STAT 8051 HW 5

Jingxiang Li

October 21, 2014

Problem 11.3

(Data file: `walleye`) The data in the file `walleye` give the `length` in mm and the `age` in years of a sample of over 3,000 male walleye, a popular game fish, captured in Butternut Lake in Northern Wisconsin (LeBeau, 2004). The fish are also classified according to the time `period` in which they were captured, with `period` $= 1$ for pre-1990, `period` $= 2$ for 19901996, and `period` $= 3$ for 19972000. Management practices on the lake were different in each of the periods, so it is of interest to compare the length at age for the three time periods.

Using the *von Bertalanffy* length at age function (11.22), compare the three time periods. If different, are all the parameters different, or just some of them? Which ones? Summarize your results.

Solution

the *von Bertalanffy* function is given by

$$E(\texttt{Length}|\texttt{Age} = t) = L_\infty(1 - \exp(-K(t - t_0)))$$

where L_∞, K and t_0 are unknown parameters.

To obtain the parameters within the *von Bertalanffy* function, we should first consider a proper way to set the initial value for estimation. Since parameter L_∞ is the expected value of `Length` for extremely large ages, we can set the initial value of L_∞ as the maximum of `Length` plus the standard error of `Length`. Then we can L_∞ as known parameter, so that the function become linear. Let's define $y = \log(1 - \frac{\texttt{Length}}{L_\infty})$ and run regression $y \sim 1 + \texttt{age}$. Therefore the estimated slope coefficient will be good initial value for $-K$, the ratio of the intercept over slope can be the initial value for t_0

Then we can estimate the parameters by using Gauss-Newton algorithm. Here we will build up 5 different models to see whether parameters are different for each time period. model `c1` will be the simplest model, which means all three parameters are the same for each time period. Then `c2` will be the most complex model, where all parameters are assumed as different for each time period. Then in `c3`, we assume that only L_∞ is the same; in `c4`, we assume that only K is the same; in `c5` we assume that only t_0 is the same, for each time period.

