Machine Learning Model

Andrew J. Otis

May 22, 2022

Overview

The first problem approaches the selection of a logistic regression model for the "SAheart.data" using cross-validation on the merged "dat.train" and "dat.valid".

The second problem compares implications of two models for "chd" in "SAheart.data" for individuals.

Data

Read in the data and break it into training, validation, and test sets. Preserve the breakdown used in the analysis of the appropriateness of the train-validate-test protocol for this analysis.

```
dat <- read.csv("SAheart.data")
# The "dplyr::"" specifies that the dplyr "select" function be used
dat <- dplyr::select(dat, -row.names)
n <- nrow(dat)
dat$famhist <- (dat$famhist=="Present")*1 # better for bestglm

set.seed(123456)
tvt <- sample(rep(0:2,c(round(n*.2),round(n*.2),n-2*round(n*.2))),n)


dat.train <- dat[tvt==2,]
dat.valid <- dat[tvt==1,]
dat.test <- dat[tvt==0,]

dat.tv <- dat[tvt>0,]
```

Reconstruct the formula for the maximal model from the analyses in class. It includes all the explanatory variables and the squares of the numeric variables.

```
nam <- names(dat.tv)[1:(ncol(dat.tv)-1)]

# Assemble the names of the numeric variables. The variable
# "famhist" is the only non-numeric variable. Its
# square should be omitted.
nam.num<-setdiff(nam,"famhist")
```

```
# Use stringr to avoid typing all the explanatory variables.
fmla.sq <- as.formula(str_c("chd~",
    str_c(nam,collapse="+"),"+",
    str_c("I(",nam.num,"^2)",
              collapse="+")))
```

Fit a forward model on all the variables. (Recall that the tvt forward model was "chd ~ 1 + age + ldl + famhist + tobacco + I(typea^2) + I(age^2) + I(sbp^2)")

```
m<-glm(chd~1,data=dat.tv,family="binomial")

m.forward.tv <- step(m,scope=fmla.sq,direction = "forward" ,trace=0)

summary(m.forward.tv)
```

Select forward model with cross validation

The basic organization is to fit forward models on each fold, select the model size that has the lowest pooled cross-validated deviance, then fit the forward model of that size on the training data. We will be updating the logistic regression algorithm on the current data.

Question 1, part 1: Cross validation helper functions

A function that fits the sequence of forward models on a given data set and evaluates the deviance on another data set.

(In the interest of readability, these functions are specific to the current problem and would need to be modified for a new application.)

```
# create a formula for "chd" in terms of sequences of variables in "vars.add"
fmla.add.fnc <- function(i,vars.add){
vars.in <- vars.add[1:i]
return(as.formula(str_c("chd~",str_c(vars.in,collapse="+"))))
}

# function to calculate validation set deviance
deviance.valid <- function(m,dat.valid){
pred.m <- predict(m,dat.valid, type="response")
-2*sum(dat.valid$chd*log(pred.m)+(1-dat.valid$chd)*log(1-pred.m))
}
# Code to extract the variables added in order in a call to "step" with
# direction equal to "forward"
vars.get <- function(model.forward){
vars.add <- model.forward$anova[,1]
vars.add <- str_replace(vars.add,"\\+ ","")
vars.add[1] <- 1
return(vars.add)
}
```

```
# function to fit a sequence forward models with scope "fmla.this
# on the data set "dat.this" and
# return the deviance for each model on "dat.valid.this".
deviance.get<-function(dat.this,fmla.this,dat.valid.this){
m.forward <- step(glm(chd~1,data=dat.this,family="binomial"),scope=fmla.this,
k=0,direction="forward",trace=0)

# Collect the variables used in the order in which they were added
vars.add <- vars.get(m.forward)
# Apply "fmla.add.fnc" to each value of "i". This
# gives the formulas for the models generated initial sequences of the
# variables in vars.add
# Note that the first formula is for the model with just the intercept.
fmlas <- apply(matrix(1:length(vars.add),ncol=1),1,
fmla.add.fnc,vars.add=vars.add)

# Make a list of models corresponding to these formulas.
models<-
lapply(fmlas,function(x){glm(x,data=dat.this,family="binomial")})
# Calculate the deviance on "dat.valid" for each model.
return(sapply(models,deviance.valid,dat.valid=dat.valid.this))
}

# Please run this code to check that your function performs as required:
set.seed(12345678)
ind <- createFolds(factor(dat.tv$chd), k = 8, list = FALSE)
deviance.get(dat.tv[ind!=1,],fmla.sq,dat.tv[ind==1,])

# [1] 59.44126 64.55894 57.54275 55.49642 57.25787 57.78025 53.53396
#  [8] 54.08417 53.85130 51.99240 51.94703 51.34293 51.91824 51.49767
# [15] 52.02923 51.87757 52.03731 52.12310 when run by instructor
```

Question 1, part 2: Function for deviance on a fold

A wrapper function for deviance.get that splits training and validation based on the index "i" and calls "deviance.get" with the training data as the folds omitting the ith and the ith fold as the validation data.

```
deviance.wrapper <- function(i,dat.w,ind.w,fmla.w){
  return(deviance.get(dat.w[ind.w!=i,],fmla.w,
                      dat.w[ind.w==i,]))
}
# Please run this code to check that your function performs as required:
set.seed(12345678)
  ind <- createFolds(factor(dat.tv$chd), k = 8, list = FALSE)

deviance.wrapper(1,dat.tv,ind,fmla.sq)
# [1] 59.44126 53.32451 48.25465 47.12235 46.41322 45.04778 44.45387
#  [8] 44.28262 44.96931 43.92987 44.36738 43.88697 45.28247 45.78695
# [15] 45.39655 45.46604 45.49909 45.31704 when run by instructor
```

Question 1, part 3: Size selection by cross-validation

A function that creates a fold structure, applies the wrapper function to each fold, sums the corresponding model deviances, and returns the vector of sums. Please run the commands at the end of the function definition to check your work.

