IEOR E4525
Martin Haugh

Due: 5pm Thursday 24^{th} March 2016

Assignment 5

Answer Question 3 and either Question 1 or Question 2.

1. *(Clustering Gene Expression Data)*

   In this problem you will be working with a cancer tumor gene expression data set that is associated with a paper published in 2001 in PNAS (Proceedings of the National Academy of Sciences). The paper itself is available at http://goo.gl/nZQI4. (You might recognize one of the co-authors, Eric Lander. He was one of the leaders in the Human Genome Project which is credited with being the first to sequence the human genome.)

   The goal is to cluster the gene expressions using Euclidean K-means and other Kernel K-means, and compare the clusters with the actual labeling of the samples.

   The gene expression data is in the table `xcancer.dat` – each row is a gene and each column is a sample from a particular type of cancer. The true cluster label for each column, i.e. each cancer sample, is in the table `ycancer.dat`. There are $K = 14$ clusters in this data set. (This is an example where you happen to know the clusters in advance.)

   In the paper listed above, the authors standardize the gene expression data as follows:

   (i) All gene expression values that were below 20 were set to 20, and all the values that were above $16,000$ units were set to $16,000$. This is because very low and very high values of gene expressions are very noisy.

   (ii) All the genes that did not show more than 5-fold variation across cancer samples, i.e. $\frac{\text{max gene expression}}{\text{min gene expression}} \leq 5$, were removed from consideration.

   (iii) All the genes that did not show more than 500 unit variation across cancer samples, i.e. max gene expression $-$ min gene expression ≤ 500, were removed from consideration.

   (iv) All the remaining rows, i.e. gene expressions, were standardized to have mean 0 and standard deviation 1.

   Standardize the data in the manner described above and do a K-means clustering with $K = 14$.

   Now we want to compare the clustering obtained with the true class labels. Since it is possible that the clustering obtained by K-means completely mixes up the clusters, this comparison is not as trivial as it sounds. We will use the following methodology. Define a matrix $Y \in \{0,1\}^{N \times N}$ according to

$$Y_{ij} = \begin{cases} 1, & y(i) = y(j), \\ 0, & y(i) \neq y(j). \end{cases}$$

where $y(i)$ denotes the class label of the i^{th} sample for $i = 1, \ldots, N$. The matrix Y encodes the pair-wise labeling of samples.

Let $z \in \mathbb{R}^N$ denote the cluster labeling obtained from the K-means clustering. Let $Z \in \{0,1\}^{N \times N}$ denote the matrix

$$Z_{ij} = \begin{cases} 1, & z(i) = z(j), \\ 0, & z(i) \neq z(j). \end{cases}$$

We can then "score" the clustering obtained by K-means according to the metric

$$\rho(Z) = \sum_{i \neq j} \Big(Y_{ij}(1 - Z_{ij}) + (1 - Y_{ij}) Z_{ij} \Big).$$

This metric encodes the fact that when $Y_{ij} = 1$ (resp. $Y_{ij} = 0$) the clustering has an error when $Z_{ij} = 0$ (resp. $Z_{ij} = 1$). If the clustering perfectly recovers the class labels, $\rho(Z) = 0$; otherwise $\rho(Z) > 0$.

(a) Check if the true clustering is stable under Euclidean K-means.

(b) Compute the best (Euclidean) K-mean clustering starting from $m = 10$ different random starting points. In this problem we score the quality of a clustering using the ρ metric defined above.

(c) Use the `kkmeans` function in the `R` package `kernlab` to compute the clustering using the RBF Kernel. Again, repeat the clustering from $m = 5$ random starting points and pick the best.

(d) Comment on the quality of the clustering.

You can also use `MATLAB` to do this problem if you prefer. `MATLAB` has a function called `kmeans`. There is no function to do Kernel K-means directly in `MATLAB` but some code that will work is described here: http://goo.gl/WtmG0. (You can also find kernel K-means code on the `MATLAB CENTRAL File Exchange`.)

2. *(Clustering Movies)*

In this problem we will use K-medoid clustering together with MDS visualization to see whether movie ratings cluster similar movies together.

