IEOR E4525 Machine Learning for OR and FE
Martin Haugh

Due: Thursday 24 March 2016

Solutions to Assignment 5

1. **(Clustering Gene Expression Data)**

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

   Solution: The true labeling is not stable under Euclidean k-means. We can show this by initializing k-means with 14 centroids defined by gene expression averages for each "true" cancer group. The k-means algorithm changes some of the class labels and converges to a different cluster assignment. This suggests that Euclidean k-means cannot fully capture the structure of gene expression data.

   (d) Comment on the quality of the clustering.

   Solution: With 10 trials the Euclidean k-means gave a minimal value for $\rho = 3070$. Kernel k-means produced $\rho = 2932$ after 5 trials. Again, it appears that the structure of this dataset is not adequately represented in a Euclidean metric space, and is somewhat better represented in an infinite-dimensional space corresponding to the RBF kernel. The `R` code `Genes_HW5_Q1.R` is provided in the solutions.

2. **(Clustering Movies)**

   (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`.

   Solution: We will use the idea from the `daisy` function to treat missing ratings. For every pair of movies $(\mathbf{m}_i, \mathbf{m}_j)$ we find a set of critics K_{ij} who rated both movies and compute dissimilarities using ratings of these common critics. For the Euclidean distance metric:

$$d(\mathbf{m}_i, \mathbf{m}_j) = \sqrt{\frac{1}{|K_{ij}|} \sum_{k \in K_{ij}} (m_{ik} - m_{jk})^2}. \tag{1}$$

   Note that we have controlled for the number of common critics $|K_{ij}|$ to prevent scaling issues.

   The drawback of this method is that it gives noisy similarity measurements for movies rated by few critics: if a cardinality of the common critics set is small, the dissimilarity between two movies depends on opinions of just a few people. An extreme case of this are movies rated by disjoint sets of critics for which dissimilarity cannot be computed.

   To address this issue we can explore the data on critics and possibly find some proxy values for their rankings. For simplicity we will just set missing values in the dissimilarity matrix to the matrix average - this will ensure that movies without known similarity numbers are on the "background" during the clustering stage, i.e. they are not closer or farther apart than the average. Other options include setting m_{ik} to the average rank of critic k, the average rank of the movie i or a weighted combination of the two.

Computing dissimilarity matrices by "brute force" with *daisy* or *dist* functions can take a lot of time and memory and may even crash the RStudio. To make this computation feasible, we will compute dissimilarities ourselves directly using the *critics* matrix.

(b) Perform the clustering.

Solution: The `R` code `Movies_HW5_Q2.R` is provided in the solutions.

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

Solution: Figure 1 shows an approximation of movie distances embedded in $\mathbb{R}^3$ using the *cmdscale* function, with movies colored by 5 clusters. The clustering was done with k-medoids on the Euclidean dissimilarity data. It is apparent from these plots that Euclidean dissimilarities do not represent the movie data well.

The problem is caused by our method for treating missing ratings. By filling in the missing data $|K_{ij}|$ becomes the same for all movies. For movies with few critics $m_{ij} = \bar{m}$ for most i, j, where $\bar{m}$ the average rating in the movie-critic matrix. Thus most terms in equation (1) are zero, but since $|K_{ij}|$ is the same for all movies we do not adjust for this! As a result, movies with few critics have smaller dissimilarities and they all appear together in the center of the plot.

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

Solution: To resolve this issue we need to properly normalize dissimilarities, accounting for data sparsity - not all movies have the same number of user ratings. Here we use the Gower distance (see references in the *daisy* function help) which accounts for this issue. The resulting plots on Figure 2 show that Gower distance better captures the structure of sparse ratings data and we can see some distinct movie clusters. We also plot 3-dimensional scatter plots for each cluster and note that they mostly occupy separate regions in $\mathbb{R}^3$

