Homework V

STAT/CS 5810/6655 - Spring semester 2024

Please upload your solutions in a single pdf file in Canvas. Any requested plots should be sufficiently labeled for full points. Include any code requested.

Unless otherwise stated, programming assignments should use built-in functions in your chosen programming language (Python, R, or Matlab). However, exercises are designed to emphasize the nuances of machine learning and deep learning algorithms - if a function exists that trivially solves an entire problem, please consult with the TA before using it.

1. **PHATE paper (10 pts).** Read the PHATE paper, which you can find on Canvas, up to at least the "Methods" section. Write a short (1-2 paragraph) summary of the paper. Were there any parts that were difficult to understand? Was there anything that was particularly interesting to you?

2. **Diffusion Maps (6655- 15 pts). Turn in your code for this problem. If you are using Python, create a jupyter notebook with all the code already run and the respective images displayed. If you are using R use markdown, and if you are using Matlab use a "Live Script".**

   (a) Implement diffusion maps using the steps described in the slides. Notice that you can reuse part of your code from the previous homework in kernel PCA if applicable.

   (b) Generate 2000 observations sampled from a Gaussian Mixture model with $K = 5$ in 2D (review the Gaussian mixture model slides for the meaning of K). You are free to use the parameters you decide, just make sure you do not have too much separation or overlap between each of the Gaussians.

   (c) Run your diffusion maps code on the generated data for 5 different values of the t parameter, plot the P^t matrix, and the line plot of its eigenvalues. Interpret the results.

3. **t-SNE gradient (6655 - 10 pts).** In class we have covered a dimensionality reduction method called t-SNE. Its objective function consists of minimizing the KL divergence between the distributions P and Q, where p_{ij} is computed using the original variables $\boldsymbol{x}$ and Q using the low-dimensional representations $\boldsymbol{y}$ (*decision variables*). **Review slide 13 in *25_ Visualization_ pre* for the equations of q_{ij} and p_{ij}.**

(a) If we rewrite $(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}$ as w_{ij}, show that the objective function can be rewritten as $-\sum_{ij} p_{ij} \log(w_{ij}) + \log\left(\sum_{l \neq k} w_{kl}\right)$. Interpret the role of these two terms, i.e., what should happen to the distances between the $\boldsymbol{y}$ variables if we introduce a penalization parameter λ as: $-\sum_{ij} p_{ij} \log(w_{ij}) + \lambda \log\left(\sum_{l \neq k} w_{kl}\right)$.

(b) To find a solution for t-SNE we resort to gradient descent. Find the gradient of $KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log(p_{ij}/q_{ij})$ with respect to $\boldsymbol{y}_i$.

4. **PHATE and Clustering (5810 - 35, 6655 - 25 pts).**
Turn in your code for this problem. If you are using Python, create a Jupyter notebook with all the code already run and the respective images displayed. If you are using R use markdown, and if you are using Matlab use a "Live Script". 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 PHATE to visualize the data and a couple of clustering algorithms to this dataset. You may use any existing packages or libraries for this problem as long as you cite them. For all parts, you may assume the same number of clusters as classes (10 in this case).
You can find code for PHATE at `github.com/KrishnaswamyLab/PHATE`.

(a) Run PHATE on just the features of the training data (do not include labels) using the default parameters to obtain a 2D representation of the data. Report the value of t selected using the von Neumann entropy (VNE). Rerun PHATE using two different values of t, one value larger than the value chosen using the VNE and one value smaller. Plot the PHATE visualization for all three values of t (you should end up with 3 different plots) with the data points colored by the labels. Comment on the plots. Which of the three values seems to give better separation between the classes? Does the relative position of the different classes make sense?

(b) 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. You should be able to find code online that computes the ARI automatically. Choose one of the PHATE plots from part (a) (a specific value of t) and plot the PHATE visualization colored by the cluster labels. Based on the visualization and the ARI value, does k-means match the true labels well?
Note: You may need to do subsampling to make this computationally feasible. If you do, repeat the clustering for multiple (say 10-20) random subsamples and report the average ARI. For the PHATE plot, choose one of the subsamples and show the results on that.

(c) 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. Choose one of the PHATE plots from part (a) and plot the PHATE visualization colored by the cluster labels. Based on the visualization and the ARI value, does spectral clustering do better or worse than k-means?

Note: You may need to do subsampling to make this computationally feasible. If you do, repeat the final clustering for multiple (say 10-20) random subsamples and report the average ARI. For the PHATE plot, choose one of the subsamples and show the results on that.

