Project

Due Friday, December 4, 2020

Instructions:

1. Answer all questions and **show all work** by connecting the data procedures, and answers with Linear Algebra language.

2. Please submit your code along with your results. You are allowed to use any programming language you feel comfortable with. Make sure your code is well documented and cleaned up.

3. You are allowed to work and collaborate, however, verbatim answers (including the code) will be penalized and will be considered a violation of the Honor Code.

4. Good luck, enjoy the project questions, and try to learn as much as you can while in the process!

Questions:

1. Natural images present a simple and intuitive example of this inherent compressibility. A grayscale image may be thought of as a real-valued matrix $X \in \mathbb{R}^{n \times m}$, where n and m are the number of pixels in the vertical and horizontal direction, respectively. Depending on the basis of representation images may have very compact approximations. Capture a selfie without a rich background (to make it easier just take a picture of you in front of a wall), convert RBG to gray (256 bit to double) and compress the image using SVD. Plot several different images using different ranks, and examine how the resolution changes. Make sure to comment what rank is sufficient to store the image (identify an elbow point if possible). What do you observe?

2. The ovarian cancer data set, which is built into Matlab (`https://www.mathworks.com/help/deeplearning/ug/cancer-detection.html;jsessionid=627f60893b3f0b28b252d72a6dbd`), provides a realistic example to illustrate the benefits of PCA. This example consists of gene data for 216 patients, 121 of which have ovarian cancer, and 95 which do not. For each patient, there is a vector of data on the expression of 4000 genes. Use this data set and find the principal vectors as well as plot the data against them. Explain your thought process. Can you distinguish the two different groups from that plot?

3. Collect your own data about a few homes around a neighborhood which are for sale. For example, collect data on sq. ft. per floor separately from the basement, if any, number of bedrooms, number of bathrooms, sq. ft living, if it has a pool or not, number of car

garages, etc. Be very descriptive on your choices, what kind of data you collected for analysis and how many. Then use LASSO by employing the coordinate descent algorithm to analyze the data with price being the response variable. Write down the necessary LASSO optimization equations and its coordinate descent algorithmic implementation. Attempt to be as detailed as possible, and then report on your findings using layman's words *[Friendly Advice: do not forget this problem when you buy your own house in the future :)].*

4. Online retail is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers. Please find the data set here `https://www.kaggle.com/hellbuoy/online-retail-customer-clustering`. Use the $k-$means algorithm to find the best set of customers which the company should target. Please discuss your modeling strategy and parametric choices.

5. Consider two functions $f_1(x, y) = (x-2)^2 + (y-3)^2$ and $f_2(x, y) = [1-(y-3)]^2 + 20[(x+3) - (y-3)^2]^2$. Starting with $(x, y) = (0, 0)$ run the gradient descent algorithm for each function. Run for T iterations, and report the function value at the end of each step. (a) First, run with a fixed learning rate of $s = 0.5$. (b) Second, run with any variant of gradient descent you want (search the literature and document your choice by describing it). Try to get the smallest function value after T steps. For the function f_1 you are allowed only $T = 10$ steps. For the function f_2 you are allowed $T = 100$ steps.

6. Train a (convolutional) neural network with the following dataset `https://www.kaggle.com/harry418/dataset-for-mask-detection`, which could be used for an automated mask-detection. Then capture your picture and/or your friends' and family's one (one person at a time so the algorithm does not get confused) with and/or without mask and see if the algorithm will classify the picture(s) correctly. Also report the accuracy of the method by following cross-validation methods.