Homework 6

Due May 22, 2020 by 11:59pm

Instructions: Upload your answers to the questions below to Canvas. Submit the answers to the questions in a PDF file and your code in a (single) separate file. Be sure to comment your code to indicate which lines of your code correspond to which question part. There are 3 reading items and 2 exercises in this homework, including 1 exercise related to the data competition. The reading item is the preparation for the Week 8 class.

Reading Assignment

- Review Lecture 7.
- Review Computer Lab. 7 in *canvas.uw.edu/courses/1371621/pages/course-materials* .
- Read Section 6.2.2 in *An Introduction to Statistical Learning* .

1 Exercise 1

In this problem you will generate simulated data and then perform principal component analysis on the data. For this purpose, you will write and then use *your own power iteration algorithm.* The Power Iteration algorithm returns the top pair of eigenvalue λ and eigenvector v of a matrix A.

Algorithm 1 Power Iteration Algorithm

```
initialization  $v_0$ random vector, and large number $N$ .
repeat for $k = 1, 2, 3, \cdots, N$
    • Perform update $z_k = Av_{k-1}$ ,
    • Perform update $v_k = \frac{z_k}{\|z_k\|_2}, \lambda_k = v_k^T A v_k$ .
until the stopping criterion is satisfied.
```

Note that “first two principal component score vectors” refers to the results from projecting the original data to a two-dimensional space with principal component analysis (PCA).

(a) Generate a simulated data set with 100 observations in each of three classes (i.e. 90 observations total), and 20 features. Hint: There are a number of functions in numpy that you can use to generate data. One example is the `numpy.random.normal()` function; `numpy.random.uniform()` is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

(b) Write *your own power iteration algorithm*. Run it on the 100 observations to compute the first principal component. After performing the appropriate deflation or projection, run it on the 100 observations now to compute the second principal component. Plot the first two principal component score vectors. To check your algorithm, apply it to a matrix defined row-wise as $A = [1\,2; 0\,3]$, with initial column vector defined row-wise as $v_0 = [1; 2]$. You should get the column vector defined row-wise as $v_1 = [0.64; 0.77]$.

(c) Compare your results to the ones obtained with scikit-learn's PCA algorithm. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then you're done. If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes.

(d) (Bonus) Note that the Power Iteration algorithm finds a vector v so that $v^T A v$ is maximal. What stopping criterion, stated in terms of optimization accuracy ε, other than a maximum number of iterations, can you suggest for your own power iteration algorithm?

2 Data Competition Related Exercise

Read the announcement "Data Competition" released on Canvas. You will use *your own ℓ_2^2-regularized logistic regression* for this exercise.

(a) Download the data for the Kaggle competition.

(b) Pick two classes of your choice from the dataset. You will be training an ℓ_2^2-regularized logistic regression classifier. Find the value of the regularization parameter λ using Scikit-Learn with 5-fold cross-validation.

(c) Train a classifier using ℓ_2^2-regularized logistic regression on the training set using **your own fast gradient algorithm**.

(d) Run your code on AWS. **Please see the instructions at the end of this document.** Take a screenshot of (1) the output and (2) your instances on this webpage: `https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId`

(e) Plot, with different colors, the *misclassification error* on the training set and on the test set vs. the regularization parameter. Note that to obtain the performance on the test set you will need to submit to Kaggle, and you can only submit three times per day.

(f) (Bonus) Plot, with different colors, the *specificity* and the *sensitivity* on the training set and on the test set vs. the dimension of the projection. Plot, with different colors, the *specificity* and the *sensitivity* on the training set and on the test set vs. the percentage of the variance explained.

AWS Instructions

Launching an instance. When launching an instance, please follow the following instructions:

1. Choose the AMI titled **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type**.
2. Choose a **t2.2xlarge** instance. This instance has 8 vCPUs.
3. Leave the spot instances box unchecked. You do not have access to those with an AWS Educate account.
4. Increase the size of your storage, if desired. 50GB should probably be enough.
5. Uncheck the **Delete on termination** box in order to have your disk be saved after you stop or terminate the instance.

Installing of Anaconda and package dependencies. This machine image has very little pre-installed on it. You are likely going to want Anaconda. The script we provided you (in the location where you downloaded this assignment) will install it. To run the script on the remote machine, do the following:

1. Log into the instance.
2. Either upload the installation script or copy/paste it into a new file. To do this with vim:
   (a) Run **vim install.sh**
   (b) Right click and paste the content of the script.
   (c) Press **Esc**, followed by **:wq**. This will save the file.
3. Change the file permissions: **chmod 700 install.sh**
4. Run the script: **./install.sh** It will take a minute or two to run.
5. Run **source ∼/.bashrc**
6. Activate the conda environment: **conda activate data558**

Opening a Jupyter notebook on the remote machine.

1. Run **jupyter notebook –no-browser –port=8888**. If 8888 isn't available try another one, e.g., 8889.
2. Run the following in another terminal, replacing the red parts by your corresponding key name and DNS, respectively. If you changed 8888 in the above line you will also need to change it in the blue part here.
   ssh -N -i /.ssh/data558_sp2020.pem -L localhost:8888 :localhost:8888 ubuntu@ec2-34-234-215-160.compute-1.amazonaws.com
   If it doesn't do anything, that's expected. If the first (magenta) 8888 doesn't work (throws an error related to port), try changing it to another one e.g., 8889.

3. Go back to the original terminal where you are connected to your instance and copy the URL that looks like this:
   `http://localhost:8888/?token=9047c9934f8c000174469127713fdfe1c7aa5bc0907488e4`
   Paste it into your local internet browser. If you changed the dark blue value from above you will need to change the 8888 here to that value.

Don't forget to stop or terminate your machine when you are done!

Additional help. If you need help connecting to your instance on Windows, check out Harsha's great discussion board post. If you need other help, you can check out this AWS tutorial but note that some things, like the AMI and instance used there, are only available to people who have a regular AWS account rather than an AWS Educate account.

We are also happy to respond to questions about AWS on the discussion board and provide individual help during the TA office hours.