MATH 9880 (Spring 2017)
Coding Project 1

Description

- According to the course syllabus, coding projects consists of 20% of the course grade. For each project, choose and solve 3 questions to receive full credit. Please include all the modeling processes, the necessary calculations you performed, and if applicable, the computer programming codes you wrote. If you have any LaTeX/Word source codes, PDF documents, or computer programming source codes, please submit them to Canvas.
- Problems with straightforward solutions are marked by °. Problems marked by * may be challenging.
- Teamworking is encouraged. You may form a group of up to three members to solve project problems marked by *. For group projects, the following additional requirements apply:
  - You may work with multiple groups, if you are interested at solving multiple project problems marked by *. For example, you may participate in three groups, each formed by different members, to solve three different *-marked problems. You may also participate in one group to solve one *-marked problem, and choose and solve two other problems by yourself.
  - For each problems marked by *, the group should submit a standalone report using LaTeX. You would still need to submit your own report for the rest of problems in your project that are not marked by *.
- In your project report, please cite proper references for any of the literature you used including - but not limited to - journal papers, books, solutions manuals, past homework solutions, and MathOverflow/Math StackExchange entries.

Description

The goal of this project is to understand the implementation of a few big data models. You may use any commercial/open source (e.g., CVX, MOSEK, CPLEX, TensorFlow) software package to solve the models associated with the questions.

Problems

1.* Any suitable applications of big data analysis is acceptable. Please contact me if you would like to solve one application of your choice as the project.

2.° (Compressive sensing) We learned in class (Yibo's presentation) the seminal work by Candes, Romberg, and Tao[1]. In particular, we may use the following model to recover a sparse vector x_{true} from noisy observation $b = Ax_{true} + \varepsilon$:

$$\begin{aligned} &\min \ \|x\|_1 \\ &\text{s.t.} \ \ \|Ax - b\|_2^2 \le \delta^2. \end{aligned} \tag{1}$$

Here $A \in \mathbb{R}^{m\times n}$ and δ is a parameter for the model. In particular, if $\delta = 0$, the above problem becomes a linear program

$$\begin{aligned} &\min \ \|x\|_1 \\ &\text{s.t.} \ \ Ax = b. \end{aligned} \tag{2}$$

a) Use the linear programming model (2), solve the nonlinear programming problem with

$$A = \begin{pmatrix} 0 & 2 & 0 & 1 & 0 \\ 1 & -2 & 3 & 2 & -1 \end{pmatrix}$$

and the following choices of b:

i. $b = (-2, 23)^T$.

ii. $b = (20, -20)^T$.

b) Replace the choices of b above by $b + \varepsilon$ where $\varepsilon \sim N(0, \zeta^2 I)$, and use model (1) to recover the true sparse vector. Try solving the model using different values of noise parameter ζ and model parameter δ. What is the best choice of δ in the model? What will happen when $\delta \ll \zeta$ or $\delta \gg \zeta$?

3.° (Compressive sensing, continued) Note that the nonlinear model (1) in the previous question presents an conceptually appealing possibility. Considering two agents, namely, a sender and a receiver, both of whom have the knowledge of the matrix A. Suppose that the sender would like to transmit a message consisting of an

[1]See Candes, Romberg, and Tao, *"Stable signal recovery from incomplete and inaccurate measurements"*.

n-dimensional vector $\bar{x}$ to the receiver, but is unable to do so due to limited transmission bandwidth. Instead, by (1), the sender could opt to transmit a shorter, m dimensional vector $b = A\bar{x}$, where $m < n$. Now if the solution of (1) is still $\bar{x}$, then the receiver would be able to decode the correct message by solving (1). Indeed, in our numerical example, we could see that two original messages of dimension 5 are compressed to vectors of dimension 2.

