Assignment 1

Saturday, November 21, 2020 2:37 PM

Due: Friday, November 27 by 11:59pm Pacific (late assignments not accepted, solutions to be posted shortly thereafter for study purposes)

Instructions: Upload a well organized pdf/html document outputted from RMarkdown (or equivalent) to Canvas.

1. On Canvas in the assignment area, there is a data set named `salaries.csv` that contains data on professors and their salaries. As there is discrepency in R versions nowadays, when you use 'read.csv' (or an equivalent) please ensure that you set 'stringsAsFactors=TRUE', or otherwise coerce your character string predictors to factors after the fact. I learned the hard way, in another course, that 'tree' will skip character string predictors!

   (a) Create a regression tree for a professor's salary given the remainder of the variables in the data set. Provide the tree, including labels — using the command `text(treename, pretty=0)` will provide more understandable split labelling for the questions that follow.

   (b) The nodes that eventually result from the split "`yrs.since.phd < 46.5`" are a fairly strange result. Explain why this result is strange, then give one reasonable explanation why this has arisen. Note, it is not a mistake, you can browse the original data to see numerous examples of this phenomenon.

   (c) Based on this tree, as an early-ish career professor (Assistant or Associate) would you make more money in an 'applied' or 'theoretical' department? Explain.

   (d) Using `set.seed(6421)`, perform 20-fold cross-validation using `cv.tree`. Plot the resulting object. How many terminal nodes does cross-validation suggest?

   (e) Prune your original tree. Give the predicted salary for me, assuming I was at this university. That is, what is the predicted salary for an Assistant Professor, in an applied department (arguably, I suppose), who got their PhD in 2012 (8 years ago), has 5 years of service (usually counted as total amount of time as a professor at that particular university), and identifies as male. Use the `predict()` function to do this, but PLEASE double-check with your tree diagram and brain. My warning is to pay careful attention to how the character vectors are factored, and note that you will have to setup my entry as a 'data.frame'. You will likely find this finicky...but it is good practice for real life data science messiness.

   (f) Use the following commands to setup a training and testing set:

```
set.seed(763)
trainindex <- sample(1:nrow(Salaries), 200)
proftrain <- Salaries[trainindex, ]
proftest <- Salaries[-trainindex, ]
```

   Now fit a model to the training set, prune via 20-fold CV, and once again give the predicted salary for me via the predict function. Also provide the estimated MSE of the model — that is, calculate the MSE of the test set.

   (g) Is the MSE of the test set close to the expected MSE from the 20-fold CV from question (d)?

2. Fitting k-nearest neighbours ($k = 3$) on a data set resulted in the following classification matrices. On the left are results from fitting on the full data set, on the right is using cross-validation.

k-nearest neighbours

	Full		CV	
	Classified 1	Classified 2	Classified 1	Classified 2
1	50	6	44	12
2	4	42	6	40

Fitting a classification tree on the same data set resulted in the following classification matrices. Again, on the left are results from fitting on the full data set, on the right is using cross-validation.

Classification Tree

	Full		CV	
	Classified 1	Classified 2	Classified 1	Classified 2
1	54	2	38	18
2	1	45	6	40

Which method would we describe as the better method on this data, and why?

3. Many introductory statistics classes cover confidence intervals based on estimators such as the sample mean ($\bar{X}$) and sample variance S^2 under specific conditions (usually cases where X is normally distributed or the sample size n is sufficiently large for the Central Limit Theorem to hold). Specifically in the case of $s^2 = \frac{\sum(x_i-\bar{x})^2}{n-1}$ with normally distributed X, the confidence interval for variance is computed as:

$$\left(\frac{(n-1)s^2}{\chi^2_{\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{1-\alpha/2}} \right)$$

where $\chi^2_{\alpha/2}$ and $\chi^2_{1-\alpha/2}$ are found from the chi-squared distribution with $n-1$ degrees of freedom.

So what happens if we mistakenly use this formula when X is non-normal? Let's walk our way through some questions that will eventually (hopefully) enlighten us.

(a) Use set.seed(1111), and then generate a sample of size 30 from a standard continuous uniform distribution (bounds 0 and 1...see function "runif"). Store it in "x". Print the sample mean and sample variance of "x".

(b) Calculate the (leave one out) jackknife bias and jackknife standard error of the sample variance for this sample. Use the jackknife function from the bootstrap library.

(c) Use set.seed(2222), and then using $B = 1000$ bootstrap samples, obtain a bootstrap estimate of the bias and standard error of the sample variance. Use the bootstrap function within the bootstrap library.

(d) Obtain a 95% bootstrap CI for the population variance from the sample variance, using the quantile function shown in lecture.

(e) Obtain an improper 95% CI using the formula that assumes X is normally distributed (in the question write-up). Note that the chi-squared values can be found as follows: $\chi^2_{\alpha/2}$ = `qchisq(0.975, 30-1)`, $\chi^2_{1-\alpha/2}$ = `qchisq(0.025, 30-1)`

(f) In general, compare the two CIs you have found.

(g) The true variance is $\frac{1}{12}$. Is that contained in both of the above intervals?

(h) Of course, one run isn't particularly enlightening. Here I've expanded to 1000 simulations, finding 1000 bootstrap CIs and 1000 improper CIs.

```
> set.seed(3333)
> norm_var_ci <- boot_var_ci <- matrix(NA, nrow=1000, ncol=2)
> for(i in 1:1000){
+   dumx <- runif(30)
+   norm_var_ci[i, 1] <- (30-1)*var(dumx)/qchisq(0.975, 30-1)
+   norm_var_ci[i, 2] <- (30-1)*var(dumx)/qchisq(0.025, 30-1)
+   dumboot <- bootstrap(dumx, 1000, var)
+   boot_var_ci[i, 1] <- quantile(dumboot$thetastar, 0.025)
+   boot_var_ci[i, 2] <- quantile(dumboot$thetastar, 0.975)
+ }
> contain_var_norm <- contain_var_boot <- rep(NA, 1000)
> for(i in 1:1000){
+   contain_var_norm[i] <- norm_var_ci[i, 1] <= 1/12 & 1/12 <= norm_var_ci[i, 2
+   contain_var_boot[i] <- boot_var_ci[i, 1] <= 1/12 & 1/12 <= boot_var_ci[i, 2
+ }
> sum(contain_var_norm)
[1] 998
> sum(contain_var_boot)
[1] 941
```

Explain what the two resulting numbers printed out represent. What is the observed confidence level of the improper 95% interval? Of the bootstrap 95% interval? If the value which corresponds to the improper CI, is larger, does that mean that it's a better way to calculate CIs in the case of uniformly distributed X? Explain.