Model Selection for Regression

Adam J Sullivan, PhD

04/04/2018

Variable Selection

- We have discussed what linear regression is and how to check the assumptions and evaluate the model we have.
- A key issue remains still and that is how do we appropriately build a good model for the data?
- How do we select the variables that we wish to include in this "good" model?

All Subsets Regression

- There are a number of methods for choosing variable selection.
- Let us consider systolic blood pressure again.
- This time we will brainstorm what all might predict a persons systolic blood pressure

Variables We Can Consider

- BMI - `bmi1`
- Age - `age1`
- Diastolic Blood Pressure - `diapbp1`
- Hypertension - `prevhyp1`
- CVD - `prevcvd1`
- Heart Attack - `prevmi1`
- Smoking Status - `cursmoke1`
- Number of Cigarettes per day - `cigpday1`
- Blood Pressure Meds - `bpmeds1`

Why These Variables?

- These may not be all the predictors but they will provide us with a wealth of models in order to best fit systolic blood pressure.
- We will begin first with a concept called all possible subsets.
- With this we consider all m variables in the full model and all 2^m possible subsets. We will show how to use this in R.

Leaps Package

```
####################
##     RUN THIS IN R FOR CLASS    ##
####################

 library(leaps)
 leaps <- regsubsets(sysbp1t~ bmi1 + age1 +
                       diabp1 + prevhyp1 + prevchd1 + prevmi1 +
 cursmoke1 + cigpday1 + bpmeds1, data=fhs, nbest=1)

summary(leaps)
```

What is This Code?

- The code for this is there, we call up the `leaps` package and then from this use the `regsubsets()` function.
- We place the full model into the command followed by what my dataset is.
- Finally we ask it to display the number of best subsets.

What does this do?

- Essentially if we choose `nbest=1` we are asking for the top subset for each of possible subsets of size 1 up to 8.
- we could view the top 2, 3 or more if we change this.
- Finally the `summary()` function gives us the results of what we have.
- Running this code gives us

Leaps Package

```
library(leaps)
leaps <- regsubsets(sysbp1t~ bmi1 + age1 +
                      diabp1 + prevhyp1 + prevchd1 + prevmi1 +
cursmoke1 + cigpday1 + bpmeds1, data=fhs, nbest=1)
```

Summary of Leaps

```
library(dplyr)
summ <- summary(leaps)
knitr::kable(as_data_frame(summ$which))
```

Summary of Leaps

BMI1	AGE1	DIABP1	PREVHYP1YES	PREVCHD1YES	PREVMI1YES	CURSMOKE1YES	CIGPDAY1	BPMEDS1YES
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE

What do We see?

- We can then see what variables would be in the best model subset from a subset of size 1 up to 8.
- A quick look into this function and we find that we can also find out a number of other pieces of information.

What Else does Leaps give?

```
names(summ)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

Useful Information

We can then see that

- would give us a vector with the R^2_{adj} value for each of the 8 models.
- would give us a vector with all of the BIC for each of the 8 models.

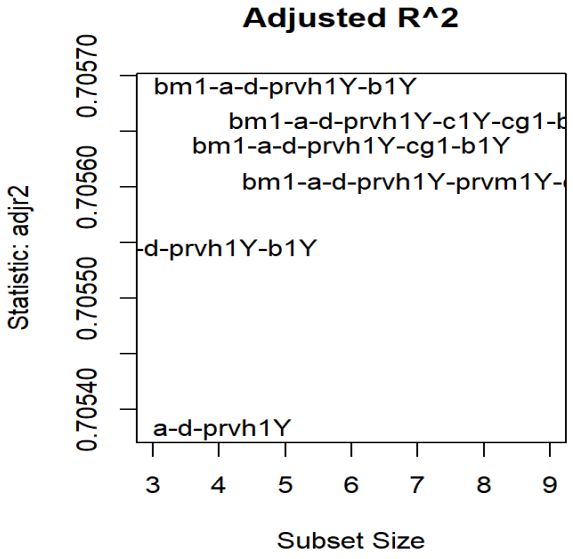
Using R^2_{adj}

- We could then use these to create a table of values we care about for model selection.
- We could also graph R^2_{adj}:

Using R^2_{adj}

```
library(car)
layout(matrix(1:2, ncol = 2))
# Adjusted R2
res.legend <- subsets(leaps, statistic="adjr2", legend = FALSE, min.size = 3,
main = "Adjusted R^2")
# Mallow Cp
res.legend <- subsets(leaps, statistic="cp", legend = FALSE, min.size = 3,
main = "Mallow Cp")
abline(a = 1, b = 1, lty = 2)
```

Using R^2_{adj}


Adjusted R^2
Statistic: adjr2
0.70570
0.70560
0.70550
0.70540
bm1-a-d-prvh1Y-b1Y
bm1-a-d-prvh1Y-c1Y-cg1-b
bm1-a-d-prvh1Y-cg1-b1Y
bm1-a-d-prvh1Y-prvm1Y-
-d-prvh1Y-b1Y
a-d-prvh1Y
3
4
5
6
7
8
9
Subset Size

Using R^2_{adj}

- We would then have the following legend for these plots