The data set "dat.tv", the merged training and validation sets from the train-validate-test work in class, is used because cross validation doesn't require a validation set.

```
deviance.wrapper <- function(i,dat.w,ind.w,fmla.w){
return(deviance.get(dat.w[ind.w!=i,],fmla.w,dat.w[ind.w==i,]))
}
# Please run this code to check that your function performs as required:
set.seed(12345678)
ind <- createFolds(factor(dat.tv$chd), k = 8, list = FALSE)
deviance.wrapper(1,dat.tv,ind,fmla.sq)


# Check work and calculate selected model size
set.seed(12345678)
fwd <- deviance.sums.xv(dat.tv,fmla.sq) # for debugging
fwd[1:4]
#[1] 477.2319 424.2526 406.2646 406.5074 when run by instructor
(model.size <- which.min(fwd))
plot(fwd)
```

Question 1, part 4: Compare deviances

Fit a forward model of the size "model.size"-1 for the value of "model.size" calculated above on "dat.tv".

The forward model selected in class using train-validate-test is also fit on "dat.tv". Recall that this model was selected by fitting a sequence of forward models on the training data. The deviances of each of the models on the validation data were computed. The model with the minimum deviance on the validation data was selected.

Please compute the deviances that result when "forward.model.xv" and "forward.model.tvt" are used to predict "chd" on the test data.

Note that in practice, model selection is finished before the performance on test data is checked. This examination of the performance of two models on the test data is for understanding the behavior of cross-validation and train-validate-test on these data, not for model selection.

```
forward.model.xv <- step(glm(chd~1,data=dat.tv,family="binomial"),scope=fmla.
sq, direction="forward",steps=model.size-1,k=0,trace=0)
```

```
summary(forward.model.xv)

(fmla.forward.best <- as.formula("chd ~ 1 + age + ldl + famhist + tobacco + I
(typea^2) + I(age^2) + I(sbp^2)"))

forward.model.tvt <- glm(fmla.forward.best,data=dat.tv,family="binomial")
summary(forward.model.tvt)
```

Question 1, part 5: test versus validation deviance

In train-validate-test, the step of assessing the performance of the selected model on the test data is used partly because the performance on the validation data may be better than the performance on fresh data since the selection was based on good performance on the validation data.

Is there evidence of that effect here for the model "chd ~ 1 + age + ldl + famhist + tobacco + I(typea^2) + I(age^2) + I(sbp^2)" selected from the forward sequence by train-validate-test?

```
forward.model.t <- glm(fmla.forward.best,data=dat.train,family="binomial")

deviance.valid(forward.model.t,dat.valid)
```

Cross validation serves the specialized purpose of only measuring the effect of the "step" (in our case a forward step) that are re-done during the cross validation process.

k-fold cross validation is a procedure used to estimate the skill of the model on new data.

The resulting value of 94.9971 is the accuracy of testing versus the validation evidence. In other words, the result is indicative of evidence that performance on the validation data is potentially better than performance on the fresh data.

Question 2

In studying the asymptotic distribution of coefficients for a logistic regression model, we developed a method for estimating ranges of the probability of the binary outcome for a case by sampling coefficients according to that distribution. The questions below relate to that method.

Question 2, part 1

The code below gives a function to return a range of "n" probabilities of a binary outcome for a case based "k" on the distribution of the coefficients in a logistic regression model "m".

```
probs.get <- function(m,k,n){
  cov.mat <- vcov(m)
  b <- mvrnorm(n,m$coefficients,cov.mat)
```

```
  # matrix with columns equal to collections of coefficients
  b <- t(b)
  mat<-model.matrix(m)
  obs <- t(mat[k,])

  # values of the linear predictor for case k
  eta <- obs%*%b

  # probabilities
  prob <- exp(eta)/(1+exp(eta))
  return(as.vector(prob))
}
```

Question 2, part 2

Consider two logistic regression models for "chd",

chd ~ age + famhist + ldl + I(typea^2) + tobacco

and

chd ~ 1 + age + ldl + famhist + tobacco + I(typea^2) + I(age^2) + I(sbp^2)

Examine the distribution of probabilities for "chd" estimated by these two models for a sampled individual when the coefficients of the logistic regressions are sampled from the Normal estimate of their distributions.

```
m1 <- glm(chd ~ age + famhist + ldl +I(typea^2)+tobacco, data=dat,family="bin
omial")

cov.mat1 <- vcov(m1)

m2 <- glm(chd ~ age + ldl + famhist + tobacco + I(typea^2) + I(age^2) + I(sbp
^2),data=dat,family="binomial")

cov.mat2 <- vcov(m2)

# Select an individual
set.seed(1234567)
indiv <- sample(nrow(dat),1)
# original estimate
predict(m1,type="response")[indiv]
predict(m2,type="response")[indiv]

# sample 1000 probabilities for "indiv" based on m1
p1 <- probs.get(m1,indiv,1000)

# sample 1000 probabilities for "indiv" based on m2
p2 <- probs.get(m2,indiv,1000)
```

```r
# Calculate the interquartile range of each
quantile(p1,.75)-quantile(p1,.25)
quantile(p2,.75)-quantile(p2,.25)


temp <- data.frame(model.num=c(rep(1,1000),rep(2,1000)),p=c(p1,p2))
ggplot(temp, aes(x=p))+geom_density(aes(color=as.factor(model.num)))
```