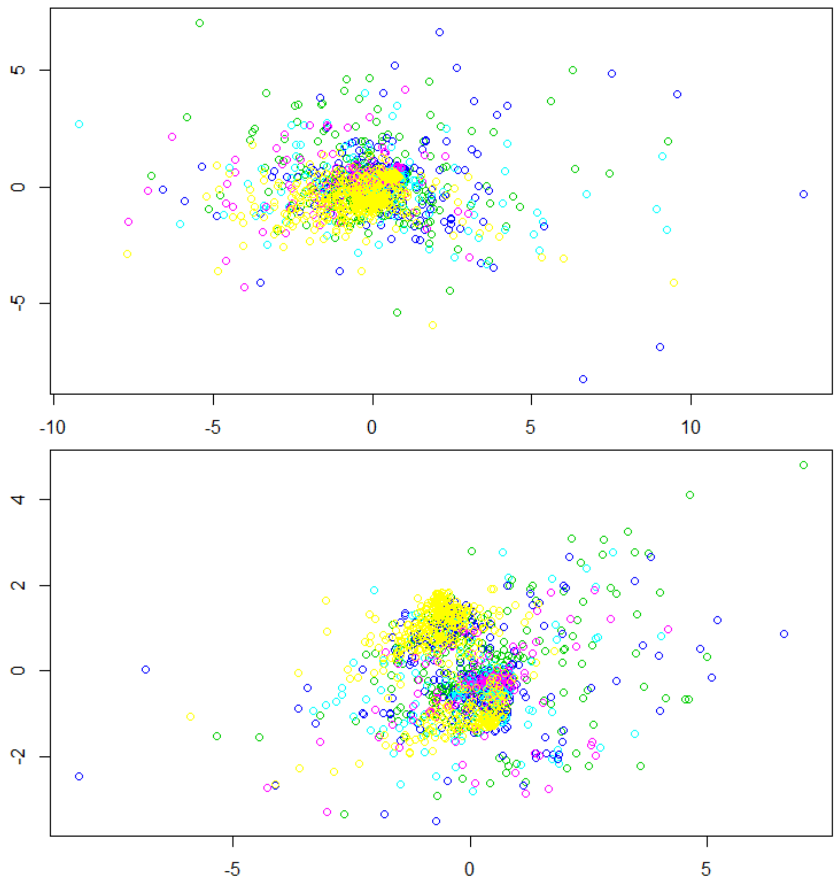


Figure 1: An approximation of Euclidean movie dissimilarities in $\mathbb{R}^3$. Top: first and second dimensions of the approximation, bottom: second and third dimensions

