Stat 2025- HW 7

Michael Discenza

March 15, 2013

We know that we have a binary response variable, "chd," where 0 represents the absence and 1 the presence of coronary heart disease. Because the goal of this exercise is predicting the score of the 42 rows, we will try to minimize cv prediction error.

First we load the data and split it into training and test sets.

```
SAH <- read.table("http://stat.columbia.edu/~madigan/W2025/data/SAHmissing.txt", header = TRUE,
    sep = "\t")
SAH.training <- na.exclude(SAH)
SAH.test <- SAH[421:462, ]
```

Here we are not so focused on creating "predictive"good" models with low BICs or AICs, we are more focused on predictive power. We don't have to be so concerned with making small models that are as well vetted as long as they predict well. Nonetheless, we start by fitting a ful model and seeing which variables are statistically significant, or nearly so, and which variables are not particularly informative. We drop the non particularly informative variables (alcohol and adiposity) and use the rest in the the modeling processes.

```
m1 <- glm(chd ~ ., data = SAH.training)
summary(m1)

##
## Call:
## glm(formula = chd ~ ., data = SAH.training)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.743  -0.332  -0.106   0.379   1.040
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.71e-01   2.14e-01   -2.20  0.02828 *
## sbp             1.59e-03   1.10e-03    1.44  0.14953
## tobacco         1.71e-02   4.99e-03    3.44  0.00065 ***
## ldl             3.44e-02   1.12e-02    3.07  0.00227 **
## adiposity       1.78e-03   4.97e-03    0.36  0.72129
## famhistPresent  1.71e-01   4.34e-02    3.94  9.7e-05 ***
## typea           4.45e-03   2.17e-03    2.06  0.04046 *
## obesity        -1.02e-02   7.22e-03   -1.42  0.15756
## alcohol         8.04e-05   8.65e-04    0.09  0.92596
## age             6.48e-03   2.07e-03    3.13  0.00188 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 0.1769)
##
##     Null deviance: 94.629  on 419  degrees of freedom
## Residual deviance: 72.540  on 410  degrees of freedom
## AIC: 476.3
##
## Number of Fisher Scoring iterations: 2
```

For the rest of the process, we will fit a number of different models with different tupes of smoothing and on different variables then compare the AIC and cross validation mean squared error. Then we will choose the optimatmal model and predict the the binary response for the 42 observations in the test data.

```
## Error:  there is no package called 'AED'
```

```
attach(SAH)
library(mgcv)
library(AED)
library(mgcv)
library(boot)

costf <- function(r, pi = 0) mean((r - round(pi))^2)  # need to define a cost function that rounds the b
results1 <- c("Regular Logistic Regression", AIC(m1), cv.glm(data = SAH.training, glmfit = m1,
    cost = costf, K = 10)$delta[2])

# model with all predictors and all smoothed cr
m2 <- gam(chd ~ s(sbp, bs = "cr") + s(tobacco, bs = "cr") + s(ldl, bs = "cr") + s(adiposity,
    bs = "cr") + s(typea, bs = "cr") + s(obesity, bs = "cr") + famhist + s(alcohol, bs = "cr") +
    s(age, bs = "cr"), data = SAH.training, family = binomial)
results2 <- c("All predictors, cr smooth", AIC(m2), cv.glm(data = SAH.training, glmfit = m2,
    cost = costf, K = 10)$delta[2])

# model with only significant predictors from full logistic model cr
m3 <- gam(chd ~ s(sbp, bs = "cr") + s(tobacco, bs = "cr") + s(ldl, bs = "cr") + s(typea, bs = "cr") +
    s(obesity, bs = "cr") + famhist + s(age, bs = "cr"), data = SAH.training, family = binomial)
results3 <- c("Only significant predictors, cr smooth", AIC(m3), cv.glm(data = SAH.training,
    glmfit = m3, cost = costf, K = 10)$delta[2])

# model with all predictors cs
m4 <- gam(chd ~ s(sbp, bs = "cs") + s(tobacco, bs = "cs") + s(ldl, bs = "cs") + s(adiposity,
    bs = "cs") + s(typea, bs = "cs") + s(obesity, bs = "cs") + famhist + s(alcohol, bs = "cs") +
    s(age, bs = "cs"), data = SAH.training, family = binomial)
results4 <- c("All predictors, cs smooth", AIC(m4), cv.glm(data = SAH.training, glmfit = m4,
    cost = costf, K = 10)$delta[2])

# looking at the GAM check for the first model, se wee that a lot of smoothers could
# potentially be dropped because the relationship between the predictors and the
# dependent variable are pretty linear
m5 <- gam(chd ~ sbp + s(tobacco, bs = "cs") + ldl + typea + obesity + famhist + age, data = SAH.training
    family = binomial)
results5 <- c("Only significant predictors, cs smooth", AIC(m5), cv.glm(data = SAH.training,
    glmfit = m5, cost = costf, K = 10)$delta[2])

results <- rbind(results1, results2, results3, results4, results5)
```

```
colnames(results) <- c("Model", "AIC", "CV.MSE (k=10)")
results
# with cs and not cr
```

