Regression II: Model selection

-Applied Multivariate Analysis-

Lecturer: Darren Homrighausen, PhD

Computing a Good Estimator of β?

On the previous slides, we proposed a sensible solution to picking a good model:

Solution: Minimize AIC (or another related criterion) to get $\hat{\beta}_{\text{good}}$

However, this doesn't completely solve the problem.

How can we do the necessary minimization?

Data Analysis Example: Prostate Cancer Data

Here is how to read in the data, along with a description of the variables:

```
prostate = read.table('prostate.data',header=T)
# Variables are:
# lcavol:  log cancer volume
# lweight: log prostate weight
# age:     patient age
# lbph:    log of amount of benign prostate hyperplasia
# svi:     seminal vesicle invasion (0,1 valued)
# lcp:     log of capsular penetration
# gleason: Gleason score
# pgg45:   Percent of Gleason scores 4 or 5
# lpsa:    log prostate specific antigen (response)
```

Goal

We wish to find which ones (if any) of the explanatory variables are important. To do this, we can fit the full linear model:

```
> fit.lm = lm(lpsa~.,data=prostate)
> summary(fit.lm)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.181561   1.320568   0.137  0.89096
lcavol       0.564341   0.087833   6.425 6.55e-09 ***
lweight      0.622020   0.200897   3.096  0.00263 **
age         -0.021248   0.011084  -1.917  0.05848 .
lbph         0.096713   0.057913   1.670  0.09848 .
svi          0.761673   0.241176   3.158  0.00218 **
lcp         -0.106051   0.089868  -1.180  0.24115
gleason      0.049228   0.155341   0.317  0.75207
pgg45        0.004458   0.004365   1.021  0.31000
```

However, generally some of the predictors are unimportant.

What happens if we estimate too many parameters?

Repercussions of estimating too many parameters

Suppose we have independent random variables $Z_1, Z_2, \ldots, Z_p$ all with variance σ^2.

If we form the sum of all of the Z's

$$S = \sum_{j=1}^{p} Z_j = Z_1 + Z_2 + \cdots + Z_p$$

then

$$\mathbb{V}S = \sum_{j=1}^{p} \mathbb{V}Z_j = \sum_{i=1}^{p} \sigma^2 = p\sigma^2$$

Conclusion: The more random variables we add to the sum, the higher the variance.

Repercussions of estimating too many parameters

Remember: $\text{pred} = \text{variance} + \text{bias}^2$

So, if we want to make good predictions with sums, we need to make sure we are only including important parameters

Linear Regression:

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \, ||Y - \mathbb{X}\beta||_2^2 = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top Y$$

To see this, note:

$$\begin{aligned} \nabla_\beta \, ||Y - \mathbb{X}\beta||_2^2 &= \nabla_\beta (-2Y^\top \mathbb{X}\beta + \beta^\top \mathbb{X}^\top \mathbb{X}\beta) \\ &= -2Y^\top X + 2\beta^\top \mathbb{X}^\top \mathbb{X} \overset{set}{=} 0 \end{aligned}$$

solve and show the Hessian is positive definite...

(or demonstrate convexity)

Repercussions of estimating too many parameters

Important Connection: Regression is just a fancy sum!.

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \, ||Y - \mathbb{X}\beta||_2^2 = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top Y$$

For instance:

$$\mathbb{X}^\top Y = \begin{bmatrix} \sum_{i=1}^n \mathbb{X}_{i1} Y_i \\ \sum_{i=1}^n \mathbb{X}_{i2} Y_i \\ \vdots \\ \sum_{i=1}^n \mathbb{X}_{ip} Y_i \end{bmatrix} \in \mathbb{R}^p \Rightarrow \text{There are } p \text{ summands}$$

All subsets regression

One idea is, as the name suggests, to compute all possible models and report the model with the smallest score. R has a very good package for doing this: leaps. Let's define the following[1]

```
Y = prostate$lpsa
X = prostate[,names(prostate)!=c('lpsa','train')]
n = length(Y)
p = ncol(X)
```

[1] Ignore the stuff about 'train,' we'll get to this soon.

