STA561/CS571 — Fall 2013 **Homework 4** Due: October 7, 2013

*Instructions:* Please put all answers in a single PDF with your name and NetID and upload to SAKAI before class on the due date (there is a LaTeX template on the course web site for you to use). Definitely consider working in a group; please include the names of the people in your group and write up your solutions separately. If you look at any references (even wikipedia), cite them. If you happen to track the number of hours you spent on the homework, it would be great if you could put that at the top of your homework to give us an indication of how difficult it was.

## Problem 1

*EM and K-means*

(a) We wish to compute the M-step in an E-M algorithm for a Gaussian mixture model. Starting with the expected complete data log likelihood (auxiliary) function:

$$
\begin{aligned}
Q(\theta, \theta^{(t-1)}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} p(z_i = k|x_i, \theta^{(t-1)})\ log[\pi_k] + \sum_{i=1}^{n}\sum_{k=1}^{K} p(z_i = k|x_i, \theta^{(t-1)})\ log[p(x_i|\theta_k)] \\
& \text{where } \ \theta = \{\mu_k, \pi_k, \Sigma_k\} \\
& \text{(based on Equation 11.26 on page 351 in Murphy's Book)}
\end{aligned}
$$

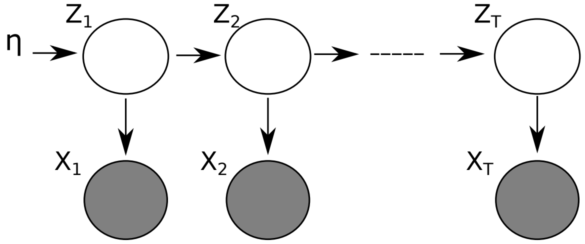
derive the maximum likelihood estimations of $\mu_k$ and $\pi_k$.

As defined in equations 11.1 and 11.2 on pages $338 - 9$ in Murphy's book: each base distribution in the mixture model is a multivariate Gaussian with mean vector $\mu_k$ and covariance matrix $\Sigma_k$. And $\pi$ are the mixing weights sastifying $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

(b) Compare these maximum likelihood updates for $\pi$ and $\mu$ to that of K-means (Section 11.4.2.5 on page 352 in Murphy's book). How are the updates similar and how do they differ? *Use* $3 - 4$ *sentences to explain your answer.*

## Problem 2

*Hidden Markov Model*



In this problem we will be using a first-order Hidden Markov Model (HMM). Where our latent variables are denoted as $Z \in \{Z_1, ..., Z_T\}$. Assume we have observed data

$\mathbb{D} = \{X_1, \ldots, X_n\}$ where each $X_i \in \{0,1\}^T$ is a vector of length $T$ of binary variables. Assume the emission and transition distributions are:

$$X_t | Z_t \sim Ber(\mu_{Z_t})$$
$$Z_t | Z_{t-1} \sim Ber(\pi_{Z_{t-1}})$$

And the initial latent state distribution is defined by $\eta$:

$$\eta = \prod_{k=1}^{K} \eta_k^{Z_1^k}$$

Note: that $Z_t$ and $Z_{t-1}$ are subscript notations of the parameters $\mu$ and $\pi$ in the Bernoulli distributions above.

Our goals are to estimate parameters $\theta = \{\eta, \mu, \pi\}$ using expectation maximization.

(a) Write out the likelihood of the data, $P(\mathbb{D}|\theta) = \prod_{i=1}^{n} P(X_i \mid \theta)$, that exploits the factorization available in the HMM construction. Make sure your final likelihood term involves the variables $Z$.

(b) Write out the complete data log likelihood ($\ell_c(\theta) = \log\, P(X, Z|\theta)$).

(c) Write out the expected complete log likelihood($Q(\theta, \theta^{\,t-1}) = \mathbb{E}[\ell_c(\theta)|\mathbb{D}, \theta^{\,t-1}]$).

(d) Explicitly write out the maximum likelihood updates for $\pi$ (the transition parameters) and $\mu$ (the emission parameters), assuming the E-step is as we discussed in class.

(e) Clearly write out the steps for updating the parameters, $\theta$, in pseudocode (but you do not need to implement it).

### Problem 3

*Conceptual clustering algorithm*

We have seen in class that we can use the K-means algorithms to identify $K$ clusters in a data set. Now consider a case where we are given a data set $X \in \mathbb{R}^{1000\times 1}$ (download from Sakai) and do not know the appropriate pre-defined cluster number, $K$. We propose to cluster the data, and determine the number of clusters $K$ from the data, using the following method:

(*Note that $\eta_k$ is the centroid of cluster k, and $K$ is the number of clusters, each with their own centroid: $\eta = \{\eta_1, ..., \eta_K\}$*)

**Adaptive K-means algorithm:**

1. Initialize by labeling all of the data in one cluster (all $X_i$'s are assigned to one cluster and call $\eta_0$ is their centroid). Store $\eta_0$.

2. Within the range of the data (i.e., based on centroid $\eta_0$), randomly generate a new centroid, which we will call $\eta_{K+1}$.

3. Assign each $X_i$ to one of $K+1$ clusters based on which $\eta_k$ minimizes the Euclidean distance:

$$z_i = \text{argmin}_k ||x_i - \eta_k||^2$$

4. Keep all centroids, $\eta_k$, that have at least one $X_i$ assigned to it. Update the value of $K$ based on the total number of labelled centroids. Update the retained centroids based on assigned data points, as in the standard K-means algorithm.

5. Repeat Steps $2 - 4$ until you have reached a reasonable stopping place.

**Question:**

1. Conceptually, how many clusters do we expect to have? In other words, what value of $K$ should we expect? Answer this before implementing.

2. When you implement this method in R, how many clusters do we end up with (please show code)?

3. How do you suggest we modify our method to prevent overfitting? Try this approach in your code and report the estimated $K$. Did it work?