The truth is out there R readers, but often it is not what we have been led to believe. The previous post examined the strong positive results bias in optimism corrected bootstrapping (a method of assessing a machine learning model’s predictive power) with increasing p (completely random features). There were 2 implementations of the method given, 1 has a slight error, 2 seems fine. The trend is still the same with the corrected code, but the problem with my code is I did not set ‘replace=TRUE’ in the call to the ‘sample’ function.

Thanks to ECOQUANT for pointing out the error.

Let’s just recap **what bootstrapping is** and **what optimism corrected bootstrapping is** before we redo the experiments:

This is from Jason’s excellent blog (<https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/>), bootstrapping is:

1. Choose a number of bootstrap samples to perform
2. Choose a sample size
3. For each bootstrap sample (b=1 … B)
   1. Draw a sample with replacement with the chosen size
   2. Calculate the statistic on the sample
4. Calculate the mean of the calculated sample statistics.

The with replacement part means we have to put each individual sample back when getting our sample in the bth bootstrap iteration. Thus, we usually have duplicate samples in our sample of the data when doing bootstrapping.

This is the optimism corrected bootstrapping algorithm:

1. Fit a model M to entire data S and estimate predictive ability C.
2. Iterate from b=1…B:
   1. Take a resample from the original data, S\*
   2. Fit the bootstrap model M\* to S\* and get predictive ability, C\_boot
   3. Use the bootstrap model M\* to get predictive ability on S, C\_orig
3. Optimism O is calculated as mean(C\_boot – C\_orig)
4. Calculate optimism corrected performance as C-O.

Since we use the same data in step 3 of the bootstrap to train and test the model (an information leak), we would expect increasing bias (C\_orig should be too high, thus O too small) when more and more random features are added. See the [previous post](https://intobioinformatics.wordpress.com/2018/12/28/part-4-more-bias-and-why-does-bias-occur-in-optimism-corrected-bootstrapping/) for more explanation on this. Another point is, the optimism corrected bootstrap is done with a sample size of N instead of just a fraction of N, usually. I found the following quote to support this:

“The small data set was repeatedly re-sampled to produce b replicated data sets, each the same size as the original. We used b = 200. The predictive model was fitted to each of  
the b replicated data sets in turn. Each fitted model was then applied both to the resampled data set from which it was generated and to the original data set.”

Smith, Gordon CS, et al. “Correcting for optimistic prediction in small data sets.” *American journal of epidemiology* 180.3 (2014): 318-324.

I have tried reducing the re-sampling size, which reduces the bias somewhat, but it is still there. This makes sense due to the information leak in this method which results in an under estimation of the optimism (O).

Your welcome to experiment with this code yourselves.

This code can be directly copied and pasted into R to repeat the experiments.

**Experiment 1: my implementation – glmnet (lasso logistic regression)**

library(glmnet)

library(pROC)

library(caret)

library(ggplot2)

library(kimisc)

### TEST 1: bootstrap optimism with glmnet

cc <- c()

for (zz in seq(2,100,1)){

print(zz)

## set up test data

test <- matrix(rnorm(100\*zz, mean = 0, sd = 1),

nrow = 100, ncol = zz, byrow = TRUE)

labelsa <- as.factor(c(rep('A',50),rep('B',50)))

colnames(test) <- paste('Y',seq(1,zz),sep='')

row.names(test) <- paste('Z',seq(1,100),sep='')

### my own implementation of optimism corrected bootstrapping

## 1. fit model to entire test data (i.e. overfit it)

orig <- glmnet(test,y=labelsa,alpha=1,family = "binomial")

preds <- predict(orig,newx=test,type='response',s=0.01)

auc <- roc(labelsa,as.vector(preds))

original\_auc <- as.numeric(auc$auc)

## 2. take resample of data and try to estimate optimism

test2 <- cbind(test,labelsa)

B <- 50

results <- matrix(ncol=2,nrow=B)

for (b in seq(1,B)){

## get the bootstrapped resampled data

boot <- test2[sample(row.names(test2),100,replace=TRUE),]

labelsb <- boot[,ncol(boot)]

boot <- boot[,-ncol(boot)]

## use the bootstrapped model to predict its own labels

bootf <- glmnet(boot,y=labelsb,alpha=1,family = "binomial")

preds <- predict(bootf,newx=boot,type='response',s=0.01)

auc <- roc(labelsb,as.vector(preds))

boot\_auc <- as.numeric(auc$auc)