All subsets regression

```
library(leaps)
leaps.out = leaps(x = X,y = Y,method='r2',nbest=40)
> names(leaps.out)
[1] "which" "label" "size"  "r2"
```

The R object 'leaps.out' contains the R^2 for each model of size 2, 3, 4, up to 9. For example, a model of size 4 would be:

$$\mathbb{E}Y_i = \beta_0 + \beta_1 \mathrm{lcavol}_i + \beta_2 \mathrm{svi}_i + \beta_3 \mathrm{lcp}_i$$

All subsets regression

The function leaps is designed to efficiently compute a bunch of models. However, it doesn't actually do any model selection.

It does report R^2, which is defined to be:

$$R^2 = 1 - \frac{\text{MSE}}{\text{total sum of squares}} \iff \text{MSE} = \text{SST}(1 - R^2)$$

We can use this to form the criterion we are interested in using

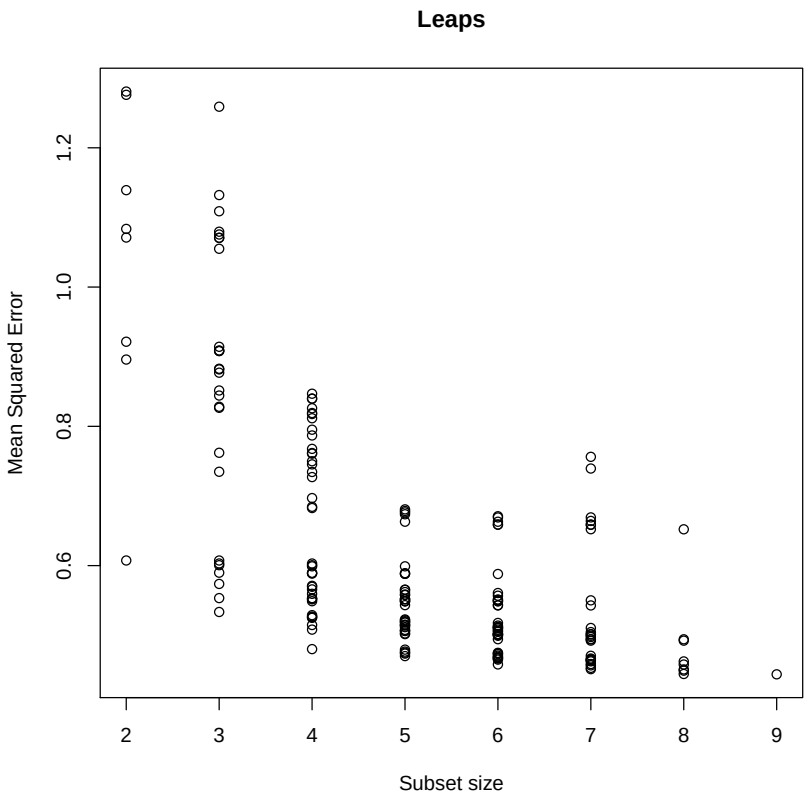
Note:

- MSE in this expression isn't divided by n, we don't need to do that for this class
- SST is the total sums of squares (residuals around grand mean)

All subsets regression

Using this definition of R^2:

```
SStot      = sum( (Y-mean(Y))^2 )
MSE        = SStot*(1 - leaps.out$r2)
```



- Note how MSE is strictly decreasing for larger models. This is the same behavior as we noticed in the polynomial example
- We can restrict our attention to the 8 models that have the smallest MSE for a given size (Why?)

