ORIGIN ≡ 0

Choosing an Optimal Linear Model

In exploring fit of a linear model relating dependent variable Y with a set of independent variables X_1, X_2, ... X_n that may be either numerical values (Multiple Regression), dummy variables (ANOVA), or a combination of the two (Analysis of Covariance or ANCOVA), some criterion is needed in order to decide which model among several might be the best. There is no simple answer. In general, a "parsimonious" model - one that is as simple as possible - is always to be preferred. However, several measures of parsimony may potentially be applied. In addition, it is often useful to have a formal test to help decide whether one model is better than another. Shown here are several useful measures of parsimony and application of one of them - AIC - in marginal and stepwise approaches to finding an optimal model. Example is from Chapter 9 of Kuter et al. (KNNL) ***Applied Linear Statistical Models*** **5th Edition.**

Example in R:

```
#CHOOSING AN OPTIMAL LINEAR MODEL
K=read.table("c:/DATA/Biostatistics/KNNLCh9SurgicalUnit.txt")
K
attach(K)
options(digits=6)

#FITTING FULL LINEAR MODEL
FM=lm(Y~X1+X2+X3+X4+X5+factor(X6))

#SETTING UP MATRIX ALGEBRA OBJECTS AND HAT MATRIX
n=length(Y)
I=diag(n)
J=matrix(nrow=n,ncol=n,1)
X=model.matrix(FM)
p=ncol(X)
H=X%*%solve(t(X)%*%X)%*%t(X) #HAT MATRIX

#CALCULATING SUMS OF SQUARES
#USING MATRIX ALGEBRA IN R
SSR=t(Y)%*%(H-((1/n)*J))%*%Y  #Sum of Squares Regression
SSE=t(Y)%*%(I-H)%*%Y  #Sum of Squares Error
SSTO=t(Y)%*%(I-(1/n)*J)%*%Y  #Sum of Squares Total
SSTO
rbind(SSR,SSE,SSTO)
```

Note: calculation of Sums of Squares as quadratic forms using matrix algebra is presented in detail in *Biostatistics* Worksheet 38

```
          [,1]
[1,]  9.90246   SSR
[2,]  2.90527   SSE
[3,] 12.80773   SSTO
```

Surgical Unit Example KNNL Table 9.1

	X1	X2	X3	X4	X5	X6	Y
1	6.7	62	81	2.59	50	0	6.544
2	5.1	59	66	1.70	39	0	5.999
3	7.4	57	83	2.16	55	0	6.565
4	6.5	73	41	2.01	48	0	5.854
5	7.8	65	115	4.30	45	0	7.759
6	5.8	38	72	1.42	65	1	5.852
7	5.7	46	63	1.91	49	1	6.250
8	3.7	68	81	2.57	69	1	6.619
9	6.0	67	93	2.50	58	0	6.962
10	3.7	76	94	2.40	48	0	6.875
11	6.3	84	83	4.13	37	0	6.613
12	6.7	51	43	1.86	57	0	5.549
13	5.8	96	114	3.95	63	1	7.361
14	5.8	83	88	3.95	52	1	6.754
15	7.7	62	67	3.40	58	0	6.554
16	7.4	74	68	2.40	64	1	6.695
17	6.0	85	28	2.98	36	1	6.526
18	3.7	51	41	1.55	39	0	5.321
19	7.3	68	74	3.56	59	1	6.309
20	5.6	57	87	3.02	63	0	6.731
21	5.2	52	76	2.85	39	0	5.883
22	3.4	83	53	1.12	67	1	5.866
23	6.7	26	68	2.10	30	0	6.395
24	5.8	67	86	3.40	49	1	6.332
25	6.3	59	100	2.95	36	1	6.478
26	5.8	61	73	3.50	62	1	6.621
27	5.2	52	86	2.45	70	0	6.302
28	11.2	76	90	5.59	58	1	7.583
29	5.2	54	56	2.71	44	1	6.167
30	5.8	76	59	2.58	61	1	6.396
31	3.2	64	65	0.74	53	0	6.094
32	8.7	45	23	2.52	68	0	5.198
33	5.0	59	73	3.50	57	0	6.019
34	5.8	72	93	3.30	39	1	6.944
35	5.4	58	70	2.64	31	1	6.179
36	5.3	51	99	2.60	48	0	6.453
37	2.6	74	86	2.05	45	0	6.519
38	4.3	8	119	2.85	65	1	5.893
39	4.8	61	76	2.45	51	1	6.457
40	5.4	52	88	1.81	40	1	6.558
41	5.2	49	72	1.84	46	0	6.283
42	3.6	28	99	1.30	55	0	6.366
43	8.8	86	88	6.40	30	1	7.147
44	6.5	56	77	2.85	41	0	6.288
45	3.4	77	93	1.48	69	0	6.178
46	6.5	40	84	3.00	54	1	6.416
47	4.5	73	106	3.05	47	1	6.867
48	4.8	86	101	4.10	35	1	7.170
49	5.1	67	77	2.86	66	1	6.365
50	3.9	82	103	4.55	50	0	6.983
51	6.6	77	46	1.95	50	0	6.005
52	6.4	85	40	1.21	58	0	6.361
53	6.4	59	85	2.33	63	0	6.310
54	8.8	78	72	3.20	56	0	6.478

```
#COMPARING WITH anova() OUTPUT
anova(FM)
```

