Fall 2018 DS 4400

Homework 3

Due date: November 13 2018

Instructions

Please make sure your name appears in all the files and source code you submit. Submit a PDF file named "%LASTNAME%_HW3.pdf" in Gradescope that will include the following:

- Link to Jupyter notebook with Python or R code. You can store the code in an online service (such as Google drive, Dropbox, or private github).
- Link to a simple README file with instructions on how to run your code.
- Answers to the questions. You can use Latex or Word to generate the PDF.

Course policy on collaboration and cheating:

- You may discuss the concepts with your classmates, but write up the answers entirely on your own.
- You cannot share your code with your classmates.
- You cannot use code from the Internet for your assignment.

Dataset: We will use 2 datasets for this assignment:

1. The SPAMBASE dataset from the UCI repository, available at: `https://archive.ics.uci.edu/ml/datasets/spambase`. This is the dataset you used in HW 2.
2. The MNIST hand-written recognition dataset, available from: `http://yann.lecun.com/exdb/mnist`

Problem 1 [Random Forest classifier] - 38 points

Use the SPAMBASE dataset for this problem. Split the original data into 75% for training and 25% for testing (chosen at random).

(a) Use an existing package to train a Random Forest classifier on the training set. Report accuracy, error, precision, and recall on both training and testing sets.

(b) Implement your own Random Forest algorithm. The Random Forest training procedure takes as input the training dataset, the number of trees, and the number of features $m \leq d$ considered at every split (d is the total number of features in the dataset).

(c) Vary the number of features m selected at random at each split. Consider $m = d$, $m = d/2$, and $m = \sqrt{d}$. Report accuracy, error, precision, and recall on the training and testing set.

(d) Fix the number of features $m = \sqrt{d}$. Compare your implementation with the package results for different number of trees (10, 50, and 100).

Problem 2 [AdaBoost classifier] - 24 points

Use the SPAMBASE dataset for this problem. Split the original data into 75% for training and 25% for testing (chosen at random).

(a) Use an existing package to train an AdaBoost algorithm with 50 base classifiers. Use a decision tree as the base classification model. Report accuracy, error, precision, and recall on both training and testing sets.

(b) Change the base classifier to logistic regression, but keep the number of base learners at 50. Report accuracy, error, precision, and recall on both training and testing set.

(c) Compare the performance of the AdaBoost classifier with different number of base learners (10, 50, and 100).

(d) Compare AdaBoost with Random Forest for the same number of base learners (consider 10, 50, and 100).

Problem 3 [Neural Networks] 38 points

Use the MNIST dataset for this problem. The dataset is already split into training and testing data.
For Neural Network implementation, you can use existing packages, such as Keras:
`https://keras.io/`.

(a) Pick 3 configurations of Feed-Forward Neural Networks and describe for each: (1) number of layers; (2) number of hidden units on each layer; (3) activation functions.

(b) Train models for all 3 architectures. Report performance metrics (loss function, accuracy, and error) on both training and testing data.

(c) Pick 3 configurations of Convolutional Neural Networks and describe for each: (1) number of layers; (2) layer type (convolution, max pooling, fully connected) (3) filter size for convolution and max pooling layer; (4) number of hidden units on each layer; (5) activation functions.

(d) Train models for all 3 architectures. Report performance metrics (loss function, accuracy, error) on both training and testing data.

(e) Compare performance of Feed-Forward and Convolutional Neural Networks for this classification task.