1. *CART for regression*

(a) For CART applied to regression, prove the relation for $w^*_{m'}$ given below (also given in Lecture 18, page 6):

$$w^*_{m'} = \frac{1}{N_{\mathcal{R}_{m'}}} \sum_{x_i \in \mathcal{R}_{m'}} y_i$$

[**Hint:** take the derivative of the cost function in that region, and set equal to 0.]

(b) Also in a regression problem, 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.

1. (a) $\text{cost}\{(x_i, y_i) \in R_m\} = \sum_{x_i \in R_{m'}} (y_i - w_{m'})^2 = \| \underline{y}_{m'} - w_{m'} \cdot \underline{1} \|_2^2$ (for a given $R_{m'}$)

$\underline{y}_{m'}$: the vector which is composed of y_i in $R_{m'}$. ($\underline{1}$: $N_{R_{m'} \times 1}$ 1 vector; $\underline{1}^T_{(N_{R_{m'} \times 1})} \times \underline{1}_{(N_{R_{m'} \times 1})} = N_{R_{m'}}$)

$$\frac{\partial \| \underline{y}_{m'} - w_{m'} \cdot \underline{1} \|_2^2}{\partial w_{m'}} = \frac{\partial [\underline{y}^T_{m'} \underline{y}_{m'} - \underline{y}^T_{m'} w_{m'} \underline{1} - w_{m'} \cdot \underline{I}^T \underline{y}_{m'} + w^2_{m'} N_{R_{m'}}]}{\partial w_{m'}} = 0$$

$$\Rightarrow -2 \underline{y}^T_{m'} \underline{I} + 2 w_{m'} N_{R_{m'}} = 0$$

$$\Rightarrow w^*_{m'} = \frac{1}{N_{R_{m'}}} \underline{y}^T_{m'} \underline{1} = \frac{1}{N_{R_{m'}}} \sum_{x_i \in R_{m'}} y_i \quad \text{Q.E.D.}$$

(b) we can use bisection method to deal with it.

$t_K \leq \log_2 N$

2. *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 8 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:

At each iteration, 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 train a random forest with the "bag" set and 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.

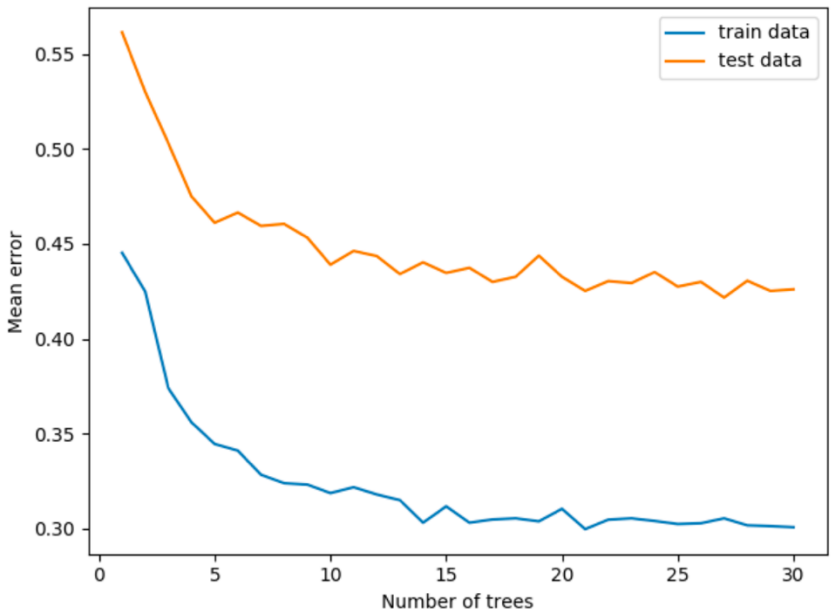
For this problem, please:

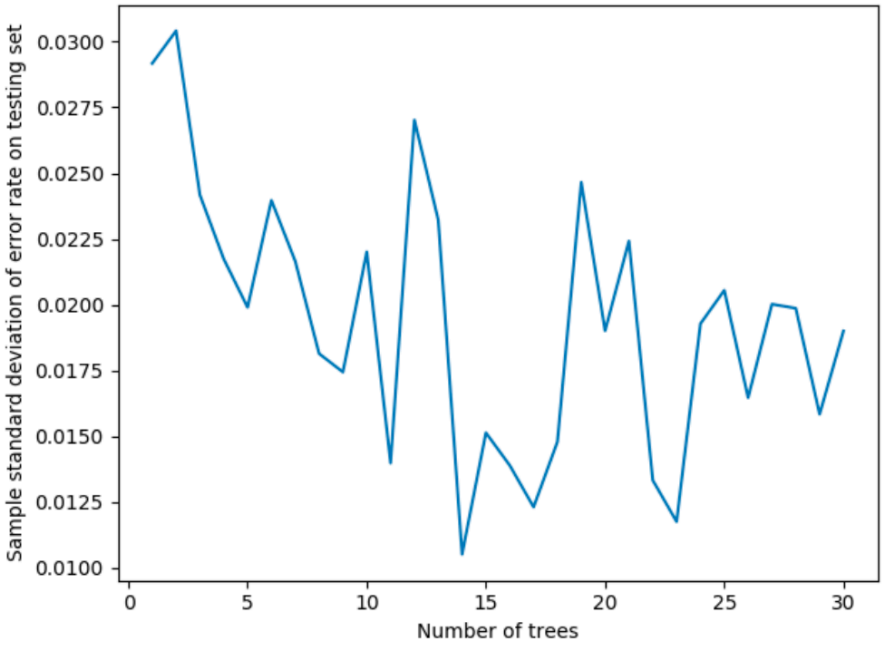
(a) State whether you used Matlab or Python; if you used different routines than given above, state what you used. And, if you needed to use parameter values different than specified above, state clearly what you used and why.

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

(a) I use Python and parameter values specified in this problem.

(b)





In the first figure, we can know that since we used the training dataset to train the model, the mean error of train data is, of course, lower than that of test data. In addition, we can see that as we increase the number of the trees, the mean error of train and test data both reduces.

In the second figure, the standard deviation of error rate on test data reaches its minimum when there are 14 trees. But there is no fixed tendency for the standard deviation curve to go up or down and the situation happened might be caused by the limited training data.

3. 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.

3. (a) semi-supervised learning problem
inductive, ∵ you need to get a decision boundary for your future image.

   (b) semi-supervised learning problem
inductive, ∵ we keep learning from unlabeled data to improve our classifier

   (c) Not a semi-supervised learning problem.

   (d) semi-supervised learning problem
transductive, ∵ you only need to label the rest of unlabeled pictures.