Sum "extra" SS for X1-X6 to obtain SSR

```
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value    Pr(>F)
X1         1  0.776   0.776  12.558 0.000904 ***
X2         1  2.589   2.589  41.880 5.19e-08 ***
X3         1  6.334   6.334 102.470 2.16e-13 ***
X4         1  0.025   0.025   0.398 0.531382
X5         1  0.126   0.126   2.046 0.159218
X6         1  0.052   0.052   0.845 0.362735
Residuals 47  2.905   0.062
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Criteria Information for Model Selection:

When removing independent variables:

Sum of Squares Error:

look for little further drop in SSE for a model with some subset of X_i's

```
#SUM OF SQUARES ERROR
SSE[1]  #from matrix calculation above converted to scalar
summary(FM)$sigma^2*summary(FM)$df[2] #extracted from summary()
```

```
[1] 2.90527
```

Coefficient of Multiple Determination (R^2):

look for little further rise in R^2 for a model with some subset of X_i's

```
#COEFFICIENT OF MULTIPLE DETERMINATION
Rsq=1-(SSE[1]/SSTO[1])
Rsq
summary(FM)$r.squared #extracted from summary()
```

```
[1] 0.773163
```

Adjusted Coefficient of Multiple Determination:

look for little further rise in R_{adj}^2 for a model with some subset of X_i's

```
#ADJUSTED COEFFICIENT OF MULTIPLE DETERMINATION
MSE=SSE[1]/(n-p)
MSTO=SSTO[1]/(n-1)
Rsqa=1-(MSE/MSTO)
Rsqa
summary(FM)$adj.r.squared #extracted from summary()
```

```
[1] 0.744205
```

Mallow's C_p Criterion:

seek subsets of X_i's with small Cp value, and Cp near p

```
#MALLOW'S Cp
Cp=SSE[1]/MSE -(n-2*p)
Cp
require(wle) #DOWNLOAD {wle} PACKAGE
 FROM CRAN WEBSITE
mle.cp(FM)  #DON'T KNOW MUCH ABOUT THIS ONE,
 BUT INTERESTING!
```

```
[1] 7
```

```
Call:
mle.cp(formula = FM)
Mallows Cp:
     (Intercept) X1 X2 X3 X4 X5 factor(X6)1   cp
[1,]           1  1  1  1  0  1           0 3.89
[2,]           1  1  1  1  0  1           1 5.00
[3,]           1  1  1  1  1  1           0 5.84
[4,]           1  1  1  1  1  1           1 7.00
Printed the first  4  best models
```

Akaike's Information Criterion (AIC):

```
#AKAIKE'S INFORMATION CRITERION AIC
AIC=n*log(SSE[1])-n*log(n)+2*p
AIC
extractAIC(FM)  #REPORTS: (equivalent df, AIC)  see ?extractAIC
```

seek minimum AIC values for subsets of X_i's

```
[1] -143.813

[1]     7.000 -143.813
```

Schwartz's Bayesian Criterion

```
#SCHWARTZ'S BAYESIAN CRITERION SBC
SBC=n*log(SSE[1])-n*log(n)+log(n)*p
SBC
extractAIC(FM,k=log(n))
```

seek minimum SBC values for subsets of X_i's

```
[1] -129.890

[1]     7.000 -129.890
```

PRESS Criterion:

