ISYE 6740 Homework 6
Fall 2021
Total 100 points

1. **Conceptual questions.** (20 points)

   (a) (5 points) Explain how do we control the data-fit complexity in regression tree.

   (b) (5 points) What's the main difference between boosting and bagging?

   (c) (5 points) Explain how OOB errors are constructed and how to use it to understand a good choice for the number of trees in random forest. Is OOB error test or training error and why?

   (d) (5 points) Explain what is "kernel trick" and why it is used?

2. **Random forest and one-class SVM for email spam classifier** (25 points)

   Your task for this question is to build a spam classifier using the UCR email spam dataset `https://archive.ics.uci.edu/ml/datasets/Spambase` came from the postmaster and individuals who had filed spam. Please download the data from that website. The collection of non-spam emails came from filed work and personal emails, and hence the word '`george`' and the area code '`650`' are indicators of non-spam. These are useful when constructing a personalized spam filter. You are free to choose any package for this homework. Note: there may be some missing values. You can just fill in zero.

   (a) (5 points) Build a CART model and visualize the fitted classification tree.

   (b) (10 points) Now also build a random forest model. Randomly shuffle the data and partition to use 80% for training and the remaining 20% for testing. Compare and report the test error for your classification tree and random forest models on testing data. Plot the curve of test error (total misclassification error rate) versus the number of trees for the random forest, and plot the test error for the CART model (which should be a constant with respect to the number of trees).

   (c) (10 points) Now we will use a one-class SVM approach for spam filtering. Randomly shuffle the data and partition to use 80% for training and the remaining 20% for testing.Extract all *non-spam* emails from the training block (80% of data you have selected) to build the one-class kernel SVM using RBF kernel (you can turn the kernel bandwidth to achieve good performance). Then apply it on the 20% of data reserved for testing (thus this is a novelty detection situation), and report the total misclassification error rate on these testing data.

3. **Locally weighted linear regression and bias-variance tradeoff.** (35 points)

Consider a dataset with n data points (x_i, y_i), $x_i \in \mathbb{R}^p$, following the following linear model

$$y_i = {\beta^*}^T x_i + \epsilon_i, \quad i = 1, \ldots, n,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent (but not identically distributed) Gaussian noise with zero mean and variance σ_i^2.

(a) (5 points) Show that the ridge regression which introduces a squared ℓ_2 norm penalty on the parameter in the maximum likelihood estimate of β can be written as follows

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left\{ (X\beta - y)^T W (X\beta - y) + \lambda \|\beta\|_2^2 \right\}$$

for property defined diagonal matrix W, matrix X and vector y.

(b) (5 points) Find the close-form solution for $\widehat{\beta}(\lambda)$ and its distribution conditioning on $\{x_i\}$.

(c) (5 points) Derive the bias as a function of λ and some fixed test point x.

(d) (5 points) Derive the variance term as a function of λ.

(e) (10 points) Now assuming the data are one-dimensional, the training dataset consists of two samples $x_1 = 1.5$ and $x_2 = 1$, and the test sample $x = 0.5$. The true parameter $\beta_0^* = 1$, $\beta_1^* = 1$, the noise variance is given by $\sigma_1^2 = 2$, $\sigma_2^2 = 1$. Plot the MSE (Bias square plus variance) as a function of the regularization parameter λ.

(f) (5 points) Now change the test sample to be a $x = 2$, and keep everything else to be same as in the previous question. Plot the MSE (Bias square plus variance) as a function of the regularization parameter λ, and comment on the difference from the previous result.

4. **Medical imaging reconstruction.** (20 points)

In this problem, you will consider an example resembles medical imaging reconstruction in MRI. We begin with a true image image of dimension 50×50 (i.e., there are 2500 pixels in total). Data is cs.mat; you can plot it first. This image is truly sparse, in the sense that 2084 of its pixels have a value of 0, while 416 pixels have a value of 1. You can think of this image as a toy version of an MRI image that we are interested in collecting.

Because of the nature of the machine that collects the MRI image, it takes a long time to measure each pixel value individually, but it's faster to measure a linear combination of pixel values. We measure $n = 1300$ linear combinations, with the weights in the linear combination being random, in fact, independently distributed as $\mathcal{N}(0, 1)$. Because the machine is not perfect, we don't get to observe this directly, but we observe a noisy version. These measurements are given by the entries of the vector

$$y = Ax + n,$$

where $y \in \mathbb{R}^{1300}$, $A \in \mathbb{R}^{1300 \times 2500}$, and $n \sim \mathcal{N}(0, 25 \times I_{1300})$ where I_n denotes the identity matrix of size $n \times n$. In this homework, you can generate the data y using this model.

Now the question is: can we model y as a linear combination of the columns of x to recover some coefficient vector that is close to the image? Roughly speaking, the answer is yes.

Key points here: although the number of measurements $n = 1300$ is smaller than the dimension $p = 2500$, the true image is sparse. Thus we can recover the sparse image using few measurements exploiting its structure. This is the idea behind the field of *compressed sensing*.

The image recovery can be done using lasso

$$\min_x \|y - Ax\|_2^2 + \lambda \|x\|_1.$$

(a) (10 points) Now use lasso to recover the image and select λ using 10-fold cross-validation. Plot the cross-validation error curves, and show the recovered image.

(b) (10 points) To compare, also use ridge regression to recover the image:

$$\min_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2.$$

Select λ using 10-fold cross-validation. Plot the cross-validation error curves, and show the recovered image. Which approaches give a better recovered image?