All Subsets Regression: First way

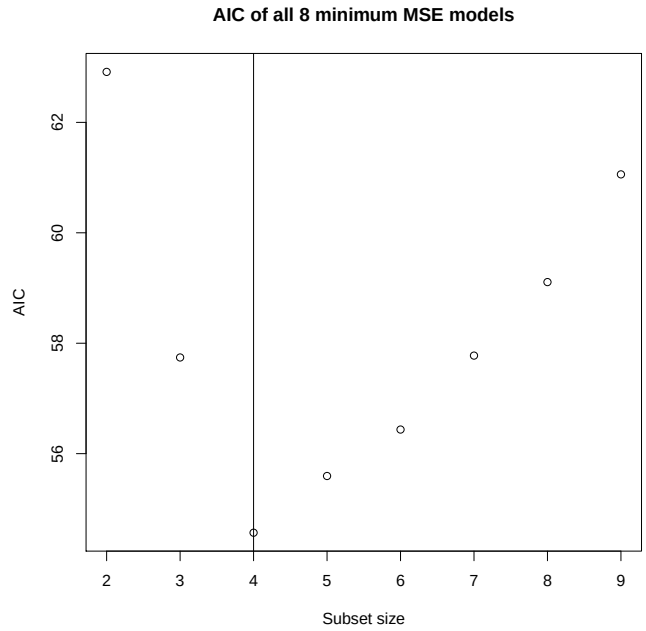
We can compute the AIC for each of the 8 MSE minimizing models using this code:

```
AIC = rep(0,8)
sweep = 1
for(i in 2:9){
  AIC[sweep] = min(MSE[leaps.out$size == i]) + 2*i
  sweep = sweep + 1
}
pdf('../lectures/figures/leapsProstateExampleAIC.pdf')
  plot(2:9,AIC,xlab='Subset size',ylab='AIC',
          main='AIC of all 8 minimum MSE models')
  abline(v=seq(2,9)[which.min(AIC)])
dev.off()
```

Here, we have used the fact that another way to write AIC is:

$$\text{AIC}(\hat{\beta}) = \text{MSE} + 2|\hat{\beta}|$$

All Subsets Regression: First way



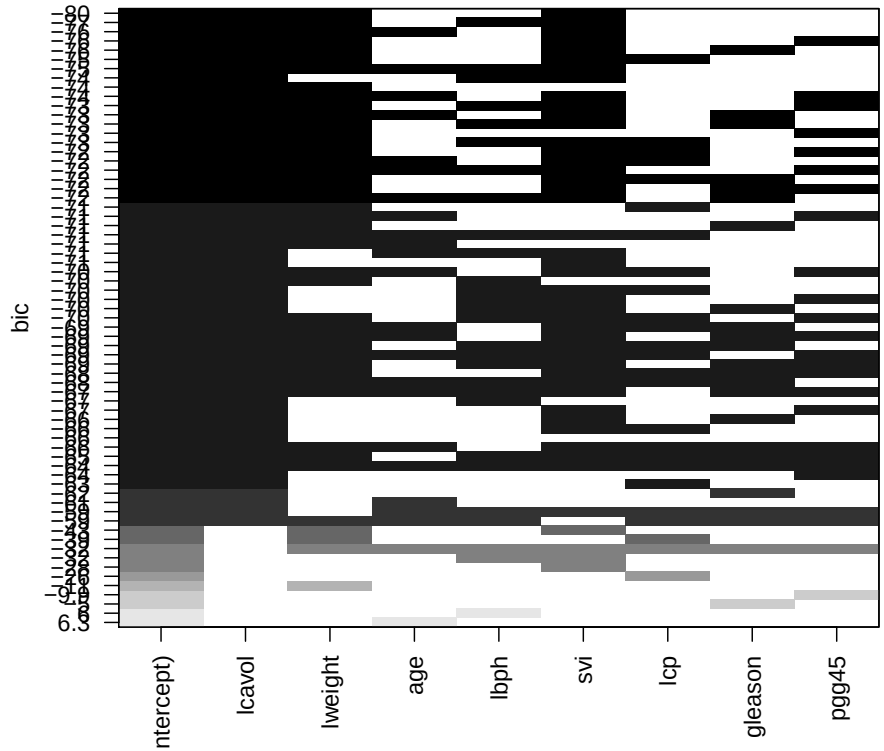
Here is the best model:

```
> leaps.out$which[leaps.out$size == 4,][1,]
    1     2     3     4     5     6     7     8
 TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
 > names(X)[leaps.out$which[leaps.out$size == which.min(AIC)+1,]
[1] "lcavol"  "lweight" "svi"
```

This corresponds to picking: lcavol, lweight, svi

All Subsets Regression: A Second Way

```
leaps.plot = regsubsets(Y~.,data=X, nbest=10)
plot(leaps.plot,scale='bic')
```



The plot is ordered from best to worst (top to bottom)

Here, BIC suggests we keep lcavol, lweight, and svi.