```
#PRESS CRITERION
e=residuals(FM)
d=e/(1-diag(H))  #KNNL Eq. 10.21a
diag(H)       #MAIN DIAGONAL OF H AS A VECTOR
hatvalues(FM)   #ALTERNATE CALCULATION USING BUILT-IN FUNCTION & FM
PRESS=sum(d^2)
PRESS
```

seek minimum PRESS values for subsets of X_i's

```
[1] 4.15084
```

Automated Model Selection:

The R statistical package contains several routines allowing one to use some of the above criteria to find relatively optimal models. Most widely used is AIC as a criterion in a "backwards" mode - that is, by removing independent variables one a a time from the full model.

Marginal Testing of single variables at a time:

```
#MARGINAL TESTING OF EACH INDEPENDENT VARIABLE
drop1(FM)
require(car)
Anova(FM)  #Marginal extra sums of squares ANOVA table
```

Each independent variable is droped from the full model at a time. In drop1() choose the smallest AIC. Note: one can also conduct a marginal F-test for each variable using Anova() in the {car} package.

drop 1:

```
Single term deletions
Model:
Y ~ X1 + X2 + X3 + X4 + X5 + factor(X6)
           Df Sum of Sq   RSS    AIC
<none>                  2.905 -143.8
X1          1     0.646 3.551 -135.0
X2          1     1.971 4.876 -117.8
X3          1     3.651 6.556 -101.9
X4          1     0.000 2.905 -145.8
X5          1     0.141 3.046 -143.2
factor(X6)  1     0.052 2.957 -144.8
```

ANOVA in {car}:

```
Anova Table (Type II tests)

Response: Y
           Sum Sq Df F value    Pr(>F)
X1          0.646  1  10.448  0.00225 **
X2          1.971  1  31.888 9.18e-07 ***
X3          3.651  1  59.061 7.55e-10 ***
X4          0.000  1   0.003  0.95670
X5          0.141  1   2.283  0.13748
factor(X6)  0.052  1   0.845  0.36273
Residuals   2.905 47
```

Automated "backwards" Stepwise Regression:

#STEPWISE REGRESSION USING AIC
step(FM,direction="backward")

```
Start:  AIC=-143.81
Y ~ X1 + X2 + X3 + X4 + X5 + factor(X6)

             Df Sum of Sq   RSS    AIC
- X4          1     0.000 2.905 -145.8
- factor(X6)  1     0.052 2.957 -144.8
<none>                    2.905 -143.8
- X5          1     0.141 3.046 -143.2
- X1          1     0.646 3.551 -135.0
- X2          1     1.971 4.876 -117.8
- X3          1     3.651 6.556 -101.9
```

Removing X4 gives > smallest AIC this round

```
Step:  AIC=-145.81
Y ~ X1 + X2 + X3 + X5 + factor(X6)

             Df Sum of Sq   RSS     AIC
- factor(X6)  1     0.055 2.960 -146.80
<none>                    2.905 -145.81
- X5          1     0.151 3.057 -145.07
- X1          1     1.157 4.062 -129.71
- X2          1     2.476 5.382 -114.52
- X3          1     5.984 8.889  -87.42
```

Refit model excluding X4 >

Removing X6 gives > smallest AIC this round

```
Step:  AIC=-146.8
Y ~ X1 + X2 + X3 + X5

       Df Sum of Sq   RSS     AIC
<none>              2.960 -146.80
- X5    1     0.148 3.109 -146.16
- X1    1     1.188 4.148 -130.58
- X2    1     2.609 5.569 -114.67
- X3    1     6.300 9.260  -87.22
```

Refit model excluding X6 >

Full model has lowest AIC this round, so STOP >

```
Call:
lm(formula = Y ~ X1 + X2 + X3 + X5)

Coefficients:
(Intercept)           X1           X2           X3           X5

    4.02810      0.09483      0.01319      0.01641     -0.00476
```

Final model >

Note: both "backward" (starting from a Full Model and removing an independent variable at each step) and "forward" (starting from a small model and adding an independent variable at each step) stepwise regression is available in R. There is even a form of "both". Most generally useful, I think, is "backward" starting with a full set of independent variables and whittling them down to a useful set. One hopes the result is optimal, but there's no guarantee. Playing with adding and subtracting sets of variables at a time sometimes gives a better result.