CS5785 Homework 1

The homework is generally split into programming exercises and written exercises.

This homework is due on **September 14, 2016 at 11:59 PM EST**. Upload your homework to CMS. Please upload code as a single `.zip` file and the writeup as a single `.pdf` file. A complete submission should include:

1. A write-up as a single `.pdf` file
2. Source code and data files for all of your experiments (AND figures) in `.py` files if you use Python or `.ipynb` files if you use the IPython Notebook. If you use some other language, include all build scripts necessary to build and run your project along with instructions on how to compile and run your code.

The write-up should be in professional lab report format. More specifically, it should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, homework number, and team member names. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and and "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them.

Please pay attention to the discussion board for relevant information regarding updates, tips, and policy changes. You are encouraged (but not required) to work in groups of 2.

IF YOU NEED HELP

There are several strategies available to you.

- If you ever get stuck, the best way is to ask your teammates on Piazza[1]. That way, your solutions will be available to the other students in the class.
- Your instructor and TAs will offer office hours[2], which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. in this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

[1] http://piazza.com/cornell/fall2016/cs5785

[2] http://cs5785-cornell-tech-16fall.github.io/contact.html

PROGRAMMING EXERCISES

1. Digit Recognizer

   (a) Join the Digit Recognizer competition on Kaggle. Download the training and test data. The competition page describes how these files are formatted.

   (b) Write a function to display an MNIST digit. Display one of each digit.

   (c) Examine the prior probability of the classes in the training data. Is it uniform across the digits? Display a normalized histogram of digit counts. Is it even?

   (d) Pick one example of each digit from your training data. Then, for each sample digit, compute and show the best match (nearest neighbor) between your chosen sample and the rest of the training data. Use L_2 distance between the two images' pixel values as the metric. This probably won't be perfect, so add an asterisk next to the erroneous examples.

   (e) Consider the case of binary comparison between the digits 0 and 1. Ignoring all the other digits, compute the pairwise distances for all genuine matches and all impostor matches, again using the L_2 norm. Plot histograms of the genuine and impostor distances on the same set of axes.

   (f) Generate an ROC curve from the above sets of distances. What is the equal error rate? What is the error rate of a classifier that simply guesses randomly?

   (g) Implement a K-NN classifier.

   (h) Using the training data for all digits, perform 3 fold cross-validation on your K-NN classifier and report your average accuracy.

   (i) Generate a confusion matrix (of size 10 × 10) from your results. Which digits are particularly tricky to classify?

   (j) Train your classifier with all of the training data, and test your classifier with the test data. Submit your results to Kaggle.

2. The Titanic Disaster

   (a) Join the Titanic: Machine Learning From Disaster competition on Kaggle. Download the training and test data.

   Pandas: working with missing data

   (b) Using logistic regression, try to predict whether a passenger survived the disaster. You can choose the features (or combinations of features) you would like to use or ignore, provided you justify your reasoning.

   (c) Train your classifier using all of the training data, and test it using the testing data. Submit your results to Kaggle.

Written Exercises

1. Variance of a sum. Show that the variance of a sum is $var[X+Y] = var[X] + var[Y] + 2cov[X,Y]$, where $cov[X,Y]$ is the covariance between random variables X and Y.

2. Bayes rule for medical diagnosis (Source: Koller) After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease, and that the test is 99% accurate (i.e., the probability of testing positive given that you have the disease is 0.99, as is the probability of testing negative given that you do not have the disease). The good news is that this is a rare disease, striking only one in 10,000 people. What are the chances that you actually have the disease? (Show your calculations as well as giving the final result.)

3. Gradient and Hessian of log-likelihood for logistic regression.

   (a) Let $\sigma(a) = \dfrac{1}{1+e^{-a}}$ be the sigmoid function. Show that $\dfrac{d\sigma(a)}{da} = \sigma(a)\,(1-\sigma(a))$.

   (b) Using the previous result and the chain rule of calculus, derive the expression for the gradient of the log likelihood given in HTF Eqn. 4.21.

   (c) As noted in HTF Eqn. 4.25, the Hessian matrix for the log likelihood can be written (up to a sign) as $\mathbf{X}^\top\mathbf{W}\mathbf{X}$. Prove that this matrix is positive definite.