Module 5: Recommended Exercises

TMA4268 Statistical Learning V2024

Sara Martino, Stefanie Muff, Kenneth Aase, Daesoo Lee
Department of Mathematical Sciences, NTNU

February 8, 2024

We strongly recommend you to work through the Section 5.3 in the course book (Lab: Cross-Validation and the Bootstrap)

Problem 1

Explain how k-fold cross-validation is implemented.

a) Draw a figure.

b) Specify algorithmically what is done, and in particular how the "results" from each fold are aggregated.

c) Relate to one example from regression. Ideas are the complexity w.r.t. polynomials of increasing degree in multiple linear regression, or K in KNN-regression.

d) Relate to one example from classification. Ideas are the complexity w.r.t. polynomials of increasing degree in logistic regression, or K in KNN-classification.

Hint: the words "loss function," "fold," "training," and "validation" are central.

Problem 2

What are the advantages and disadvantages of k-fold cross-validation relative to

a) The validation set approach
b) Leave one out cross-validation (LOOCV)
c) What are recommended values for k, and why?

Hint: the words "bias," "variance" and "computational complexity" should be included.

Problem 3

Topic: Selection bias and the "wrong way to do CV."

The task here is to devise an algorithm to "prove" that the wrong way is wrong and that the right way is right.

a) What are the steps of such an algorithm? Write down a suggestion. Hint: How do you generate data for predictors and class labels, how do you do the classification task, where is the CV in the correct way and wrong way inserted into your algorithm? Can you make a schematic drawing of the right and the wrong way?

b) We are now doing a simulation to illustrate the selection bias problem in CV, when it is applied the wrong way. Here is what we are (conceptually) going to do:

Generate data

- Simulate high dimensional data ($p = 5000$ predictors) from independent or correlated normal variables, but with few samples ($n = 100$).
- Randomly assign class labels (here only 2). This means that the "truth" is that the misclassification rate can not get very small. What is the expected misclassification rate (for this random set)?

Classification task:

- We choose a few (for example $d = 10$) of the predictors (those with the highest correlation to the outcome).
- Perform a classification rule (here: logistic empirical Bayes) on these predictors.
- Then we run CV ($k = 5$) on either only the d (the wrong way), or on all $c + d$ predictors (the right way).
- Report misclassification errors for both situations.

One possible version of this is presented in the R-code below. Go through the code and explain what is done in each step, then run the code and observe if the results are in agreement with what you expected. Make changes to the R-code if you want to test out different strategies.

We start by generating data for $n = 50$ observations

```
library(boot)
# GENERATE DATA; use a seed for reproducibility
set.seed(4268)
n = 100  #number of observations
p = 5000  #number of predictors
d = 10  #top correlated predictors chosen

# Generating predictor data
xs = matrix(rnorm(n * p, 0, 1), ncol = p, nrow = n)  #simple way to to uncorrelated predictors
dim(xs)  # n times p
xs[1:10, 1:10]

# Generate class labels independent of predictors - so if all
# classifies as class 1 we expect 50% errors in general
ys = c(rep(0, n/2), rep(1, n/2))  #now really 50% of each
table(ys)
```

WRONG CV: Select the 25 most correlated predictors outside the CV.

```
corrs = apply(xs, 2, cor, y = ys)
hist(corrs)


selected = order(corrs^2, decreasing = TRUE)[1:d]

data = data.frame(ys, xs[, selected])
```

Then run CV around the fitting of the classifier - use logistic regression and built in `cv.glm()` function

```
logfit <- glm(ys ~ ., family = "binomial", data = data)
cost <- function(r, pi = 0) mean(abs(r - pi) > 0.5)
kfold <- 10
cvres <- cv.glm(data = data, cost = cost, glmfit = logfit, K = kfold)
cvres$delta
```

Observe a misclassification rate of about 20%.

CORRECT CV: Do not pre-select predictors outside the CV, but as part of the CV. In order words, the entire analysis is done within the CV. We need to code this ourselves:

```
reorder <- sample(1:n, replace = FALSE)
validclass <- NULL
for (i in 1:kfold) {
    neach <- n/kfold
    trainids <- setdiff(1:n, (((i - 1) * neach + 1):(i * neach)))
    traindata <- data.frame(xs[reorder[trainids], ], ys[reorder[trainids]])
    validdata <- data.frame(xs[reorder[-trainids], ], ys[reorder[-trainids]])
    colnames(traindata) <- colnames(validdata) <- c(paste("X", 1:p),
        "y")
    foldcorrs <- apply(traindata[, 1:p], 2, cor, y = traindata[, p +
        1])
    selected <- order(foldcorrs^2, decreasing = TRUE)[1:d]  # Select top d correlated
    data <- traindata[, c(selected, p + 1)]
    trainlogfit <- glm(y ~ ., family = "binomial", data = data)
    pred <- plogis(predict.glm(trainlogfit, newdata = validdata[, selected]))
    validclass <- c(validclass, ifelse(pred > 0.5, 1, 0))
}
table(ys[reorder], validclass)
1 - sum(diag(table(ys[reorder], validclass)))/n
```

Problem 4

We will calculate the probability that a given observation in our original sample is part of a bootstrap sample. This is useful for us to know in Module 8.

Our sample size is n.

a. We draw one observation from our sample. What is the probability of drawing observation i (i.e., x_i)? And of not drawing observation i?
b. We make n independent draws (with replacement). What is the probability of not drawing observation i in any of the n drawings? What is then the probability that data point i is in our bootstrap sample (that is, more than 0 times)?
c. When n is large $(1 - \frac{1}{n})^n \approx \frac{1}{e}$. Use this to give a numerical value for the probability that a specific observation i is in our bootstrap sample.
d. Write a short R code chunk to check your result. (Hint: An example on how to this is on page 198 in our ISLR book.) You may also study the result in c. How good is the approximation as a function of n?

Problem 5

Explain with words and an algorithm how you would proceed to use bootstrapping to estimate the standard deviation and the 95% confidence interval of one of the regression parameters in multiple linear regression. Comment on which assumptions you make for your regression model.

Problem 6

Implement your algorithm from 5 both using for-loop and using the `boot` function. Hint: See section 5.3.4 (the Lab on the bootstrap) in our ISLR book. Use our SLID data set from the `car` R package and provide standard errors for the coefficient for age. Compare with the theoretical value $(\mathbf{X}^\top\mathbf{X})^{-1}\hat{\sigma}^2$ that you find in the output from the regression model.

```r
library(car)
library(boot)
SLID <- na.omit(SLID)
n <- dim(SLID)[1]
SLID.lm <- lm(wages ~ ., data = SLID)
summary(SLID.lm)$coeff["age", ]
```

Now go ahead and use bootstrap to estimate the 95% CI. Compare your result to

```r
confint(SLID.lm)
```