The above possibility has been confirmed, thanks to Candes, Romberg, and Tao's seminal work. The condition of correct decoding of the original message is that the original message $\bar{x}$ has to be sparse - in the sense that it has many zeros - and that A has to be carefully designed. The theory around the nonlinear problem (1) is commonly known as compressive sensing.

a) Use model (1) to recover the sparse vector x from the observation b. Please use the dataset of A and b in Canvas. Will the performance of recovery change when we tune the model parameter δ?

b) Model (1) can also be formulated as

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{3}$$

where λ is a model parameter. What is the relationship between the above model and the model (1)? Use the above model to recover the sparse vector x from the observation b. Please use the dataset of A and b in Canvas. Will the performance of recovery change when we tune the model parameter λ?

c) From compressive sensing point of view, we are trying to recover a sparse signal. From linear regression point of view, can you give an alternative description of the meaning of model (3)?

4.° In Fisher's 1936 paper "The use of multiple measurements in taxonomic problems", he used a dataset that consists of the measurements of type, petal width, petal length, sepal width, and sepal length for a sample of 150 irises collected by Edgar Anderson. In particular, there are 50 samples from each of three species of iris (iris setosa, iris virginica and iris versicolor). Figure 1 is a plot of the dataset, cited from Wikipedia entry "Iris flower data set".

Perform the following tasks on classifying iris virginica and iris versicolor:

- Choose 40 samples from iris virginica and 40 samples from iris versicolor to form the training dataset. Save the remaining samples of iris virginica and iris versicolor as the testing dataset.
- Use the training dataset, compute the parameters of a classification model (e.g., logistic model or SVM).
- Use your model with computed parameters to classify the samples from the training dataset. Report the accuracy of your model in terms of its correctness in classification.