All Subsets Regression: A Big Problem (Literally)

If there are p possible predictors (the previous example had 8), then there are $2^p - 1$ possible models (the previous example had 255).

For instance, if $p = 40$ (which is considered a small problem these days), then the number of possible models is

$$2^{40} - 1 \approx 1,099,512,000,000 \Rightarrow \text{More than 1 trillion!}$$

This is huge!

If $p = 265$, then the number of possible models is the same as the number of atoms in the universe[2]

We need a way to sift through the models in a computationally feasible way

[2] It is estimated there are 10^{80} atoms in the universe.

Forward Selection

A good way to do this sifting is through greedy algorithms.

If we can't look at all possible models, we can do the following instead:

1. Find the best possible one predictor model[3]
2. Keep that predictor and find the best possible two predictor model given the original predictor is selected.
3. Keep both those predictors and find the best possible three predictor model given the first two selected.

   ⋮
4. Keep going until adding another predictor no longer reduces AIC[4].

[3] All of these steps assume an intercept is included and always retained

[4] Some programs/people will advocate using F-tests for the selection criterion. There are reasons to prefer either, but generally choose AIC.

Forward Selection: R Output

```
> null = lm(Y~1,data=X)
> full = lm(Y~.,data=X)
> step(null,scope=list(lower=null,upper=full),
            direction='forward')
Start:  AIC=28.84
Y ~ 1

           Df Sum of Sq     RSS     AIC
+ lcavol    1    69.003  58.915 -44.366
+ svi       1    41.011  86.907  -6.658
+ lcp       1    38.528  89.389  -3.926
+ lweight   1    24.019 103.899  10.665
+ pgg45     1    22.814 105.103  11.783
+ gleason   1    17.416 110.502  16.641
+ lbph      1     4.136 123.782  27.650
+ age       1     3.679 124.239  28.007
<none>                 127.918  28.838
```

Forward Selection: R Output

```
Step:  AIC=-44.37
Y ~ lcavol

           Df Sum of Sq    RSS     AIC
+ lweight   1    7.1726 51.742 -54.958
+ svi       1    5.2375 53.677 -51.397
+ lbph      1    3.2658 55.649 -47.898
+ pgg45     1    1.6980 57.217 -45.203
<none>                  58.915 -44.366
+ lcp       1    0.6562 58.259 -43.452
+ gleason   1    0.4156 58.499 -43.053
+ age       1    0.0025 58.912 -42.370
```

Forward Selection: R Output

```
Step:  AIC=-54.96
Y ~ lcavol + lweight

           Df Sum of Sq    RSS     AIC
+ svi       1    5.1737 46.568 -63.177
+ pgg45     1    1.8158 49.926 -56.424
<none>                  51.742 -54.958
+ lcp       1    0.8187 50.923 -54.506
+ gleason   1    0.7163 51.026 -54.311
+ age       1    0.6456 51.097 -54.176
+ lbph      1    0.4440 51.298 -53.794
```

Forward Selection: R Output

```
Step:  AIC=-63.18
Y ~ lcavol + lweight + svi

           Df Sum of Sq    RSS     AIC
+ lbph      1   0.97296 45.595 -63.226
<none>                  46.568 -63.177
+ age       1   0.62301 45.945 -62.484
+ pgg45     1   0.50069 46.068 -62.226
+ gleason   1   0.34449 46.224 -61.898
+ lcp       1   0.06937 46.499 -61.322
```

Forward Selection: R Output

```
Step:  AIC=-63.23
Y ~ lcavol + lweight + svi + lbph

           Df Sum of Sq    RSS     AIC
+ age       1   1.15879 44.437 -63.723
<none>                  45.595 -63.226
+ pgg45     1   0.33173 45.264 -61.934
+ gleason   1   0.20691 45.389 -61.667
+ lcp       1   0.10115 45.494 -61.441
```

Forward Selection: R Output

```
Step:  AIC=-63.72
Y ~ lcavol + lweight + svi + lbph + age

          Df Sum of Sq    RSS     AIC
<none>                  44.437 -63.723
+ pgg45    1   0.66071 43.776 -63.176
+ gleason  1   0.47674 43.960 -62.769
+ lcp      1   0.13040 44.306 -62.008
```