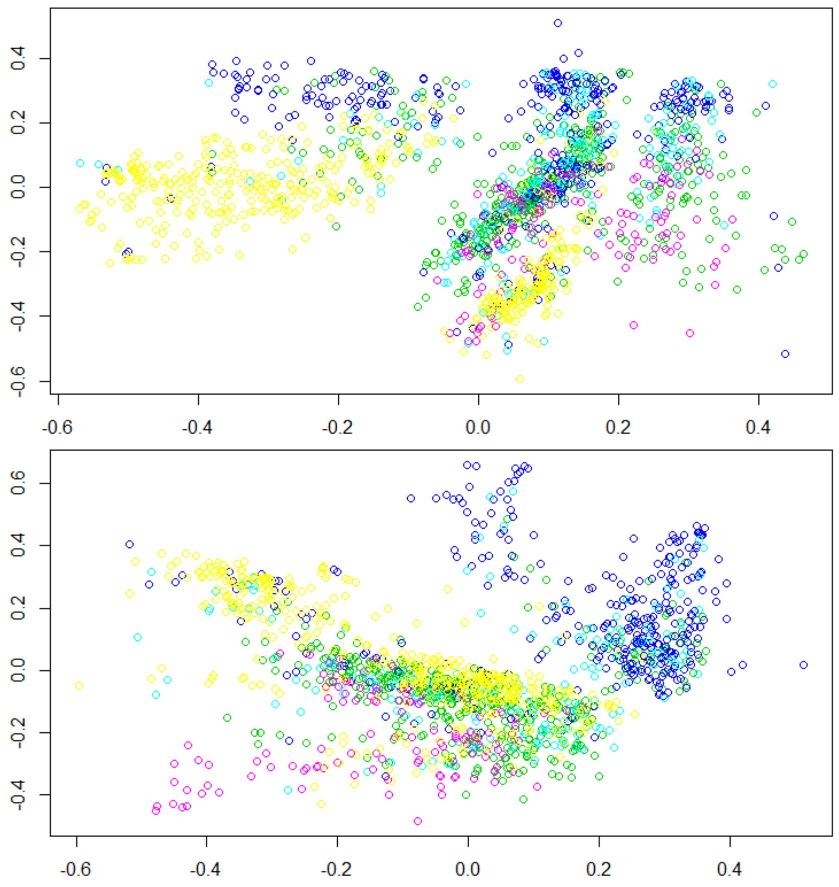
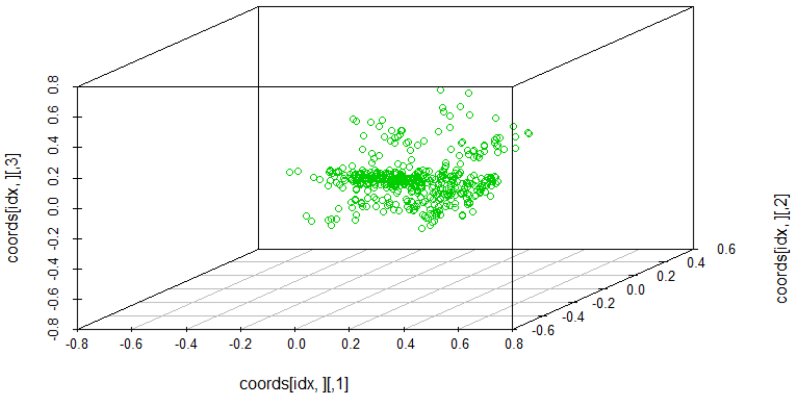
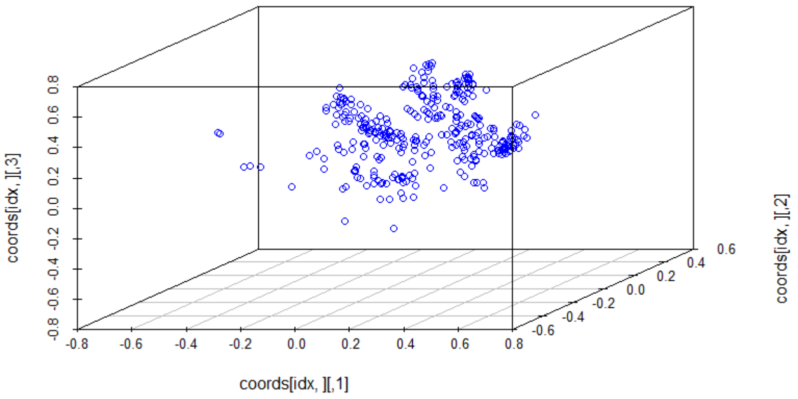
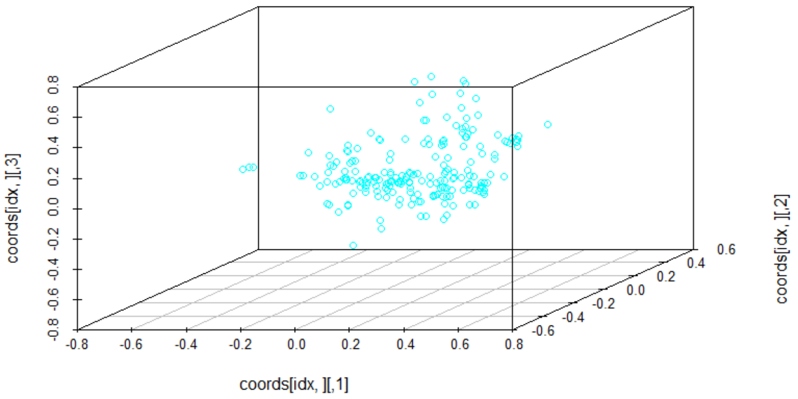


Figure 2: An approximation of Gower movie dissimilarities in $\mathbb{R}^3$. Top: first and second dimensions of the approximation, bottom: second and third dimensions


coords[idx,][,1]
coords[idx,][,2]
coords[idx,][,3]
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8


coords[idx,][,1]
coords[idx,][,2]
coords[idx,][,3]
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8


coords[idx,][,1]
coords[idx,][,2]
coords[idx,][,3]
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8