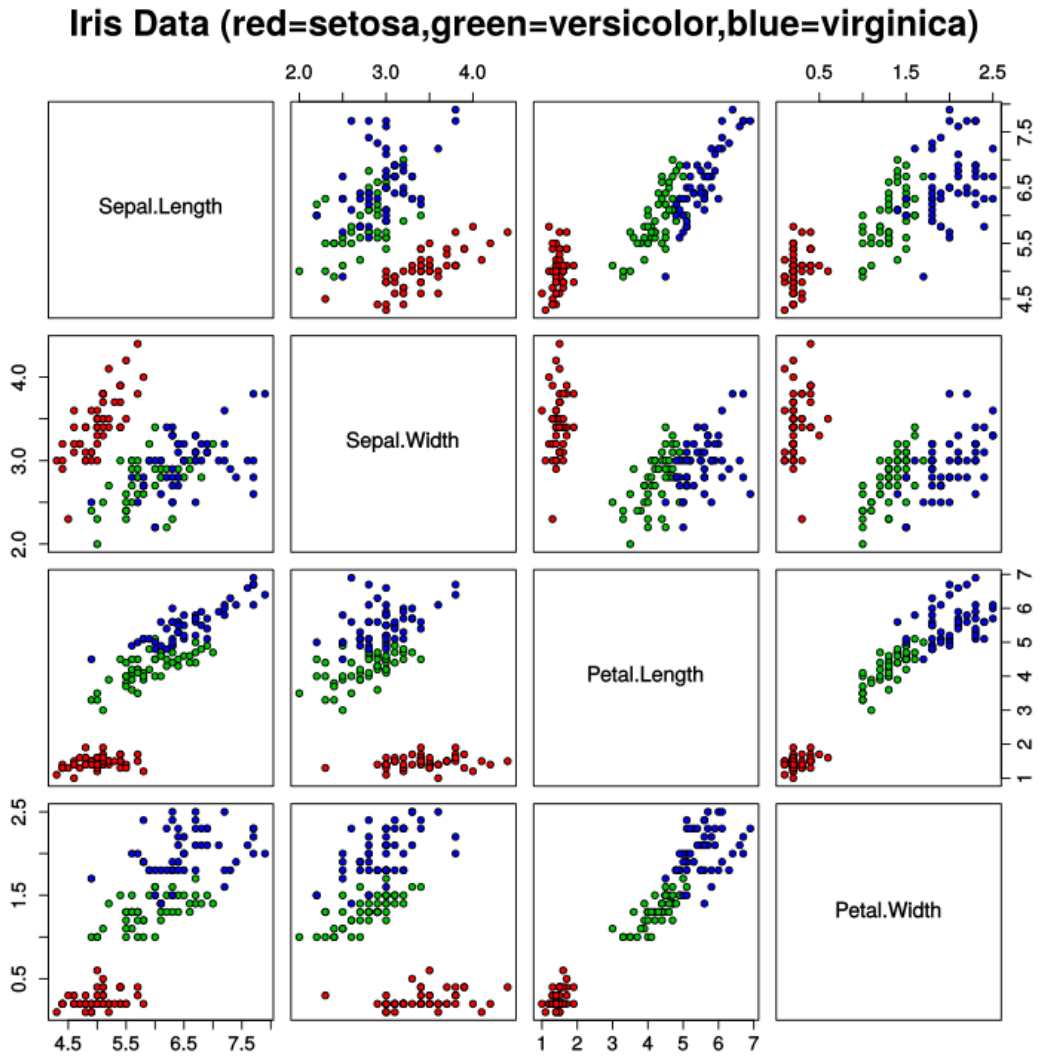

Figure 1: The illustration of Fisher's iris data

Figure 2: Sample of images in the MNIST dataset.

5.° Using any model we learned in class (e.g., logistic model or SVM) to perform the handwritten digit recognition on the MNIST dataset. The training dataset consists of images of handwritten digits from 0 to 9. Each image is of dimension 28x28, so $n = 784$ for our models. See below a sample of images in the MNIST

dataset. More details on the MNIST dataset is explained in its website (http://yann.lecun.com/exdb/mnist/). Can you use your model to separate digits 3 and 8?

6. In this question, we will perform a classification of three possible classes[2].

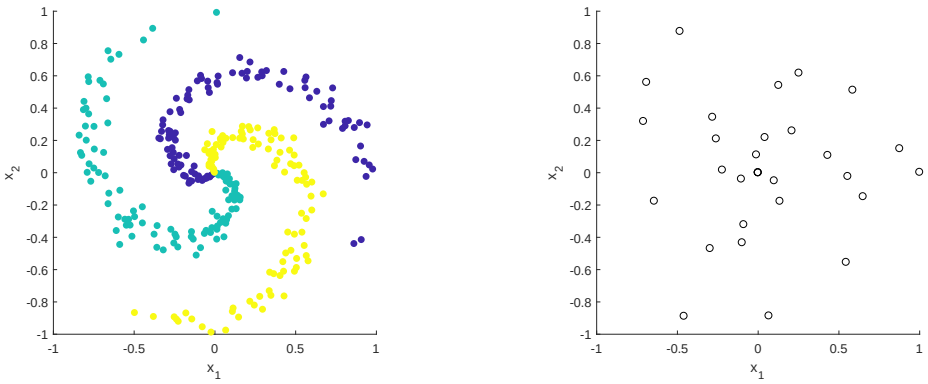

Figure 3: Dataset for a three-class classification problem. Left: three classes of datasets. Right. Newcomers. Can you correctly classify the newcomers to the three classes?

Design a model to classify the newcomers in Figure 3 to their respective classes (see the model described at http://cs231n.github.io/neural-networks-case-study/).

7.* Let us perform a image classification task using the built-in image dataset in Matlab. There is a small image dataset of 5 different objects (Mathworks logoed cap, cube, playing cards, screwdriver, and torch). For each object, there are 15 pictures of them. See Figure 4 for the pictures of the 5 objects.

Use 11 images from each class of images to train a model in order to classify them. Use the remaining images to verify the accuracy of your classification model (Read the Matlab help https://www.mathworks.com/help/nnet/examples/feature-extraction-using-alexnet.html).

8.* (Big data visualization) We are able to visualize the dataset when they are of low dimension. For example, in Figure 1 we can visualize the clusters of different sepal length and width and petal length and width for different types of iris flower. Also, in Figure 3 we can easily illustrate the 3 difference clusters of data. However, when the dimension of the learned data is higher, it becomes difficult to visualize them.

a) For the Fisher iris problem (see Problem 4), the measurement dataset should be of dimension 4: petal width, petal length, sepal width, sepal width. Can

[2]This project question is from the CS231n (Convolutional Neural Networks for Visual Recognition) course designed by Fei-Fei Li et al., Stanford university.

