Homework Assignment 8: Ensemble Learning

Due Friday, April 14, 2023 at 11:59 pm EST

What to submit

Create a folder: ps8_xxxx_LastName_FirstName and add in your solutions (xxxx = python or MATLAB):

`ps8_xxxx_LastName_FirstName/`

- input/ - input images, videos or other data supplied with the problem set
- output/ - directory containing output images and other files your code generates
- ps8.m or ps8.py - code for completing each part, esp. function calls.
- *.m or *.py Matlab/python function files, or any utility code.
- ps8_report.pdf - a PDF file with all output images and text responses

Zip it as `ps8_xxxx_LastName_FirstName.zip`, and submit on Canvas.

Guidelines

1. Include all the required images in the report to avoid penalty.
2. Include all the textual responses, outputs and data structure values (if asked) in the report.
3. Make sure you submit the correct (and working) version of the code.
4. Include your name and ID on the report.
5. **Comment your code appropriately**.
6. Please avoid late submission. Late submission is not acceptable.
7. Plagiarism is prohibited as outlined in the Pitt Guidelines on Academic Integrity.
    a. **Please don't share your codes with any of your colleagues.**

Questions

1. Bagging and Handwritten-digits classification

In this part, you are allowed to use any MATLAB built-in function or Python machine learning Library. **However, you're not allowed to use the classification learner or any code that is generated automatically or copied from anywhere**. In this problem, you will use a short sample of the MNIST dataset (HW8_data1.mat) to build ensemble of classifiers to classify handwritten digits (i.e., 10-class classification problem). The file contains a features matrix X and labels vector y. These data can be read directly into your program by using the `load` command. Each row in X corresponds to one feature vector example. There are 5000 examples in HW8_data1.mat, where each example is a 20 pixel by 20 pixel grayscale image of the digit. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 20 by 20 grid of pixels is "**unrolled**" into a 400-dimensional vector. Each of these examples becomes a single row in our data matrix X. This gives us a 5000 by 400 matrix X where every row is a single feature vector for a handwritten digit image. The second part of the training set is a 5000-dimensional vector y that contains labels for the training set. To make things more compatible with Matlab indexing, where there is no zero index, we have mapped the digit zero to the value ten. Therefore, a **"0" digit is labeled as "10"**, while the digits "1" to "9" are labeled as "1" to "9" in their natural order.

a. Download the mat file to your input directory. Load the data into matlab. Randomly pick 25 images and display them in a 5x5 grid. Save your output figure and include it in your report. Hint: the `resize` function could be useful.

b. Randomly split the data into 4500 samples training set (X_train and y_train) and 500 samples testing set (X_test and y_test).

c. Apply bagging on **the training set** to create five equally sized (1000 samples each) subsets X_1, X_2, X_3, X_4 and X_5; don't forget to save the corresponding label vectors. The samples in X_i ($i = 1, 2, 3, 4, 5$) should be **randomly** picked from the original training set that you obtained in part b. Please save these subsets to local files (in the input folder as .mat files) and include them along with your submission.

d. Train a One-vs-All SVM 10-class classifier, with RPF kernel, using subset X_1. Use default parameters of the radio-basis function (RPF). Use your trained classifier to:

   i. Compute the classification error on the training set X_1.
   ii. Compute the classification error on the other training subsets, i.e., X_2 *to* X_5.
   iii. Compute the classification error on the testing set.

e. Train a KNN ($K = 7$) classifier using subset X_2. Use your trained classifier to:

i. Compute the classification error on the training set X_2.
ii. Compute the classification error on the other training subsets.
iii. Compute the classification error on the testing set.

f. Train a logistic regression classifier using subset X_3 . Use your trained classifier to:

i. Compute the classification error on the training set X_3.
ii. Compute the classification error on the other training subsets.
iii. Compute the classification error on the testing set.

g. Train a decision tree classifier using subset X_4. Use your trained classifier to:

i. Compute the classification error on the training set X_4.
ii. Compute the classification error on the other training subsets.
iii. Compute the classification error on the testing set.

h. Train a random forest (with **75** trees) classifier using subset X_5 (Hint: the `TreeBagger` function may be useful). Use your trained classifier to:

i. Compute the classification error on the training set X_5.
ii. Compute the classification error on the other training subsets, i.e., X_1 to X_4.
iii. Compute the classification error on the testing set.

i. Use the majority voting rule to combine the output of your five classifiers and report the error rate on the testing set.

j. Your report must document the following:
i. Summarize your results.
ii. Discussion and comparison of each classifier performance, and your interpretation of the results. Does bagging help?