligned}\ell_c(\theta) &= \log P(X, Z|\theta) \\ &= \log(\prod_{i=1}^{n} P(X_i, Z_i|\theta)) \\ &= \sum_{i=1}^{n} (\mu_{Z_t} \times X_t + (1-\mu_{Z_t}) \times (1-X_t)) \times (\pi_{Z_{t-1}} \times Z_t + (1-\pi_{Z_t}) \times (1-Z_{t-1})) \\ &= (\mu_{Z_t} \times \pi_{Z_{t-1}} \times n_X) + ((1-\mu_{Z_t}) \times (1-\pi_{Z_t}) \times (n-n_X))\end{aligned}$$

C Omitted.

D Omitted.

E Omitted.

Problem 3

A I would expect to have a single cluster ($K = 1$) because in a single dimension the mean value should minimize the sum of squared distances from all X_i.

B The following code is my implementation of the adaptive K-means algorithm in R:

Listing 1: R Code for 3B

```
mykmeans = function(x, maxiters=100){
  # step 1
  centroids = eta_0 = c(mean(x)) #eta_0
  k = 1
  iters = 0
  clusters = matrix(1, nrow=length(x), ncol=1)

  converged = FALSE

  while(!converged){
    # step 2
    last_eta = centroids[length(centroids)]
    new_eta = rnorm(1, eta_0, sd(x))
    centroids = c(centroids, new_eta)

    # step 3
    new_clusters = cluster(x=x, centroids=centroids)
    clusters = cbind(clusters, new_clusters)

    # step 4
    to_keep = which(c(1:length(centroids)) %in% new_clusters)

    keep = centroids[to_keep]

    lastk = k
    k = length(keep)

    centroids = keep

    # update retained centroids
```

```
    for(i in 1:length(centroids)){
      subset = x[which(new_clusters[, 1]==i)]
      centroids[i] = mean(subset, na.rm=TRUE)
    }

    iters = iters + 1
    if(k==lastk || iters>=maxiters){ converged=TRUE }
  }

  output = list(K=k, numiters=iters, means=centroids)
  return(output)
}

cluster = function(x, centroids){
  # assign each X_i to one of k+1 clusters
  n = length(x)
  k = length(centroids)
  labels = matrix(NA, nrow=n, ncol=1)

  for(i in 1:n){
    dists = matrix(NA, nrow=1, ncol=k)
    for(j in 1:k){
      dists[ , j] = euclid(x[i], centroids[j])
    }
    labels[i, ] = which(dists == min(dists, na.rm=TRUE))
  }
  return(labels)
}

euclid = function(a, b){
  return((a-b)^2)
}

answer3 = mykmeans(X[1:1000,1])
answer3
```

Using this code, I end up with $K = 100$, and the centroid means are not consistent each time the function is called.

C My implementation clearly overfits the data. One way to prevent overfitting is to adjust step 4 so that we only keep centroids η_k with at least c X_i's assigned to them. I reimplemented the algorithm with $c = 10$. This reduced K, but the result is again not consistent when the function is called multiple times.

Problem 4

Omitted.

Problem 5

A The width of the tree is $\log_2 K$.

B Omitted.

Problem 6

Omitted.