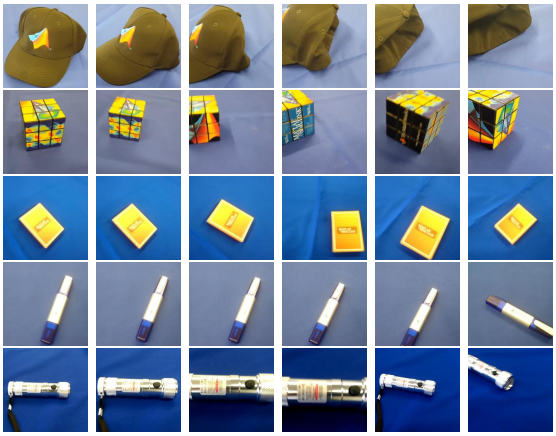

Figure 4: Images from the small datasets in Matlab. From top to bottom: Mathworks logoed cap, cube, playing cards, screwdriver, and torch.

you find a way to illustrate the difference of the three iris flower types through their 4-D data of measurements?

b) For the NMIST problem (see Problem 5), can you visualize the difference of the digits through their images?

c) Can you do the same with the Matlab image dataset in Figure 4 (see Problem 7)?

Remark: Read presentation topic #13.

9.* In this problem, we will learn TensorFlow, a widely used machine learning tool, and run it on the Palmetto cluster.

a) Register a Palmetto cluster account, and install the TensorFlow environment with GPU implementation. You may also set up the Clemson JupyterHub kernel. The guide for installation is available on Canvas (thanks to Yuanxun). If you setting-up process is different, please also include a description of your setting-up process in your submission.

b) (Optional) If you would like to run TensorFlow in your own computer, you may also install it following the guide at the following website: `https://www.tensorflow.org/install/`

c) Use TensorFlow to solve the Fisher iris classification in Problem 4 (you may follow `https://www.tensorflow.org/get_started/get_started_for_beginners`).

d) Use TensorFlow to solve the MNIST classification in Problem 5 (you may follow `https://www.tensorflow.org/tutorials/layers`).

Remark: It is recommended to solve Problem 6 first, so that we have background on some multi-layer neural network before solving this problem.

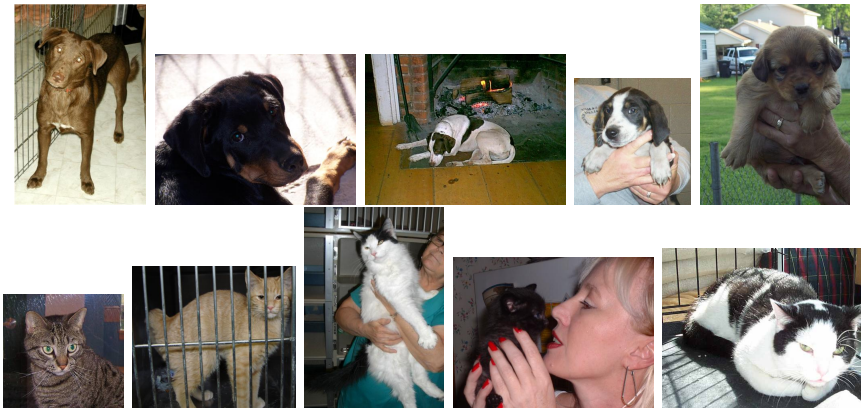

Figure 5: Images of dogs and cats in the dataset.

10.* In this problem, we will attempt to work on Kaggle's "Dogs vs. Cats" challenge, and design a binary classification model to classify images of cats and dogs. See Figure 5 for some images in the dataset.

It should be noted that this is indeed an extremely challenging task, since the image dataset in this problem has not been preprocessed. Not only that the images could be of different size and color, but some image files may be corrupted and unreadable. Download the Kaggle cats and dogs dataset from Microsoft website (https://www.microsoft.com/en-us/download/details.aspx?id=54765). Use 80% of the images to train your model, and the remaining images to verify the accuracy of your model.