Lastly we will test the difference between each model by ANOVA.

```
rm(list = ls())
require(alr4)

data (walleye)

## Initialization
Linf <- max(walleye$length) + sd(walleye$length)
m0 <- lm(log(1 - length / Linf) ~ age, data = walleye)
K  <-  - coef(m0)[2]
```

```
t0 <- coef(m0)[1] / coef(m0)[2]

## c1 Simplest model, same K, same Linf and same t0
c1 <- nls(length ~ Linf * (1 - exp( - K * (age - t0))),
          start = list(Linf = Linf, K = K, t0 = t0),
          data = walleye)
## c2 Most complex model, different K, different Linf and different t0
c2 <- nls(length ~ (period == 1) * Linf1 * (1 - exp( - K1 * (age - t01))) +
                   (period == 2) * Linf2 * (1 - exp( - K2 * (age - t02))) +
                   (period == 3) * Linf3 * (1 - exp( - K3 * (age - t03))),
          start = list(Linf1 = Linf, Linf2 = Linf, Linf3 = Linf,
                       K1 = K, K2 = K, K3 = K,
                       t01 = t0, t02 = t0, t03 = t0),
          data = walleye)
## c3 same Linf
c3 <- nls(length ~ (period == 1) * Linf * (1 - exp( - K1 * (age - t01))) +
                   (period == 2) * Linf * (1 - exp( - K2 * (age - t02))) +
                   (period == 3) * Linf * (1 - exp( - K3 * (age - t03))),
          start = list(Linf = Linf,
                       K1 = K, K2 = K, K3 = K,
                       t01 = t0, t02 = t0, t03 = t0),
          data = walleye)
## c4 same K
c4 <- nls(length ~ (period == 1) * Linf1 * (1 - exp( - K * (age - t01))) +
                   (period == 2) * Linf2 * (1 - exp( - K * (age - t02))) +
                   (period == 3) * Linf3 * (1 - exp( - K * (age - t03))),
          start = list(Linf1 = Linf, Linf2 = Linf, Linf3 = Linf,
                       K = K,
                       t01 = t0, t02 = t0, t03 = t0),
          data = walleye)
## c5 same t0
c5 <- nls(length ~ (period == 1) * Linf1 * (1 - exp( - K1 * (age - t0))) +
                   (period == 2) * Linf2 * (1 - exp( - K2 * (age - t0))) +
                   (period == 3) * Linf3 * (1 - exp( - K3 * (age - t0))),
          start = list(Linf1 = Linf, Linf2 = Linf, Linf3 = Linf,
                       K1 = K, K2 = K, K3 = K,
                       t0 = t0),
          data = walleye)

## Anova
anova(c1, c3, c2)

## Analysis of Variance Table
##
## Model 1: length ~ Linf * (1 - exp(-K * (age - t0)))
```

```
## Model 2: length ~ (period == 1) * Linf * (1 - exp(-K1 * (age - t01))) + (period == 2)
## Model 3: length ~ (period == 1) * Linf1 * (1 - exp(-K1 * (age - t01))) + (period == 2
##   Res.Df Res.Sum Sq Df Sum Sq F value    Pr(>F)
## 1   3195    2211448
## 2   3191    1994577  4 216871  86.740 < 2.2e-16 ***
## 3   3189    1963513  2  31064  25.226  1.35e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(c1, c4, c2)
```

```
## Analysis of Variance Table
##
## Model 1: length ~ Linf * (1 - exp(-K * (age - t0)))
## Model 2: length ~ (period == 1) * Linf1 * (1 - exp(-K * (age - t01))) + (period == 2)
## Model 3: length ~ (period == 1) * Linf1 * (1 - exp(-K1 * (age - t01))) + (period == 2
##   Res.Df Res.Sum Sq Df Sum Sq F value    Pr(>F)
## 1   3195    2211448
## 2   3191    2014863  4 196585  77.834 < 2.2e-16 ***
## 3   3189    1963513  2  51350  41.700 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(c1, c5, c2)
```

```
## Analysis of Variance Table
##
## Model 1: length ~ Linf * (1 - exp(-K * (age - t0)))
## Model 2: length ~ (period == 1) * Linf1 * (1 - exp(-K1 * (age - t0))) + (period == 2)
## Model 3: length ~ (period == 1) * Linf1 * (1 - exp(-K1 * (age - t01))) + (period == 2
##   Res.Df Res.Sum Sq Df Sum Sq F value    Pr(>F)
## 1   3195    2211448
## 2   3191    1989989  4 221458  88.779 < 2.2e-16 ***
## 3   3189    1963513  2  26476  21.500 5.307e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see, three ANOVA tables above show that all the tests on difference are significant, suggesting that all three parameters should be different for each period of time, and we should accept the most complex model `c2`. However, since the sample size is too large, statistical significance may not be equivalent to practical significance.

Problem 2

Analyze the Boston Housing Data. The data set (named Boston) is available in the package MASS. The response is median value of owner-occupied homes (the last variable in the data set). You may restrict your attention to linear models. Please consider the following methods: full model, all subset selection based on AIC, all subset selection based on BIC, Lasso, and ridge regression.

Use half-half CV to compare their performances.

Solution

```
require(sampling)
require(MASS)
require(glmnet)
require(lars)
require(leaps)

rm(list = ls())

data(Boston)
n <- nrow(Boston)
p <- ncol(Boston)
X <- as.matrix(Boston[, 1 : (p - 1)])
y <- Boston$medv

set.seed(123123)
index <- as.logical(srswor(n / 2, n))
index_1 <- as.logical(1 - index)
index <- list(index, index_1)

error <- matrix(nrow = 5, ncol = 2)

## Half Half CV
for (i in 1 : 2)
{
    data_train <- Boston[index[[i]], ]
    data_test <- Boston[index[[3 - i]], ]

    X_train <- data_train[, 1 : (p - 1)]
    y_train <- data_train[, p]
    X_test <- data_test[, 1 : (p - 1)]
    y_test <- data_test[, p]
    n <- nrow(X_train)
    p <- ncol(X_train) + 1
```