(d) Run PHATE on just the features of the training data using the default parameters to obtain a 10-dimensional representation of the data. Report the value of t selected using the VNE. Apply k-means to the 10-dimensional representation. This can be viewed as a variation on spectral clustering. Compute the ARI between your cluster outputs and the true labels of the data points. Choose one of the PHATe plots from part (a) and plot the PHATE visualization colored by the cluster labels. Based on the visualization and the ARI value, which of the three clustering approaches does the best?

Note: You may need to do subsampling to make this computationally feasible. If you do, repeat the clustering for multiple (say 10-20) random subsamples and report the average ARI. For the PHATE plot, choose one of the subsamples and show the results on that.

(e) 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 for spectral clustering. What is another way you could tune the bandwidth without using cluster or class labels? You may have to do some research to answer this question.

5. **Information theory (10 pts).**

(a) **(6655 only)** Consider a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. Show that the differential entropy is equal to $\frac{d}{2}\log(2\pi e) + \frac{1}{2}\log\det(\Sigma)$. **Hint:** you may find the trace operator helpful.

(b) Consider two multivariate Gaussian random variables X and Y with means $\mu_X \in \mathbb{R}^{d_X}$ and $\mu_Y \in \mathbb{R}^{d_Y}$, covariance matrices $\Sigma_X \in \mathbb{R}^{d_X \times d_X}$ and $\Sigma_Y \in \mathbb{R}^{d_Y \times d_Y}$, and cross covariance matrices Σ_{XY} and Σ_{YX}. Find an expression for the mutual information between X and Y. **Hint:** you may find the result in part (a) helpful.

6. **Outlier Detection with Kernel Density Estimation (5810 - 45, 6655 - 30 pts).** Download the `anomaly.mat` file from Canvas. This file has training data sampled from a univariate density f stored in the variable `X` and two test points `xtest1` and `xtest2`. The goal of this problem is to calculate a KDE of the density f using the training data and use the KDE to determine whether the two test points are outliers/anomalies.

(a) Using least squares leave-one-out cross-validation with a Gaussian kernel and the training data, select a value for the bandwidth parameter. Report the bandwidth parameter.

(b) Using the training data, estimate the density f at points uniformly spaced between -2 and 4 using a KDE with a Gaussian kernel and the bandwidth parameter selected in part (a). Include a plot of the KDE. Choose enough points between -2 and 4 so that the plot looks smooth. Based on the KDE, what do you think the true density f looks like?

Do not use packages that automatically compute the KDE but use the equations from class

instead. You may use packages that calculate distances efficiently. If in doubt about a specific package, you can ask on Piazza.

(c) There are multiple approaches for using the KDE in anomaly detection. Here is one approach. Let N be the number of points in the training data. Let x_j, $j = 1, \ldots, N$ be the training data. Let x be a point that you wish to test. Define an outlier score for x as

$$OutlierScore_1(x) = \frac{\hat{f}_h(x)}{\frac{1}{n}\sum_{j=1}^{N} \hat{f}_h^{(-j)}(x_j)}.$$

See lecture slides for a definition of $\hat{f}_h^{(-j)}(x_j)$. Provide a conceptual interpretation of this score. I.e., would the score be low or high if x is an outlier? What if x is not an outlier? Are there any potential weaknesses of this score?

(d) Calculate the $OutlierScore_1$ for `xtest1` and `xtest2` using the same h you selected in part (a). Based on the calculated scores, do you think either of these points are outliers?

(e) $OutlierScore_1$ has some weaknesses. Let's define another score based on k-nearest neighbors. Let $\rho_k(x)$ be the distance of x to its kth nearest neighbor in the training data. Let $\mathcal{N}(x)$ be the set of k nearest neighbors of x in the training data. So $|\mathcal{N}(x)| = k$. Define another outlier score for x as

$$OutlierScore_2(x) = \frac{\rho_k(x)}{\frac{1}{k}\sum_{x_i \in \mathcal{N}(x)} \rho_k(x_i)},$$

where $\rho_k(x_i)$ is the distance of x_i to its kth nearest neighbor in the training data (not including x_i). Provide a conceptual interpretation of this score. I.e., would the score be low or high if x is an outlier? What if x is not an outlier? How is this score related to k-nn density estimation? What potential advantages would this score have over the one defined in part (c)?

(f) Calculate $OutlierScore_2$ for `xtest1` and `xtest2` using $k = 100$, 150, and 200. Based on the calculated scores, do you think either of these points are outliers?

(g) Include your code.