Conclusion: Forward selection with AIC suggests we keep lcavol, lweight, age, lbph, and svi

Backward Selection: R Output

```
> out  = step(full,direction='backward')
Start:  AIC=-60.78
Y ~ lcavol + lweight + age + lbph + svi + lcp +
      gleason + pgg45

           Df Sum of Sq    RSS     AIC
- gleason  1    0.0491 43.108 -62.668
- pgg45    1    0.5102 43.569 -61.636
- lcp      1    0.6814 43.740 -61.256
<none>                43.058 -60.779
- lbph     1    1.3646 44.423 -59.753
- age      1    1.7981 44.857 -58.810
- lweight  1    4.6907 47.749 -52.749
- svi      1    4.8803 47.939 -52.364
- lcavol   1   20.1994 63.258 -25.467
```

Backward Selection: R Output

```
Step:  AIC=-62.67
Y ~ lcavol + lweight + age + lbph + svi + lcp + pgg45

           Df Sum of Sq    RSS     AIC
- lcp       1    0.6684 43.776 -63.176
<none>                  43.108 -62.668
- pgg45     1    1.1987 44.306 -62.008
- lbph      1    1.3844 44.492 -61.602
- age       1    1.7579 44.865 -60.791
- lweight   1    4.6429 47.751 -54.746
- svi       1    4.8333 47.941 -54.360
- lcavol    1   21.3191 64.427 -25.691
```

Backward Selection: R Output

```
Step:  AIC=-63.18
Y ~ lcavol + lweight + age + lbph + svi + pgg45

           Df Sum of Sq    RSS     AIC
- pgg45     1    0.6607 44.437 -63.723
<none>                  43.776 -63.176
- lbph      1    1.3329 45.109 -62.266
- age       1    1.4878 45.264 -61.934
- svi       1    4.1766 47.953 -56.336
- lweight   1    4.6553 48.431 -55.373
- lcavol    1   22.7555 66.531 -24.572
```

Backward Selection: R Output

```
Step:  AIC=-63.72
Y ~ lcavol + lweight + age + lbph + svi

           Df Sum of Sq    RSS     AIC
<none>                  44.437 -63.723
- age       1    1.1588 45.595 -63.226
- lbph      1    1.5087 45.945 -62.484
- lweight   1    4.3140 48.751 -56.735
- svi       1    5.8509 50.288 -53.724
- lcavol    1   25.9427 70.379 -21.119
```

Conclusion: Backward selection with AIC suggests we keep lcavol, lweight, age, lbph, and svi

Stepwise Selection: R Output

```
> out  = step(null,scope=list(upper=full),
              direction='both')
Start:  AIC=28.84
Y ~ 1

           Df Sum of Sq     RSS     AIC
+ lcavol    1    69.003  58.915 -44.366
+ svi       1    41.011  86.907  -6.658
+ lcp       1    38.528  89.389  -3.926
+ lweight   1    24.019 103.899  10.665
+ pgg45     1    22.814 105.103  11.783
+ gleason   1    17.416 110.502  16.641
+ lbph      1     4.136 123.782  27.650
+ age       1     3.679 124.239  28.007
<none>                 127.918  28.838
```

Stepwise Selection: R Output

```
Step:  AIC=-44.37
Y ~ lcavol

           Df Sum of Sq     RSS     AIC
+ lweight   1     7.173  51.742 -54.958
+ svi       1     5.237  53.677 -51.397
+ lbph      1     3.266  55.649 -47.898
+ pgg45     1     1.698  57.217 -45.203
<none>                   58.915 -44.366
+ lcp       1     0.656  58.259 -43.452
+ gleason   1     0.416  58.499 -43.053
+ age       1     0.003  58.912 -42.370
- lcavol    1    69.003 127.918  28.838
```

Stepwise Selection: R Output

```
Step:  AIC=-54.96
Y ~ lcavol + lweight

           Df Sum of Sq     RSS     AIC
+ svi       1     5.174  46.568 -63.177
+ pgg45     1     1.816  49.926 -56.424
<none>                   51.742 -54.958
+ lcp       1     0.819  50.923 -54.506
+ gleason   1     0.716  51.026 -54.311
+ age       1     0.646  51.097 -54.176
+ lbph      1     0.444  51.298 -53.794
- lweight   1     7.173  58.915 -44.366
- lcavol    1    52.157 103.899  10.665
```