## use bootstrap model to predict original labels

preds <- predict(bootf,newx=test,type='response',s=0.01)

auc <- roc(labelsa,as.vector(preds))

boot\_original\_auc <- as.numeric(auc$auc)

## add the data to results

results[b,1] <- boot\_auc

results[b,2] <- boot\_original\_auc

}

## calculate the optimism

O <- mean(results[,1]-results[,2])

## calculate optimism corrected measure of prediction (AUC)

corrected <- original\_auc-O

##

cc <- c(cc,corrected)

print(cc)

}

## print it

df <- data.frame(p=seq(2,100,1),optimism\_corrected\_boot\_AUC=cc)

p1 <- ggplot(df, aes(x=p, y=optimism\_corrected\_boot\_AUC)) +

geom\_line() + ggtitle('glmnet - random data only gives positive result with optimism corrected bootstrap')

print(p1)

png('glmnet\_test\_upto100.png', height = 15, width = 27, units = 'cm',

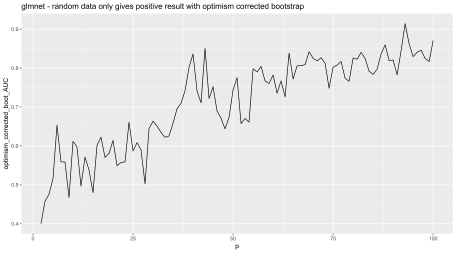
res = 900, type = 'cairo')

print(p1)

dev.off()

Here are the results with 100 samples and 50 bootstrap iterations from 2 to 100 random features from a Gaussian distribution. We are re-sampling using the original sample size (N=100).

Random features are being added iteratively on the X axis, and on the Y, we have AUC. The AUC should be 0.5 to reflect the data has no real predictive power, but it is highly inflated.



**Experiment 2: another implementation – glm (logistic regression)**

## TEST2

auc.adjust <- function(data, fit, B){

fit.model <- fit

data$pred.prob <- fitted(fit.model)

# get overfitted AUC

auc.app <- roc(data[,1], data$pred.prob, data=data)$auc # require 'pROC'

auc.boot <- vector (mode = "numeric", length = B)

auc.orig <- vector (mode = "numeric", length = B)

o <- vector (mode = "numeric", length = B)

for(i in 1:B){

boot.sample <- sample.rows(data, nrow(data), replace=TRUE) # require 'kimisc'

fit.boot <- glm(formula(fit.model), data = boot.sample, family = "binomial")

boot.sample$pred.prob <- fitted(fit.boot)

# get bootstrapped AUC

auc.boot[i] <- roc(boot.sample[,1], boot.sample$pred.prob, data=boot.sample)$auc

# get original data boot AUC

data$pred.prob.back <- predict.glm(fit.boot, newdata=data, type="response")

auc.orig[i] <- roc(data[,1], data$pred.prob.back, data=data)$auc

# calculated optimism corrected version

o[i] <- auc.boot[i] - auc.orig[i]

}

auc.adj <- auc.app - (sum(o)/B)

return(auc.adj)

}

cc <- c()

for (zz in seq(2,100,1)){

print(zz)

## set up test data

test <- matrix(rnorm(100\*zz, mean = 0, sd = 1),

nrow = 100, ncol = zz, byrow = TRUE)

labelsa <- as.factor(c(rep('A',50),rep('B',50)))

colnames(test) <- paste('Y',seq(1,zz),sep='')

row.names(test) <- paste('Z',seq(1,100),sep='')

test2 <- data.frame(cbind(labelsa,test))

test2$labelsa <- as.factor(test2$labelsa)

## 1. make overfitted model

model <- glm(labelsa ~., data = test2, family = "binomial")

## 2. estimate optimism and correct

d <- auc.adjust(test2, model, B=200)

cc <- c(cc,d)

}

## print it

df <- data.frame(p=seq(2,100,1),optimism\_corrected\_boot\_AUC=cc)

p2 <- ggplot(df, aes(x=p, y=optimism\_corrected\_boot\_AUC)) +

geom\_line() + ggtitle('glm - random data only gives positive result with optimism corrected bootstrap')

print(p2)

png('glmt\_test\_upto100.png', height = 15, width = 27, units = 'cm',

res = 900, type = 'cairo')

print(p2)

dev.off()