```
res.legend
```

```
##              Abbreviation
## bmi1                  bm1
## age1                    a
## diabp1                  d
## prevhyp1Yes         prvh1Y
## prevchd1Yes         prvc1Y
## prevmi1Yes          prvm1Y
## cursmoke1Yes           c1Y
## cigpday1               cg1
## bpmeds1Yes             b1Y
```

What does this tell us?

- We can see from the figure that the model with the 5 predictors has the highest R^2_{adj}. This is the model

$$sysbp1t = \beta_0 + \beta_1 bmi1 + \beta_2 age1 + \beta_3 diabp1 + \beta_4 prevhyp1 + \beta_5 bpmeds1$$

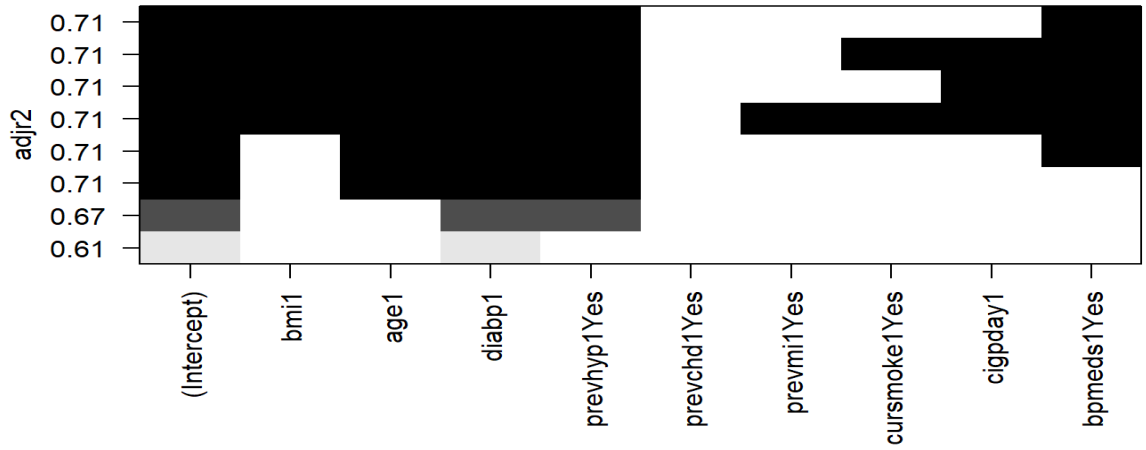
R^2_{adj} Plot

- Finally we could create one more plot with R^2_{adj}

```
plot(leaps, scale="adjr2", main="")
```

R^2_{adj} Plot

- Finally we could create one more plot with R^2_{adj}



Stepwise Regression

- Another concept for model selection can be is stepwise methods.
- The most used stepwise methods would either be a **forward selection** or **backwards elimination**.

Forward Selection Procedure

- **Forward selection** starts with no variables in the mode.
- It then runs all possible regressions univariately.
- It chooses the value with the lowest p-value.
- Once a variable is in the model it then checks all regressions with 2 variables and adds the one with the lowest p-value.
- It will continue to do this until everything is in the model or adding another variable will make the model worse.

Forward Selection Example

- Before we try this method we will do some data cleaning.
- Our first goal will be to create a dataset where all of the variables in our model have no missing data.
- This is called complete case analysis.

Complete Case Preparation

```
#Variables we care about in model
myvars <- c("sysbp1t", "bmi1", "age1", "diabp1", "prevhyp1", "prevchd1",
"prevmi1", "cursmoke1", "cigpday1", "bpmeds1")

#Selecting only these variables
fhs_sub_sel <- fhs[myvars]

#Selecting only those rows with complete cases
fhs_sub_sel <- fhs_sub_sel[complete.cases(fhs_sub_sel),]
```

Running Forwards Stepwise

Now we are ready to run forwards selection either using AIC or BIC

```
####################
##     RUN THIS IN R FOR CLASS    ##
####################

library(MASS)
#Begin with model of only intercept
int_mod <- lm(sysbp1t~ 1, data=fhs_sub_sel)

#Fit forward with AIC
step_aic <- stepAIC(int_mod, scope= ~ bmi1 + age1 + diabp1 + prevhyp1 + prevchd1 + prevmi1 +  cursmoke1
          cigpday1 + bpmeds1,  direction="forward", k=2)
step_aic$anova

#Define penalty for BIC
n = dim(fhs_sub_sel)[1]

#Fit forward with BIC
step_bic <- stepAIC(int_mod, scope= ~ bmi1 + age1 + diabp1 +
prevhyp1 + prevchd1 + prevmi1 +  cursmoke1 +
  cigpday1 + bpmeds1,  direction="forward", k=log(n))
step_bic$anova
```

Backwards Stepwise

- **Backwards elimination** starts with everything in the model.
- It then removes the variable with the largest p-value.
- It runs the model again and looks for the next variable with the largest p-value.
- It will continue until either there are no variables in the model or removing a variable will make the model worse.

Running Backwards Regression