Stepwise Selection: R Output

```
Step:  AIC=-63.18
Y ~ lcavol + lweight + svi

           Df Sum of Sq    RSS     AIC
+ lbph      1    0.9730 45.595 -63.226
<none>                  46.568 -63.177
+ age       1    0.6230 45.945 -62.484
+ pgg45     1    0.5007 46.068 -62.226
+ gleason   1    0.3445 46.224 -61.898
+ lcp       1    0.0694 46.499 -61.322
- svi       1    5.1737 51.742 -54.958
- lweight   1    7.1089 53.677 -51.397
- lcavol    1   24.7058 71.274 -23.893
```

Stepwise Selection: R Output

```
Step:  AIC=-63.23
Y ~ lcavol + lweight + svi + lbph

           Df Sum of Sq    RSS     AIC
+ age       1    1.1588 44.437 -63.723
<none>                  45.595 -63.226
- lbph      1    0.9730 46.568 -63.177
+ pgg45     1    0.3317 45.264 -61.934
+ gleason   1    0.2069 45.389 -61.667
+ lcp       1    0.1012 45.494 -61.441
- lweight   1    3.6907 49.286 -57.675
- svi       1    5.7027 51.298 -53.794
- lcavol    1   24.9384 70.534 -22.906
```

Stepwise Selection: R Output

```
Step:  AIC=-63.72
Y ~ lcavol + lweight + svi + lbph + age

           Df Sum of Sq    RSS     AIC
<none>                  44.437 -63.723
- age       1    1.1588 45.595 -63.226
+ pgg45     1    0.6607 43.776 -63.176
+ gleason   1    0.4767 43.960 -62.769
- lbph      1    1.5087 45.945 -62.484
+ lcp       1    0.1304 44.306 -62.008
- lweight   1    4.3140 48.751 -56.735
- svi       1    5.8509 50.288 -53.724
- lcavol    1   25.9427 70.379 -21.119
```

Conclusion: Stepwise selection with AIC suggests we keep lcavol, lweight, age, lbph, and svi

A Comparison

Summary of results:

- All Subsets (BIC): lcavol, lweight, svi
- All Subsets (AIC): lcavol, lweight, svi
- Forward Selection: lcavol, lweight, age, lbph, svi
- Backward Selection: lcavol, lweight, age, lbph, svi
- Stepwise Selection: lcavol, lweight, age, lbph, svi

Note that

- For all subsets, both AIC and BIC chose the same model
- All the greedy approaches selected the same model for this data

Important: None of this is not necessarily the case!

An Alternate Selection Method in R

Inside the package leaps, the function regsubsets[5] can be used to do:

- forwards
- backwards
- all subsets

It uses Mallows Cp as the criterion instead of AIC

Let's investigate it further

[5] The nice thing about regsubsets over step is that it doesn't need to fit the full model for scope.

An Alternate Selection Method in R: Forward

```
library(leaps)
regfit.for = regsubsets ( x = X,y = Y, nvmax =19,
                          method ="forward")
regfit.for.sum = summary(regfit.for)

> regfit.for.sum$which[which.min(regfit.for.sum$cp),]
(Intercept)      lcavol     lweight         age     lbph
       TRUE        TRUE        TRUE        TRUE     TRUE
        svi         lcp     gleason       pgg45
       TRUE       FALSE       FALSE       FALSE
```

An Alternate Selection Method in R: Backward

```
library(leaps)
regfit.bac = regsubsets ( x = X,y = Y, nvmax =19,
                                method ="backward")
regfit.bac.sum = summary(regfit.bac)

> regfit.bac.sum$which[which.min(regfit.bac.sum$cp),]
(Intercept)      lcavol     lweight         age      lbph
       TRUE        TRUE        TRUE        TRUE      TRUE
        svi         lcp     gleason       pgg45
       TRUE       FALSE       FALSE       FALSE
```

Warning: You might be climbing...