STAT 542, Homework 4

November 3, 2017
Due date: Nov 17 (Fri), 11:59 pm to Compass

Requirements: You should submit your report and R code(s), preferably in separate files. Your report should be in PDF/MS Word format. Font size should be 12pt and plots need to be clearly labeled. Your report should include necessary explanations and should not be a simple output file of the R code. The R code should include comments to help our grading process. There is a 15 page limit to the report. This homework worth 100 points total. Late submission penalty is 5 points for each day (round up) of delay.

Question 1: [30 points] Write your own code to fit a Nadaraya-Watson kernel regression estimator for a one dimensional problem. Let's consider just the Gaussian kernel function, and your function should include a tuning parameter for the bandwidth. Perform the following:

a) [15 points] Just like what we did in previous homework, you should generate a toy data example and validate that your code is correct.

b) [15 points] Download the "Video Game Sales with Ratings" dataset from Kaggle https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings. Perform your kernel estimator to estimate the `log(1+Global_Sales)` using each of the three scores: `Critic_Score`, `Critic_Count` and `User_Score`. Ignore observations with any missing value (ignore text value too). You should try to tune the bandwidth using cross-validation. Which score gives the best model? Demonstrate your results and make a conclusion.

Question 2: [35 points] Recall Question 2 in HW1, we did a simulation study to estimate the degrees of freedom of kNN. Lets do a similar experiment for random forests. The purpose is to understand the effect of different tuning parameters of random forests. Generate X from independent standard normal distribution with $n = 200$ and $p = 20$. For the true model, use $f(X) = 1 + 0.5\sum_{j=1}^{4} X_j$ and standard normal errors. Use the `randomForest` package for this question.

a) [20 points] Use the default values of `ntree`, tune different values of `mtry` and `nodesize` (make your own choice). Estimate and compare the degrees of freedom for these models. Summarize your results and comment on the effect of these two parameters.

b) [15 points] Fix `mtry` and `nodesize` as the default value, and tune `ntree`. Calculate (estimate) the variance of this estimator, i.e. $\frac{1}{n}\sum_i E_{\widehat{f}}\big(\widehat{f}(x_i) - E[\widehat{f}(x_i)]\big)^2$. Summarize your results and comment on the effect of `ntree`.

Question 3: [35 points] Lets write our own code for a one-dimensional adaboost using a stump model as the weak learner.

a) [15 points] The stump model is a CART model with just one split, hence two terminal nodes. Since we consider only one dimensional predictor, the only thing that needs to be searched in this tree model is the cutting point. Write a function to fit the stump model with subject weights:

Algorithm 1 A stump model, with weights

Input: A set of data $\{x_i, y_i, w_i\}_{i=1}^n$

1. Search for a splitting rule $I(x \le c)$ that will maximize the weighted reduction of Gini impurity.

   $$\texttt{score} = -\frac{\sum_{\mathcal{T}_L} w_i}{\sum w_i}\text{Gini}(\mathcal{T}_L) - \frac{\sum_{\mathcal{T}_R} w_i}{\sum w_i}\text{Gini}(\mathcal{T}_R)$$

   where, for given data in a node $\mathcal{T}$, the weighted version of Gini is

   $$\text{Gini}(\mathcal{T}) = \widehat{p}(1-\widehat{p}),\ \widehat{p} = (\sum w_i)^{-1}\sum w_i I(y_i = 1)$$

2. Calculate the left and the right node weighted predictions f_L, f_R $\in \{-1, 1\}$ respectively.

Output: The cutting point c, and node predictions f_L, f_R.

---

a) [20 points] Following the lecture slides page 6 in "Boosting.pdf", write your own code to fit the adaboost model using the stump as the base learner. For this implementation, you are not required to do bootstrapping for each tree (you still can if you want); You should generate the following data to test your code and demonstrate that it is correct. Also generate an independent set of testing data using this model to see if there is any potential overfitting. Comment on your findings.

```
n = 300
x = runif(n)
y = (rbinom(n, 1, (sin(4*pi*x)+1)/2) -0.5)*2
```