STAT 6910-003 – SLDM II – Homework #6

Due: 5:00 PM 12/7/18

1. **PHATE (35 pts).** Download code for the PHATE visualization tool at `github.com/KrishnaswamyLab/PHATE`. Download the `EBdata2.mat` file from Canvas. This data file contains a sparse matrix labeled "data", a vector labeled "cells", and a vector labeled "libsize". The data matrix contains cleaned single-cell RNA-sequencing data from embryoid bodies with 16,825 cells and 17,580 genes. The cells vector indicates which time point the cell was extracted from. The libsize vector gives the sum of each row in the data matrix which will be used to normalize the data. You will apply PHATE to visualize this data.

   (a) (2 pts) Load the data. Indicate what packages or libraries you used. In particular, the data matrix is saved as a sparse Matlab matrix. You should be able to find R packages that can load sparse matlab matrices into R.
   Note: if you are using Python, you may find it easier to download the data from the Python tutorial `github.com/KrishnaswamyLab/PHATE/blob/master/Python/tutorial/EmbryoidBody.ipynb`. You can also directly modify the Python code in the tutorial for this problem if you wish.

   (b) (3 pts) Normalize the data using the libsize (stands for library size) vector. To do this, divide each entry in the ith row of the data matrix by the ith entry in the libsize vector.
   Note: if you are using the Python tutorial, the normalization process is built-in to the tutorial so you do not need to add anything to the code.

   (c) (10 pts) Run PHATE on the data using default parameters. Plot the PHATE coordinates colored by time point (the "cells" vector) and include the plot.
   Note: For single-cell data (like this data), we tend to get better results when we take the square root of each entry in the data matrix first. If you get a big "swoosh", then this is probably what's wrong.
   Note: On a high-powered laptop running Matlab, this takes about 2.5 minutes. Python is typically slower and R is typically slower than that. If it seems to be taking a long time to process then feel free to subsample the cells.

   (d) (10 pts) Run PHATE on the data using a different value of t. Plot the PHATE coordinates colored by time point and include the plot. Based on the results, do you think your chosen value of t is better than the parameter chosen using the "knee point" of the VNE plot (the value chosen using default parameters)? Will the VNE be higher or lower for your chosen value of t than that selected in part (b)?
   Note: You should be able to save on computation by inputting the previous results. See the tutorials.

   (e) (10 pts) Run PHATE on the data using default parameters to obtain 3d coordinates. Plot the 3d coordinates. You may need to rotate the plot to get a good view.

2. **Spectral clustering and GMM (55 pts).** Download all of the data for the MNIST dataset from `yann.lecun.com/exdb/mnist/`. This should give you 60,000 images in the training data and 10,000 images in the test data. You will apply a couple of clustering algorithms to this dataset. You may use any existing packages or libraries for this problem. For all parts, you may assume the same number of clusters as classes (10 in this case).

   (a) (7 pts) Apply k-means clustering to just the features of the training data (do not include labels). Compute the adjusted Rand index (ARI) between your cluster outputs and the true labels of the data points and report the value.
   Note: You may need to do subsampling to make this computationally feasible. If you do, repeat the clustering for multiple (say 10-20) subsamples and report the average ARI.

(b) (5 pts) Apply k-means clustering to just the features of the test data (do not include labels). Compute the ARI between your cluster outputs and the true labels of the data points and report the value.

(c) (10 pts) Apply spectral clustering to just the features of the training data (do not include labels) using a radial or Gaussian kernel. Compute the ARI between your cluster outputs and the true labels of the data points. Use the ARI to tune the kernel bandwidth parameter. Report the ARI using your selected bandwidth.
Note: You will likely need to do subsampling to make this computationally feasible. If you do, repeat the final clustering for multiple subsamples and report the average ARI.

(d) (7 pts) Apply spectral clustering to the features of the test data using the same kernel and bandwidth you selected in part (b). Report the ARI.

(e) (10 pts) Learn a Gaussian mixture model from the training data using the EM algorithm to cluster the data. Calculate and report the ARI.
Note: You may need to do subsampling to make this computationally feasible. If you do, repeat the clustering for multiple (say 10-20) subsamples and report the average ARI.

(f) (7 pts) Learn a Gaussian mixture model from the test data to cluster the data. Calculate and report the ARI.

(g) (5 pts) Based on the reported ARI numbers, which clustering algorithm seems to work the best on this data?

(h) (4 pts) Generally when we're clustering data, we don't have access to the true labels which makes it difficult to tune parameters like the kernel bandwidth. What is another way you could tune the bandwidth without using cluster or class labels?

3. How long did this assignment take you? (5 pts)

4. Type up homework solutions (5 pts)