The details of the movies are in the file `u.item` and the ratings are in the file `u.data`. The details of the contents are in the file `u.info`. There is also a snippet of R code that takes the information from the two files and merges the information appropriately.

(a) Use the ranking data to create a dissimilarity matrix between movies. Decide how you will take care of the missing data? Justify your answer. You might want to take a look at the `daisy` function from the package `cluster`.

(b) Perform the clustering.

(c) Embed the movies into $\mathbb{R}^2$ (colored by cluser) and interpret the clusters if you see any.

(d) You might want to try distances other than the Euclidean distance to see if it improves the performance.

3. *(EM Algorithm in Conjoint Analysis)*

In this problem, we will be working on a data set from a conjoint study of cell phone plan choices. This conjoint study was designed to investigate the "values" consumer ascribe to the different features of a cell phone plan by asking them to make a series of choices in an experimental setting.

- A total of 72 participants were recruited for this study.
- Each of them was presented with a sequence of 18 individualized choice sets.
- Each choice set had 3 (conjoint) profiles of cell phone plans and a no-choice option.
- Each (conjoint) profile described a cell phone plan using six features or attributes:
  (1) service provider (categorical)
  (2) access fee
  (3) plan minutes
  (4) the per-minute rate
  (5) Internet access (categorical 0/1)
  (6) rollover of unused minutes (categorical 0/1)
- Each participant was asked to choose her most preferred (conjoint) profile, or the no-choice option if she would like to choose none of the three (conjoint) profiles, from each choice set.
- We randomly selected 15 out of the 18 choice sets for each participant for calibration, and used the remaining 3 choice sets for out-of-sample prediction.

Indices for the general conjoint problem

- I = number of participants.
- J = number of choice sets.
- Q = number of conjoint profiles in each choice set
- p = number of attributes
- $x_{ijq} \in \mathbb{R}^p$ = q-th profile in the j-th choice set of participant i
- q^* = index of the profile that participant i chooses from the j-th choice set
- x_{ijq^*} = attributes of the chosen profile

MATLAB data description The conjoint data, $X := \left\{\{x_{ijq}\}_{q=1}^{Q}, x_{ijq^*}\right\}_{i,j}$, is stored in the MAT-file `cellphone_data.mat`. There are four arrays in the file.

- The 4-D array `Design-Array-Calib` stores the calibration conjoint profiles. Specifically, `Design-Array-Calib(q,:,j,i)` is the q-th conjoint profile in the j-th choice set of participant i, i.e. the vector x_{ijq}. This an 11-dimensional vector that encodes information regarding the six attributes according to:
  (a) the intercept, every profile gets a "1" except the no choice option
  (b) the dummy for the brand Cingular
  (c) the dummy for the brand T-Mobile
  (d) the dummy for the internet option
  (e) the dummy for the rollover option
  (f) the access fee
  (g) the plan minutes
  (h) the per-minute rate
  (i) the log of the access fee
  (j) the log of the plan minutes
  (k) the log of the per-minute rate

  The final six components were standardized to have zero mean and unit variance.
- The 2-D array `Choice-Matrix-Calib` stores the indices of the chosen profiles of `Design-Array-Calib`. Specifically, `Choice-Matrix-Calib(j,i)` stores the index of the chosen profile in the j-th choice set of participant i.
- `Design-Array-Holdout` and `Choice-Matrix-Holdout` are similarly structured arrays that hold the information of the test profiles.

Finite mixture (FM) model for conjoint data In the FM model, the preference function of the i-th participant is given by the logit model. Let $\beta_i \in \mathbb{R}^p$ denote a weight vector and assume the utility she derives from the q-th profile of her j-th choice set is given by

$$\mathrm{U}_{ijq} = x_{ijq}^\top \beta_i + \epsilon_{ijq},$$

where $\{\epsilon_{ijq}\}$ are i.i.d. Gumbel random variables. It can be shown that the probability that x_{ijq^*} is chosen from $\left\{x_{ijq}\right\}_{q=1}^{Q}$ is

$$\mathbb{P}(q^* = q) = \frac{e^{x_{ijq^*}^\top \beta_i}}{\sum_{q=1}^{Q} e^{x_{ijq}^\top \beta_i}}.$$

