CSCI 635: Introduction to Machine Learning

Assignment 1, Fall 2020

Due Date: Sunday, October 4, 11:59pm

Late: -20% until October 5, 11:59pm — *Afterward*: grade of 0

Notes

- **The assignment is out of 50 points.**
- Submit your assignment through the Dropbox in MyCourses **as two files:**
  - A .pdf file for the write-up, named **a1.pdf**.
  - A .zip file containing your code, trained parameter values, and a README explaining (briefly!) how to run your trained classifiers, named **a1.zip**.
- Code must be able to run servers for the course, granger.cs.rit.edu or weasley.cs.rit.edu

Grade Penalties will be applied for:

- Not submitting the write-up as instructed above.
- Submitting code with incorrect file names.
- Not providing trained parameter values - we should be able to easily run your programs using the trained parameter values - we will not retrain your networks.
- Submitting code that cannot run on the class servers (*this penalty will be substantial).*

Question 1 - Data Analysis and Visualization (20 points)

For Questions 1 and 2, we will use two data sets (available through MyCourses):

- *Frogs-subsample.csv,* and
- *Frogs.csv,*

The features are Mel Frequency Cepstrum Coefficients (MFCCs) representing two frequency band intensities measured from South American frog calls in audio recordings. There are two species of frog in the data sets. This data was selected from the Anuran Calls dataset available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29. The subsampled data set was producing by obtaining a (class-)balanced random sample (25 instances per class).

a. **Visualization. Separately** for each of *Frogs-subsample.csv* and *Frogs.csv*, use *numpy* and *matplotlib* to produce the following:

- **Plotting Raw Features**
  - A scatter plot of the 'raw' samples (both features, separate color for each class)
  - **For each frog class in the file:**
    - 2 histograms (1 per feature/attribute)
    - 2 line graphs (1 per feature/attribute - *after sorting* feature values)
- **Plotting Feature Distributions**
  - A boxplot showing the distribution of features for both classes (For each class, 1 box+whiskers per feature; 4 boxes total)
  - Bar graph with *error bars* (For each class, 1 error bar per feature; 4 errors bars total)

b. **Descriptive Statistics.** Separately for each data set, use numpy to compute 1) the mean (*expected value*), 2) covariance matrix, and 3) standard deviation for each individual feature.

In the write-up, provide one or two tables that *clearly* and *attractively* allow the plots of each type to be easily compared visually (e.g., putting the two scatter plots beside one another). Provide another (text) table providing the descriptive statistics from part b. for each dataset. Then discuss the distributions of the features in the two data sets, **making direct reference** to your plots and descriptive statistic tables. In what ways are the distributions of the two classes similar or dissimilar? In what ways are the class distributions in the two data sets similar or dissimilar?

!! **Fair Warning !!** the content and clarity of both the visual presentation and text will be considered when grading. If the presentation is vague, messy, unclear, and/or incorrect, a grade penalty will be incurred.

Name your program for generating plots and statistics q1.py, and include this in a1.zip.

Question 2 - The Effect of Training Data (15 points)

Using numpy (or PyTorch) create a binary classifier for the data in each file using a *single* logistic regressor (i.e., a single 'perceptron' using the sigmoid activation function). Create a scatter plot for each data set, and then visualize the class regions and decision boundaries (similar to the Hastie et al. figures seen in class).

In the write-up, present the two plots in a table. Discuss the decision boundaries that you obtained, and both *how* and *why* the different data sets produced the results obtained. 2-3 paragraphs should be sufficient.

Name your program q2.py, and include the program along with the saved parameters for your networks in a1.zip.

Question 3 - Let Us Not Forget Probability! (15 points)

• In creating a product, 85% are produced without defects. Of the products inspected, 10% of the good ones are seen as defective and not shipped, while only 5% of the defective products are approved and shipped. If a product is shipped, what is the probability that it has a defect?

• Consider randomly generated bit strings of length four. Demonstrate whether or not the event of generating bit string with an even number of 1's is *independent* of the event producing bit strings that end in 1.

• Let's flip a (fair) coin n times to generate a dataset, where we choose to represent the state of the coin, i.e., heads or tails, as the variable X = {0, 1}, where the first attribute value represents "tails" and the other "heads". Suppose that the experiment's outcome yields the following state sequence:

S = {1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0}

Estimate the probability that p(X = 1) from the data, using a maximum likelihood approach (Hint: the frequency approach). Also, what is the probability of getting tails under the sequence above, or p(X = 0)?

Bonus: Provide a maximum a posteriori (MAP) estimate of the probability of getting a heads, p(X = 1) assuming this prior belief about the coin being fair. (Hint: Adapt your MLE estimate to account for your prior, and since this is a coin toss, or a Bernoulli random variate.)

Put your answers into a1.pdf and make sure your questions for Q3 are cleanly organized (and show your work/calculation steps). You may scan your math if hand-written (though LaTex is a better choice) BUT your writing must be clear or points will be deducted for sloppiness & poor readability.