```
## Full model
m_full <- lm(medv ~ ., data = data_train)
y_pred <- predict(object = m_full, newdata = X_test)
error[1, i] <- mean((y_pred - y_test) ^ 2)

cbind(1, as.matrix(X_test)) %*% as.matrix(coef(m_full))

## All Subset
outs <- summary(regsubsets(x = X_train, y = y_train,
                           nvmax = p, method = "exhaustive"))
RSS <- outs$rss

AIC <- n * log(RSS / (n)) + 2 * (apply(outs$which, 1, sum))
BIC <- n * log(RSS / (n)) + log(n) * (apply(outs$which, 1, sum))
ix_AIC <- which.min(AIC)
ix_BIC <- which.min(BIC)
x_AIC <- X_train[, outs$which[ix_AIC, -1]]
x_BIC <- X_train[, outs$which[ix_BIC, -1]]
data_AIC <- data.frame(x_AIC, medv = y_train)
data_BIC <- data.frame(x_BIC, medv = y_train)
m_AIC <- lm(medv ~ ., data = data_AIC)
m_BIC <- lm(medv ~ ., data = data_BIC)
y_pred <- predict(object = m_AIC, newdata = data.frame(X_test))
error[2, i] <- mean((y_pred - y_test) ^ 2)
y_pred <- predict(object = m_BIC, newdata = data.frame(X_test))
error[3, i] <- mean((y_pred - y_test) ^ 2)

## Lasso
m_lasso <- lars(x = as.matrix(X_train), y = y_train, type = "lasso")
foo <- summary(m_lasso)
RSS <- foo$Rss
BIC <- n * log(RSS / n) + log(n) * foo$Df
ix_BIC <- which.min(BIC)
y_pred <- predict(object = m_lasso, newx = as.matrix(X_test), s = ix_BIC)
error[4, i] <- mean((y_pred$fit - y_test) ^ 2)

## ridge
m_ridge <- glmnet(as.matrix(X_train), y_train,
                  family = "gaussian", alpha = 0,
                  intercept=TRUE, lambda = seq(10, 0.01, -0.01),
                  standardize = FALSE)
tmp <- predict(m_ridge, newx = as.matrix(X_train))
foo <- tmp - as.matrix(y_train) %*% matrix(1, nrow = 1, ncol = ncol(tmp))
k <- function(x) {sum(x ^ 2)}
RSS <- apply(foo, 2, k)
```

```
    BIC <- n * log(RSS / n) + log(n) * (m_ridge$df + 1)
    ix_BIC <- which.min(BIC)
    y_pred <- predict(m_ridge, newx = as.matrix(X_test), s = m_ridge$lambda[ix_BIC])
    error[5, i] <- mean((y_pred - y_test) ^ 2)
}

rownames(error) <- c("Full Model", "All Subset AIC",
                     "All Subset BIC", "Lasso BIC",
                     "Ridge BIC")
colnames(error) <- c("CV1", "CV2")
```

Here we use half-half CV to compare their performances. Since only the best linear model without polynomial terms will be considered, it is reasonable to assume that this problem is in a kind of parametric senario, hence we apply BIC to tune parameters for Lasso and Ridge regression. we report the CV result as an error matrix in table 1.

Table 1: Error Matrix for Half-half CV

	CV1	CV2
Full Model	25.83	25.05
All Subset AIC	25.89	24.14
All Subset BIC	26.53	24.76
Lasso BIC	25.81	24.69
Ridge BIC	25.90	25.12

As we can see in table 1, the overall prediction error for each model is almost the same, **therefore it is hard to determine which modelling strategy is the best.** The reason of similar preditcion capability lies in the fact that **all the linear models are underfitted.** When we try to tune parameters for the ridge model, the lambda selected by BIC is very close to 0, suggesting that there is no need to introduce penalty on this problem, which means models of linear form is underfitted. Since we still have enough degrees of freedom to play, we can involve some polynomial terms into the model to achieve better generealziation power.