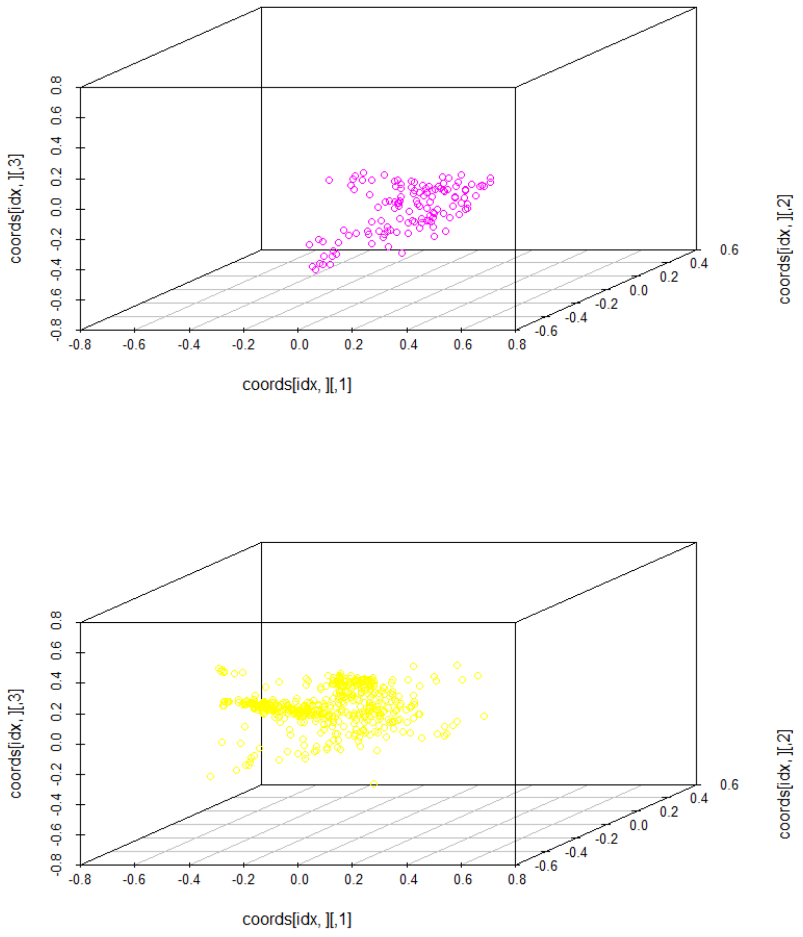


Figure 3: Three-dimensional scatter plots showing approximate Gower dissimilarities between movies, colored by cluster.

3. **(EM Algorithm in Conjoint Analysis)**

(a) E-Step Details

Solution: This is a simple application of the Bayes rule:

$$\mathbb{P}(z_i = k|x,\theta) = \frac{\mathbb{P}(x|z_i = k,\theta)\,\mathbb{P}(z_i = k|\theta)}{\mathbb{P}(x|\theta)}$$

We can further factorize $\mathbb{P}(x|z_i = k,\theta) = \mathbb{P}(x_i|z_i = k,\theta)\,\mathbb{P}\left(x_{(i)}|z_i = k,\theta\right)$, where x_i are data from a participant i and $x_{(i)}$ are data from all other participants. Since data from different participants is assumed to be independent, $\mathbb{P}\left(x_{(i)}|z_i = k,\theta\right) = \mathbb{P}\left(x_{(i)}|\theta\right)$ and after factorizing the denominator in a similar fashion, we get

$$\mathbb{P}(z_i = k|x,\theta) = \frac{\mathbb{P}(x_i|z_i = k,\theta)\,\mathbb{P}(z_i = k|\theta)}{\mathbb{P}(x_i|\theta)} \propto \mathbb{P}(x_i|z_i = k,\theta)\,\mathbb{P}(z_i = k|\theta)$$

After substituting $\mathbb{P}(z_i = k|\theta) = \omega_k$ and the likelihood expression for

$$\mathbb{P}(x_i|z_i = k,\theta) = \prod_{j=1}^{J} \frac{e^{x_{ijq*}{}^\top \gamma_k}}{\sum_{q=1}^{Q} e^{x_{ijq}{}^\top \gamma_k}}$$

we recover the answer.

(b) M-Step Details

Solution: The M-step maximizes $Q(\omega,\gamma) = \mathbb{E}_{\mathbb{Q}}\left[\mathcal{L}(z,\omega,\gamma)\right]$ whose expression is provided in the assignment. Specifically, we want to $\max\limits_{\omega,\gamma} Q(\omega,\gamma)$ subject to a constraint $\sum_k \omega_k = 1$. Note that $Q(\omega,\gamma)$ separates into two terms. The first term depends only on ω and the second - only on γ so we can maximize them separately. To find the optimal ω the problem is $\max\limits_{\omega,\gamma} \sum_{k=1}^{K}\left(\sum_{i=1}^{I} q_{ik}\right)\log\omega_k$ subject to a constraint $\sum_k \omega_k = 1$. Setting a Lagrangian for this problem and differentiating it with respect to ω_k leads to an equation

$$\frac{\sum_{i=1}^{I} q_{ik}}{\omega_k} - \lambda = 0, \quad \text{i.e.} \quad \omega_k = \frac{\sum_{i=1}^{I} q_{ik}}{\lambda}$$

Summing ω_k and noting that $\sum_{k=1}^{K}\sum_{i=1}^{I} q_{ik} = 1$, we see that $\lambda = 1$ and therefore $\omega_k = \sum_{i=1}^{I} q_{ik}$.