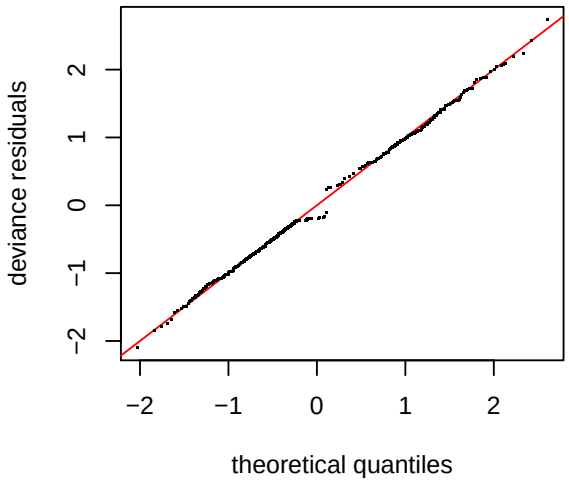
After fitting a number of models, we compare their cross validated mean squared errror and AIC.

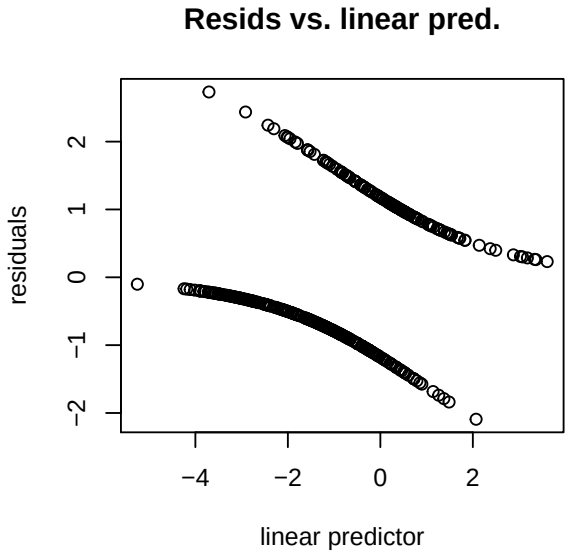
	Model	AIC	CV.MSE (k=10)
Regular Logistic Regression	476.33735549369	0.268095238095238	
All predictors, cr smooth	442.281672512403	0.261428571428571	
Only significant predictors, cr smooth	437.611549623085	0.28452380952381	
All predictors, cs smooth	438.502335034372	0.28952380952381	
Only significant predictors, cs smooth	439.3932621128	0.284761904761905	

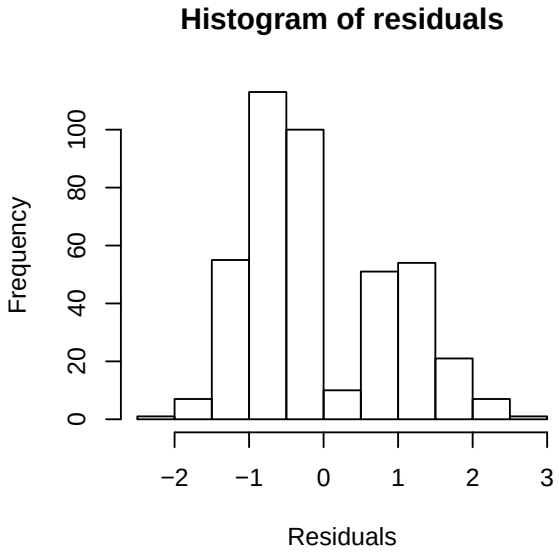
For the final model that we will use to make predictions, we choose, the second model, which includes all predictors smoothed with cubic regression splines. Before making the predictions we, examine the model and see that assumptions are met.

