EE 660

Homework 12

Posted 11/8/2018

Due: Tues. 11/15/2018, 2:00 PM

1. *Computer problem on CART for 3-class classification*

   This problem is designed to be solved using PMTK in Matlab.

   (a) Run the PMTK demo **dtreeDemoIris.m** and report cross validation errors of the decision tree, before and after pruning.

   If your code prompts an error on line 18 “RandStream.setDefaultStream(s);”, change it to “RandStream.setGlobalStream(s);”

   (b) Now modify the code so the tree takes all 4 features. Show the tree structure. The top levels of your plot should look similar to Figure 1 below, but please include all levels in your plot. No need to prune. Report the cross-validation error.

   Hint: Since you don’t need to draw decision boundaries for the modified tree, you can comment out that part of the code.

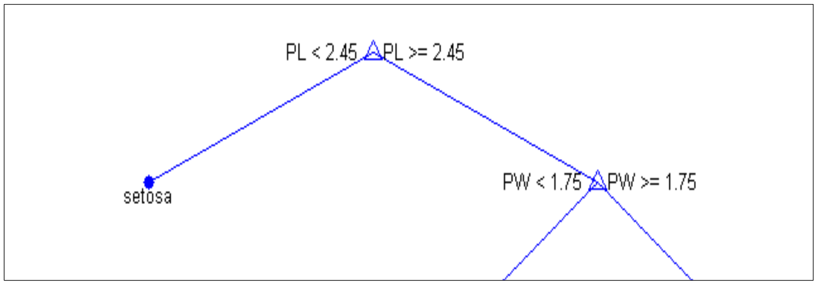

Fig. 1. Top part of modified tree. PL and PW are short for the features “pedal length” and “pedal width”

   (c) Why is the depth of the unpruned modified tree of (b) (based on the full feature set) smaller than the unpruned original tree of (a) (based on the original feature set)?

   (d) Compare cross validation error of your modified tree of part (b) (based on the full feature set, without pruning), to the original trees of part (a) (based on the original feature set, both before and after pruning). Explain what you observe.

2. *CART for regression*

   (a) For CART applied to regression, prove the relation for $w_{m'}^*$ in Lecture 19, middle of page 7. [**Hint:** take the derivative of the cost function in that region, and set it equal to 0.]

(b) For a given region $\mathcal{R}_{m'}$ containing $N_{\mathcal{R}_{m'}}$ data points, and a given feature to threshold x_j, suppose you want to find an optimal threshold value t_k by trying different values; how many values for t_k need to be tried? (Give an upper bound.) Justify your answer.

3. *Random Forest for yeast data classification*

This problem is intended to give some hands on experience using random forest. You are given a dataset adapted from the Yeast Data Set on UCI repository:

https://archive.ics.uci.edu/ml/datasets/Yeast

containing 1484 data points and 9 features, and has been partitioned into a training set of 1000 data points, and a testing set of 484 data points.

The goal is to estimate the **Protein Localization Site** of each instance. The **Protein Localization Site** is a categorical label that takes 10 different values, originally in form of strings, and has been preprocessed into categorical integers for you. In the provided **.mat** file, the original string labels are stored in a cell array "classes". The provided .csv files are in the usual format for Python use.

Matlab users: Train a random forest with the following parameters:

randomFeatures = 3

bagSize = 1/3 (percent of training samples that are used to grow a tree)

ntree = 1 to B, step size 1, B=30.

Python users:

First create a smaller training set, 'bag", randomly drawn from the given training set. Use a size similar to that given for the Matlab users above (1/3 of the training set size). You can use "train_test_split()" or any other technique for this purpose. For instance, you can consider train_size =1/3 in train_test_split() to use for "bag".

Then you can use "sklearn.utils.resample" for randomly sampling from "bag" with replacement, and then train a random forest with the following parameters:

n_estimators = 1 to B, step size 1, B=30.

bootstrap = True

max_features = 3

Everyone:

For each value of number of trees (estimators), repeat the experiment 10 times (selecting different bag samples every time), and calculate the following results:

Mean error rate on testing set

Mean error rate on training set

Sample standard deviation of error rate on testing set

Each of the 3 sets of results should be a 30 by 1 vector.

For this problem, please:

(a) State whether you used Matlab or Python; if you used different routines than given above, state what you used. And, if you needed to use parameter values different than specified above, state clearly what you used and why.

(b) Plot your 3 sets of results (above) against the number of trees as *x*-axis. Explain the results.

Hints:

(1) PMTK provides a brief demo of training a random forest with decision trees. You can find it by searching for "forestProstateDemo" in the pmtk3 folder. You can specify the value of ntree in the function call, e.g.:

fitForest(Xtrain,Ytrain,'randomFeatures', d ,'bagSize', bs ,'ntrees',ntree);

Note: this code may take a few minutes to run.

(2) Python / sklearn users can use:

sklearn.ensemble.RandomForestClassifier

and its method fit(). For more info, see the example in

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

You may also find the following functions useful (feel free to use them):

pandas.get_dummies (to convert categorical features)

sklearn.metrics.accuracy_score (to measure accuracy on training and test data)

matplotlib.pyplot (to show the results)

Reading: Semi-Supervised Learning

This reading is from:

Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2009). [In short, "Zhu"]. It is available for download from USC Library. You can access the book by following these steps:

1. Go to USC Libraries home page at libraries.usc.edu
2. **In the upper right corner, click on "login"**; then enter your info to log in

3. Enter “Introduction to Semi-Supervised Learning” in the library’s search field

3. The first search result should be:

   Xiaojin Zhu (and) Andrew Goldberg, Introduction to Semi-Supervised Learning.

   Click on its “online access” link

4. This should give you the publisher’s page for the book; **note that you must see “USC Libraries” on the upper right side** to be able to freely access the book.

5. Below the abstract, use the link “PDF” or “PDF Plus” to access the book (open in browser, or download the pdf file).

Please read:

Ch. 1, up to end of Sec. 1.2 (pp. 1-3) (background)

Ch. 2, all

Ch. 3, beginning of chapter to end of Sec. 3.5 (pp. 21-31). Note that you may find Sec. 3.3 (EM algorithm) a bit complicated, but we are going through it in lecture also.

Problem on reading

1. For each example learning problem given below, state whether it is a semi-supervised learning problem (as described in Zhu and Goldberg Ch. 1-2), and if so, whether it is inductive or transductive.

   (a) You want to build a machine learning system that can recognize the breed of dog from its picture, hoping to make it a mobile app. You collect a training dataset that includes N pictures of various dogs from Google. You happen to have a friend that is a dog breeder and is willing to label some of the pictures, so she labels l of the pictures for you. You want to use all N pictures to develop your system.

   (b) Same example as part (a), except you want the mobile app to continue to learn from pictures it is given of dogs.

   (c) Instead, you have a relatively small set of pictures of various kinds of ships at sea, that are labeled according to type of ship. You also have a larger set of computer generated drawings of ships, also labeled according to type. You would like to train from both sets of data. The goal is to autonomously recognize ships from pictures taken at sea.

   (d) A large set of pictures of Pluto, taken from a spacecraft as it flew by, has been received and needs to be classified by type of terrain. Some of them have been hand labeled by experts. The goal is to have the computer label the rest of them.