```
####################
##     RUN THIS IN R FOR CLASS    ##
####################

library(MASS)
# Start with everything in the model
full_mod <- lm(sysbp1t~ bmi1 + age1 + diabp1 + prevhyp1 + prevchd1 +
prevmi1 + cursmoke1 + cigpday1 + bpmeds1, data=fhs_sub_sel)


#Fit backward with AIC
step_aic <- stepAIC(full_mod,   direction="backward", k=2)
step_aic$anova

#Define penalty for BIC
n = dim(fhs_sub_sel)[1]

#Fit bacward with BIC
step_bic <- stepAIC(full_mod,   direction="backward", k=log(n))
step_bic$anova
```

Comparing different Models

- Sometimes we wish to directly compare models.
- For example when we have a categorical variable, many times we want to treat this as a continuous variable and others we want to declare a reference group and let each group have it's own slope.

Example: The PBC-3 trial in liver cirrhosis

- The PBC-3 trial was a randomized clinical trial conducted in six European hospitals.
- Between 1983 and 1987, 349 patients with liver disease, primary biliary cirrhosis (PBC), were randomized to a treatment or a placebo.
- The purpose of the trial was to study the effect of the treatment on Survival time.

PBC Data

- We have considered the `pbc` data before.
- We considered platelet counts and we have a categorical variable of histological disease stage.
- We may be interested in seeing if we can treat this as linear or by categories.

Read in PBC Data

```
pbc <- read.table("pbc.csv", header=TRUE, sep=",")
colnames(pbc) <- c("casenum", "n.days", "death", "treated", "age" , "sex",
"ascites", "hepatomegaly", "spiders", "edema",
"bilirubin", "choles", "albumin", "ur.cop",
"alk.phos", "sgot", "trigly",
"plat.cnt", "pt", "dis.stage")
```

Fit Histologic Disease Stage As Linear

```
fit.pbc1 <- lm(plat.cnt ~ dis.stage, data=pbc)
kable(tidy(fit.pbc1, conf.int=T)[, -c(3,4)])
```

Fit Histologic Disease Stage As Linear

TERM	ESTIMATE	P.VALUE	CONF.LOW	CONF.HIGH
(Intercept)	345.2	0	307.5	383.0
dis.stage	-27.4	0	-39.3	-15.5

Conclusions

- In this model we viewed the change in disease stages to be linear and that it was the same change from group 1 to 2 as from 2 to 3.
- However we may have wanted to let each group have their own slope and compare these models.

Fit Histologic Disease Stage As Categorical

```
fit.pbc2 <- lm(plat.cnt ~ as.factor(dis.stage), data=pbc)
kable(tidy(fit.pbc2, conf.int=T)[, -c(3,4)])
```

Fit Histologic Disease Stage As Categorical

TERM	ESTIMATE	P.VALUE	CONF.LOW	CONF.HIGH
(Intercept)	278.47	0.000	231.5	325.388
as.factor(dis.stage)2	18.76	0.478	-33.2	70.741
as.factor(dis.stage)3	-7.91	0.755	-57.7	41.907
as.factor(dis.stage)4	-49.51	0.052	-99.6	0.533

F-Test for Model Comparison

- Then we could use the F-test to compare both of these models and determine which one is better to use.
- The F-test considers two models M_1 and M_2. M_1 has p_1 covariates and a residual sum of squares, RSS_1.
- M_2 has $p_2 > p_1$ and a residual sum of squares, RSS_2.
- We also have that M_1 is nested inside of M_2.

F-Test for Model Comparison

- Then we have

$$F = \frac{\frac{RSS_1 - RSS_2}{p_2 - p_1}}{\frac{RSS_2}{N - p_2 - 1}}$$

- Under the assumptions that the residuals of both models are normally distributed we have that F follows an F distribution with $p_2 - p_1$ and $N - p_2 - 1$ degrees of freedom.

Conclusion

- In our example we have that the model with disease stage coded with dummy variables is the larger model and when we consider a linear trend across the stages we have this model nested inside the larger model.
- We can perform this F-test in R.

F-Test in R

```
anova(fit.pbc1, fit.pbc2)
```

F-Test in R

```
## Analysis of Variance Table
##
## Model 1: plat.cnt ~ dis.stage
## Model 2: plat.cnt ~ as.factor(dis.stage)
##   Res.Df     RSS Df Sum of Sq    F Pr(>F)
## 1    306 2630489
## 2    304 2592651  2     37838 2.22   0.11
```

Conclusion

- This yields an $F = 2.218$, with a p-value of 0.111.
- Recall: we are testing the null hypothesis that the smaller model fits the data just as well as the larger model.
- This means the alternative would be that we need the larger model.
- In this case we reject in favor of the simpler model.
- We would then not need a specific slope for each disease stage but would allow for a linear trend across the disease stages.