(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.

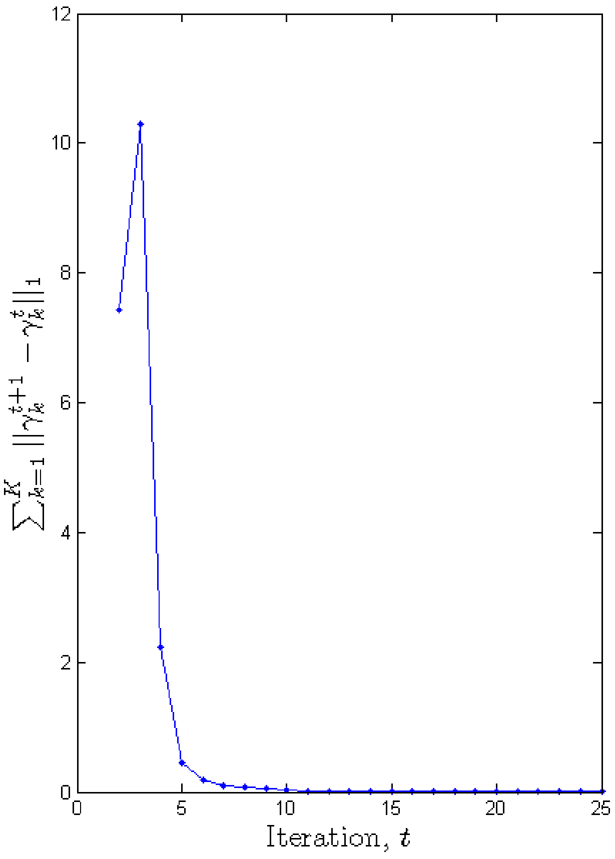
Solution: The code for this part is attached to the solutions. Note that it uses a fixed number *Nsteps* of EM iterations because each iteration is computationally expensive. If they were fast, it would be better to repeat EM iterations until a specified convergence criterion is met, for example the difference in parameter values ω, γ between successive iterations is smaller than a specific threshold.

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

Solution: The average prediction error (percentage of wrong choice predictions) is 40% with $I = 40, K = 3$ and $Nsteps = 25$. Figure 3d shows that the EM algorithm converged after about 10 iterations.

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

Solution: A good value for K can be selected by doing a cross-validation.



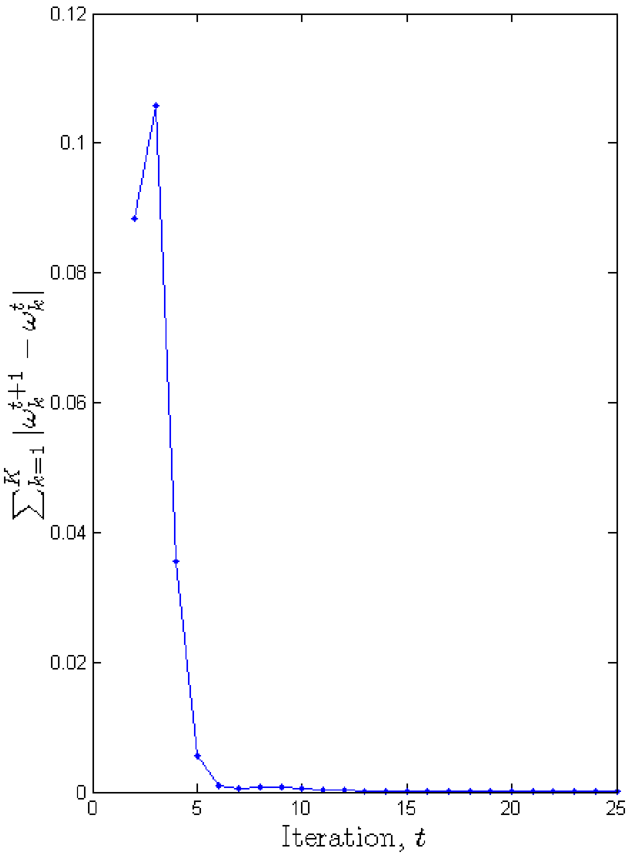


Figure 4: Differences in parameter estimates between EM iterations