CSE 5523: Problem Set 4

Due via Carmen dropbox at 11:59 PM 10/25.

1. PROGRAMMING:

   (In general for programming problems you should hand in the following as separate files:

   - a copy of each program file you write,
   - a representative sample of each input file you use,
   - a representative sample of each output you produce.

   Your programs should be as short as possible, and may be based on linear algebra and data analysis functions used in the lecture notes, but should not use higher-level packages or functions.)

   You want to see whether momentum-based optimization is helpful on your data.

   (a) [15 pts.] Write a program called 'mome.py' that implements a logistic regression classifier using the following momentum-based optimizer:

   $$\begin{aligned} \mathbf{M}^{(0)} &\overset{\text{def}}{=} \mathbf{0} \\ \mathbf{M}^{(i)} &\overset{\text{def}}{=} \beta_M \, \mathbf{M}^{(i-1)} + (1 - \beta_M) \, \nabla_{\mathbf{W}^{(i-1)}} \\ \mathbf{W}^{(i)} &\overset{\text{def}}{=} \mathbf{W}^{(i-1)} - \alpha \mathbf{M}^{(i)} \end{aligned}$$

   similar to the one described in the lecture notes on optimization for gradient descent. Your program should take the following as command-line arguments in the following order:

   i. a filename of a csv file containing variable dimensions of training data points,

   ii. a filename of a csv file containing the same dimensions of test data points minus the first column,

   and output a csv file containing one dimension (the first column of the input) of predicted values. Your program may be based on the example logistic regression code in the lecture notes. Note that you may need to run 1000 or so epochs of gradient descent for your model to converge.

   (b) [5 pts.] Run your regressor and the un-optimized logistic regression code from the lecture notes on the 'iris-train.csv' and 'iris-test.csv' data on the course web page. Evaluate the accuracy of your program on the 'iris-test.csv' data as compared with un-optimized logistic regression in the lecture notes. Do you notice a statistically significant improvement for using momentum-based optimization? You may use the code at the end of this problem set for credible interval determination and code from the lecture notes for significance testing.

   (c) [5 pts.] Train both regressors on the 'iris-train.csv' data on the course web page and evaluate the speed in terms of clock time for a given number of epochs. Do you notice any improvement for using momentum-based optimization?

2. PROGRAMMING:

   You are told that the below function makes a good substitute for the logistic function as an activation/transfer function in a neural network:

$$\mathbf{z}_{[i]} = \ln(1 + e^{(\mathbf{W}\,\mathbf{x})_{[i]}})$$

   (a) [5 pts.] Calculate the derivative of the above function.

   (b) [15 pts.] Write a program called ‘substnet.py’ that implements a two-layer neural network classifier (i.e. with one hidden layer) using five hidden units, a logistic activation/transfer function at the output layer, and using the above activation/transfer function at the hidden layer. Your program should take the following as command-line arguments in the following order:

      i. a filename of a csv file containing variable dimensions of training data points,

      ii. a filename of a csv file containing the same dimensions of test data points minus the first column,

   and output a csv file containing one dimension (the first column of the input) of predicted values. Your program may be based on the example neural network code in the lecture notes.

   (c) [5 pts.] Run your classifier and the all-logistic neural network classifier from the lecture notes on neural networks on the ‘iris-train.csv’ and ‘iris-test.csv’ data on the course web page. Evaluate the accuracy of both programs on the ‘iris-test.csv’ data. Do you notice a statistically significant improvement for using your new transfer function over the logistic? You may use the code at the end of this problem set for credible interval determination and code from the lecture notes for significance testing.

   (d) [5 pts.] Train both regressors on the ‘iris-train.csv’ data on the course web page and evaluate the speed in terms of clock time for a given number of epochs. Do you notice any improvement for using your new transfer function?

3. [9 pts.] PROJECT:

   Your project may be in any area (vision, natural language, etc), may use any programming language you choose, and may use any data, even data used in previous projects or from data repositories or shared task challenges (e.g. https://archive.ics.uci.edu/ml/datasets.php), but should involve new code that you write to implement techniques you did not use previously and should therefore produce new results. You may work in groups, but in this case you should specify (in item c below) what portion of the project code will be produced by you and the other group members.

   (a) [3 pts.] Describe in just a few sentences what *problem* your project will solve.

   (b) [3 pts.] Describe in just a few sentences what *data* your project will use, including how many items (N) the data include, how many variables (V) are involved, and where your data will come from.

(c) [3 pts.] Describe in just a few sentences what *machine learning techniques* your part of your project will use.

NOTE FOR PROGRAMMING PROBLEMS:

You may use the following code for credible interval estimation:

```
import sys
import numpy
import pandas

numsamples = 1000

X = pandas.read_csv(sys.argv[1])                            ## read 0/1 scores

S = pandas.Series( numpy.random.beta( len(X[ X[X.columns[0]]==1 ]) + 1,
                                      len(X[ X[X.columns[0]]==0 ]) + 1,
                                      numsamples ) )        ## get 1000 samples
m = S.mean()                                                ## get sample mean

for a in S:                                                 ## for each sample
  n = len( S[ abs(S-m) <= abs(a-m) ] )                      ## test as boundary
  if n==numsamples*.95: print( 'mean: ' + str(m) + ' +/-' + str(abs(a-m)) )
```