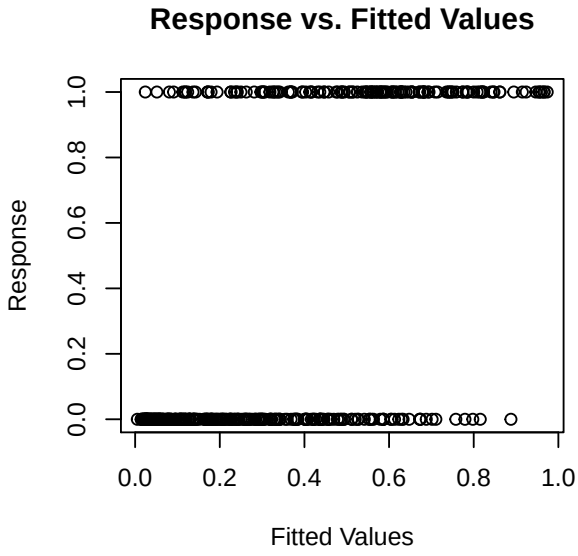
```
## Warning:  matrix not positive definite
## Warning:  matrix not positive definite
## Warning:  matrix not positive definite
## Warning:  matrix not positive definite
```

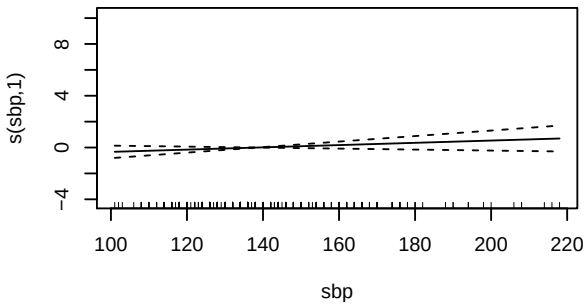
```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 12 iterations.
## Gradient range [-1.821e-07,1.686e-08]
## (score 0.05305 & scale 1).
## Hessian positive definite, eigenvalue range [2.892e-08,0.001906].
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                 k'   edf k-index p-value
## s(sbp)       9.000 1.000   0.979    0.38
## s(tobacco)   9.000 6.045   1.015    0.66
## s(ldl)       9.000 1.000   1.007    0.52
## s(adiposity) 9.000 3.421   1.006    0.56
## s(typea)     9.000 7.260   1.094    0.98
## s(obesity)   9.000 1.552   0.987    0.45
## s(alcohol)   9.000 1.000   1.028    0.79
## s(age)       9.000 1.000   1.052    0.88
```

