CSCI 1950-F Homework 8:
K-Means Clustering & Bernoulli Mixture Models

Brown University, Spring 2012

Homework due at 12:00pm on April 26, 2012

Question 1:

In this question, we use the K-means algorithm to cluster the handwritten digit data. For all sections, we use $1,000$ examples of each of the 10 digit classes, so that there are $N = 10,000$ data items in total. Letting μ_k denote the mean for cluster k, the K-means objective function can be written as

$$J(y, \mu) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} ||x_i - \mu_k||^2$$

where $y_{ik} = 1$ if example i is assigned to cluster k, and 0 otherwise. Note that since we are testing a clustering algorithm, the true MNIST digit class labels will be used only to evaluate hypothesized clusterings, *not* as part of the clustering algorithm.

See Homework 3 for instructions on reshaping and plotting digit vectors as images. Note that the `subplot` command can be used to plot several images in a single figure.

a) *Implement and submit a function which runs the K-means algorithm to convergence for any K, from an initialization specified via a set of starting cluster centers $\{\mu_k\}_{k=1}^{K}$. Your function should record and return the value of the K-means objective function $J(y, \mu)$ at each iteration. You must write your own implementation, not use or copy an existing Matlab function.*

b) *Run the K-means algorithm on the digit data with $K = 10$, the true number of clusters. Randomly initialize K-means by choosing $K = 10$ of the observations at random, and setting the initial cluster centers to be these observations. Plot the K-means objective as a function of iteration, and verify that it monotonically decreases. Furthermore, plot the resulting centroids learned by K-means for all 10 clusters.*

c) *Repeat part (b) for 10 different random initializations, running the K-means algorithm to convergence from each. Evaluate the consistency of each resulting clustering with the true digit labels by computing the Rand index (the second output argument of the function* `valid_RandIndex.m`*). Make a scatter plot of the Rand index values, versus the corresponding values of the K-means objective function $J(y, \mu)$. Does the K-means objective provide a good predictor of cluster quality?*

d) *When clustering algorithms do not perform perfectly, there are two major sources of error: the objective function or model may not match the data well, or the algorithm used to optimize that objective may be stuck in local optima. We can sometimes separate these issues by "cheating". Consider a K-means initialization in which the cluster centers μ_k are set to the means of the 10 digit classes, as determined via the* true *class labels. Run K-means to convergence from this initialization, and compute the resulting Rand index and objective function values. How do these compare to those from part (c)? What does this suggest about how we should try to build a better clustering method for this data?*

e) *We conclude by considering how K-means performs on this data as the number of clusters, K, is varied. For each $K \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, run K-means to convergence from a single random initialization determined as in part (b). Plot both the K-means objective function and the Rand index (computed using the true cluster labels) versus K. Which value of K gives the lowest objective function? Which gives the largest Rand index?*

f) *For the model that achieved the highest Rand index, plot the inferred centroids as images. How do these clusters compare to the centroids from part (b)?*

Question 2:

In this question, we will implement a Bernoulli mixture model to perform unsupervised clustering on a dataset of English words annotated by a large set of binary features. There are a total of 541 unique words labeled with 824 binary features describing properties of each word, such as "is a musical instrument" or "has buttons".

We will use the EM algorithm implementation from the `ptmk` toolbox, as called by the `mixDiscreteFit` function. See the skeleton code for more details.

a) *Run the EM algorithm on the word feature dataset, using $K = 8$ clusters and the options specified in the skeleton code. Notice that the α parameter is set to 1. How would modifying this value change the type of inference EM is doing? Plot the log-likelihood for this run versus iteration. Does this log-likelihood monotonically increase? Why or why not?*

b) *Repeat this experiment 10 times, running EM from 10 random initializations. Select the models with the highest and lowest final log-likelihoods. For these two models, use* `mixDiscreteInferLatent` *to calculate the posterior distributions for assigning each word to the various mixture components. List the 5 words most likely to be associated each of the $K = 8$ mixture components, for each model. Do they correspond to meaningful groupings or categorizations? How do the best and worst runs differ in terms of their word clusterings?*

c) *Use the feature labels included in the dataset to list the top five features associated with each mixture component. You can do this by ranking the probabilities found in the resulting model structure under* `model.cpd.T(k,2,:)` *where k refers to the component index. Compare the top five feature labels for the best and worst runs. Is there a significant difference in the coherence or interpretability of these feature labels?*