The FM model assumes that each participant belongs to one of K segments. The utility of each participant in segment k is defined by the weight vector γ_k. Each participant is

assumed to be drawn independently from these latent, i.e. unobserved, segments with probability ω_k for segment k. That is, each β_i is drawn from the following distribution:

$$\mathbb{P}(\beta_i = \gamma_k) = \omega_k, \quad i = 1, \ldots, I, k = 1, \ldots, K.$$

Hence, the log-likelihood function of the FM model is

$$\mathcal{L}(\theta; X) = \sum_{i=1}^{I} \log \left(\sum_{k=1}^{K} \omega_k \prod_{j=1}^{J} \frac{e^{x_{ijq^*}^\top \gamma_k}}{\sum_{q=1}^{Q} e^{x_{ijq}^\top \gamma_k}} \right).$$

where the parameter vector $\theta = \left\{\omega_k, \gamma_k\right\}_{k=1}^{K}$. (Be sure that you do indeed understand why $\mathcal{L}(\theta; X)$ is as given.) The goal is to compute the maximum likelihood estimate θ^* by solving $\max_\theta \mathcal{L}(\theta; X)$.

We will use the expectation-maximization (EM) framework to numerically maximize the log-likelihood function. The EM framework introduces a set of unobserved data to characterize the latent segment memberships of the participants:

$$z_{ik} = \begin{cases} 1 & \text{iff participant } i \text{ belongs to segment } k, \\ 0 & \text{otherwise.} \end{cases}$$

Using $\{z_{ik}\}$, we can define the *complete-data* log-likelihood:

$$\mathcal{L}(\theta; X, z) = \sum_{i=1}^{I} \sum_{k=1}^{K} z_{ik} \log \left(\omega_k \prod_{j=1}^{J} \frac{e^{x_{ijq^*}^\top \gamma_k}}{\sum_{q=1}^{Q} e^{x_{ijq}^\top \gamma_k}} \right).$$

(a) *E-Step Details*

Let $q_{ik} = \mathbb{P}(z_i = k \mid X, \theta^{(t)})$, $i = 1, \ldots, I$, $k = 1, \ldots, k$, denote the probability of the latent variables conditional on the data and the current parameter estimate $\theta^{(t)}$. Show that

$$q_{ik} \propto \omega_k^{(t)} \prod_{j=1}^{J} \frac{e^{x_{ijq^*}^\top \gamma_k^{(t)}}}{\sum_{q=1}^{Q} e^{x_{ijq}^\top \gamma_k^{(t)}}}.$$

Note that the expected complete-data log likelihood is then given by

$$\begin{aligned} Q(\theta; \theta^{(t)}) := \mathbb{E}[\mathcal{L}(\theta; X, z) \mid X, q_{ik}] \;\; = \;\; & \sum_{k=1}^{K} \Big(\sum_{i=1}^{I} q_{ik} \Big) \log \omega_k \\ & + \sum_{k=1}^{K} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} q_{ik} \Big(x_{ijq^*}^\top \gamma_k - \ln \Big(\sum_{q=1}^{Q} e^{x_{ijq}^\top \gamma_k} \Big) \Big) \right) \end{aligned}$$

(b) *M-Step Details*

The M-step maximizes $Q(\theta; \theta^{(t)})$ over θ.

(i) Show that the optimization over the ω_k's yield

$$\omega_k \propto \sum_{i=1}^{I} q_{ik}$$

(ii) Note that the optimization problem for γ_k is given by

$$\max_{\gamma} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} q_{ik} \Big(x_{ijq^*}^{\top} \gamma - \ln \Big(\sum_{q=1}^{Q} e^{x_{ijq}^{\top} \gamma} \Big) \Big) \right).$$

This is a convex optimization problem that can be solved in MATLAB using CVX. You can download CVX from http://cvxr.com/cvx/. The CVX formulation is posted on *CourseWorks*.

(c) Write the complete EM code to estimate $\theta = (\omega, \gamma)$ assuming there are $K = 3$ segments. First start with the number of respondents $I = 10$, and then scale up to $I = 72$ if your machine can handle it.

(d) Report the average prediction error on the holdout test samples.

(e) How would you compute the "optimal" number of segments?