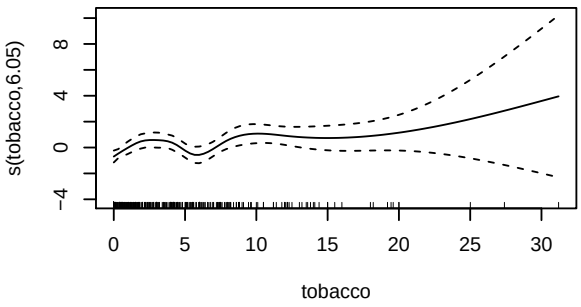

deviance residuals
theoretical quantiles
−2
−1
0
1
2

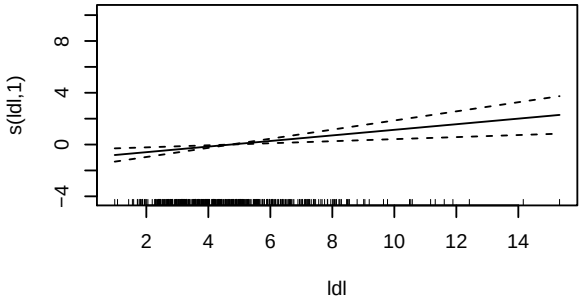

Resids vs. linear pred.
residuals
linear predictor
−4
−2
0
2

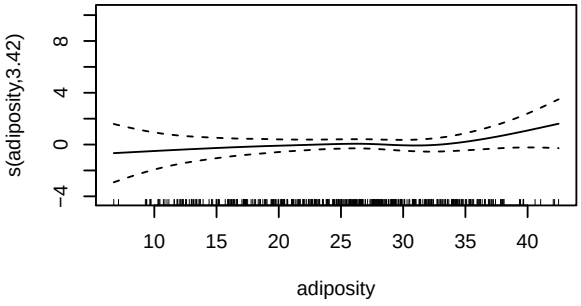

Histogram of residuals
Frequency
Residuals
0
20
40
60
80
100
−2
−1
0
1
2
3

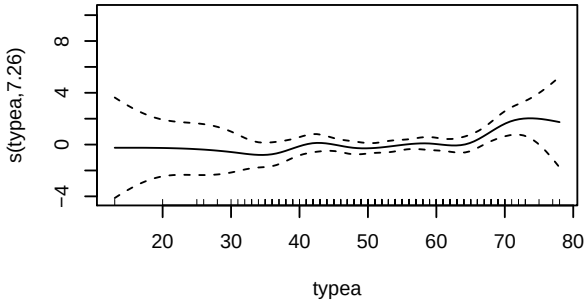

Response vs. Fitted Values
Response
Fitted Values
0.0
0.2
0.4
0.6
0.8
1.0

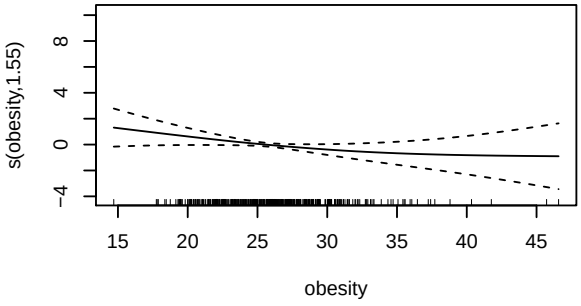

s(sbp,1)
−4
0
4
8
100
120
140
160
180
200
220
sbp

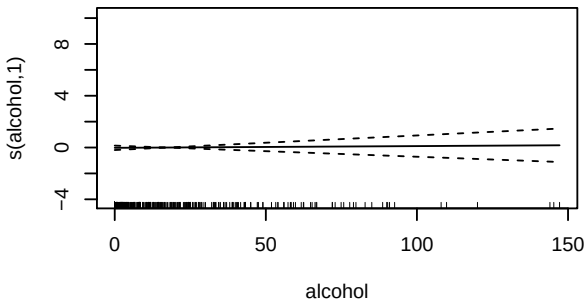

s(tobacco,6.05)
−4
0
4
8
0
5
10
15
20
25
30
tobacco

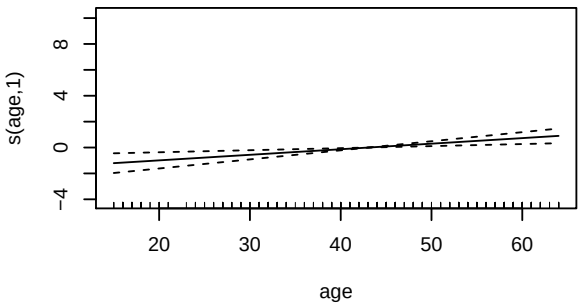

s(ldl,1)
−4
0
4
8
2
4
6
8
10
12
14
ldl


s(adiposity,3.42)
−4
0
4
8
10
15
20
25
30
35
40
adiposity


s(typea,7.26)
−4
0
4
8
20
30
40
50
60
70
80
typea


s(obesity,1.55)
−4
0
4
8
15
20
25
30
35
40
45
obesity





Looking at the residual, plot, we see that it is a little funky. The qq norm plot indicates that there are a few issues with residuls being normally distributed, but since the m2 model still provides the best CV MSE, we will use it.

Our last step is to apply the model to the test data and round the results to 1 or zero:

```
fitted_test_values <- predict.gam(object = m2, newdata = SAH.test, se = TRUE, type = "response")
round(fitted_test_values$fit[1:42])

## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442
##   1   1   0   0   0   1   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0
## 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462
##   0   0   0   1   0   1   0   1   1   1   0   0   0   0   0   0   0   0   0   1
```