model selection in linear regression

basic problem: how to choose between competing linear regression models

model too small: "underfit" the data; poor predictions; high bias; low variance

model too big: "overfit" the data; poor predictions; low bias; high variance

model just right: balance bias and variance to get good predictions
Bias-Variance Tradeoff

High Bias - Low Variance

Low Bias - High Variance
“overfitting” - modeling the random component
Too Many Predictors?

When there are lots of $X$’s, get models with high variance and prediction suffers. Three “solutions:”

1. Pick the “best” model
   Cross-validation
   Score: AIC, BIC
   All-subsets + leaps-and-bounds,
   Stepwise methods,

2. Shrinkage/Ridge Regression

3. Derived Inputs
Cross-Validation

• e.g. 10-fold cross-validation:
  ▪ Randomly divide the data into ten parts
  ▪ Train model using 9 tenths and compute prediction error on the remaining 1 tenth
  ▪ Do these for each 1 tenth of the data
  ▪ Average the 10 prediction error estimates

“One standard error rule”
pick the simplest model within one standard error of the minimum
Library(DAAG)
houseprices.lm <- lm(sale.price ~ area, data=houseprices)
CVlm(houseprices, houseprices.lm, m=3)

fold 1
Observations in test set: 3 12 13 14 15
   X11  X20  X21  X22  X23
x=area  802.0 696 771.0 1006.0 1191
Predicted  204.0 188 199.3  234.7  262
sale.price 215.0 255 260.0  293.0  375
Residual   11.0   67  60.7   58.3  113

Sum of squares = 24000   Mean square = 4900   n = 5

fold 2
Observations in test set: 2 5 6 9 10
   X10  X13  X14  X17  X18
x=area  905  716 963.0 1018.00 887.00
Predicted  255  224 264.4  273.38 252.06
sale.price 215.0 113 185.0  276.00 260.00
Residual   -40  -112  -79.4   2.62  7.94

Sum of squares = 20000   Mean square = 4100   n = 5
fold 3
Observations in test set: 1 4 7 8 11

<table>
<thead>
<tr>
<th></th>
<th>X9</th>
<th>X12</th>
<th>X15</th>
<th>X16</th>
<th>X19</th>
</tr>
</thead>
<tbody>
<tr>
<td>x=area</td>
<td>694.0</td>
<td>1366</td>
<td>821.00</td>
<td>714.0</td>
<td>790.00</td>
</tr>
<tr>
<td>Predicted</td>
<td>183.2</td>
<td>388</td>
<td>221.94</td>
<td>189.3</td>
<td>212.49</td>
</tr>
<tr>
<td>sale.price</td>
<td>192.0</td>
<td>274</td>
<td>212.00</td>
<td>220.0</td>
<td>221.50</td>
</tr>
<tr>
<td>Residual</td>
<td>8.8</td>
<td>-114</td>
<td>-9.94</td>
<td>30.7</td>
<td>9.01</td>
</tr>
</tbody>
</table>

Sum of squares = 14000    Mean square = 2800    n = 5

Overall ms
3934

> summary(houseprices.lm)$sigma^2
[1] 2321

> CVlm(houseprices,houseprices.lm,m=15)

Overall ms
3247
Quicker solutions

• AIC and BIC try to mimic what cross-validation does

• AIC(MyModel)

• Smaller is better
Quicker solutions

• If have 15 predictors there are $2^{15}$ different models (even before considering interactions, transformations, etc.)

• “Leaps and bounds” is an efficient algorithm to do all-subsets
# All Subsets Regression

```r
library(leaps)

leaps <- regsubsets(newAccounts~., data=bank, nbest=1, nvmax=15)
summary(leaps)

MySummary <- summary(leaps)
MySummary$bic

# plot a table of models showing variables in each model
# models are ordered by the selection statistic.
# plot(leaps)

# plot statistic by subset size
library(car)
subsets(leaps, statistic="rsq")
```
Variable selection with pure noise using leaps

```r
y <- rnorm(100)
xx <- matrix(rnorm(4000), ncol=40)
dimnames(xx) <- list(NULL, paste("X", 1:40, sep=""))

library(leaps)
xx.subsets <- regsubsets(xx, y, method="exhaustive", nvmax=3, nbest=1)
subvar <- summary(xx.subsets)$which[3,-1]
best3.lm <- lm(y ~ -1 + xx[, subvar])
print(summary(best3.lm, corr=FALSE))

or...bestsetNoise(m=100,n=40)
```
run this experiment ten times:

- all three significant at $p<0.01$ 1
- all three significant at $p<0.05$ 3
- two out of three significant at $p<0.05$ 3
- one out of three significant at $p<0.05$ 1
• Stepwise methods are very popular but can perform badly

library(MASS)
fit <- lm(newAccounts~.,data=bank)
step <- stepAIC(fit, direction="both")
step$anova # display results
Transformations

- log, exp, sqrt, sqr, cube root, cube, etc.

**box-cox:**

$$y^{(\lambda)} = \begin{cases} 
(y^\lambda - 1)/\lambda, & \lambda \neq 0 \\
\log y, & \lambda = 0 
\end{cases}$$
bc <- function(x,l) {(x^l-1)/l}
l<-seq(1,4,0.5)
x<-seq(-3,3,0.1)
par(new=FALSE)
for (i in l) {
    plot(x,bc(x,i),type="l",ylim=c(-20,20),col=2*i);
    par(new=TRUE)
}
legend("top",paste(l),col=2*l,lty=rep(1,length(l)))
y=number of days absent from school in Quine, Australia

> attach(quine)
> table(Lrn,Age,Sex,Eth)
  , , Sex = F, Eth = A

<table>
<thead>
<tr>
<th></th>
<th>Lrn</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

, , Sex = M, Eth = A

<table>
<thead>
<tr>
<th></th>
<th>Lrn</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

, , Sex = F, Eth = N

<table>
<thead>
<tr>
<th></th>
<th>Lrn</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1</td>
<td>11</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

, , Sex = M, Eth = N

<table>
<thead>
<tr>
<th></th>
<th>Lrn</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
plot(lm(Days ~ Eth*Sex*Age*Lrn, data=quine),which=1:4)
> boxcox(Days+1 ~ Eth*Sex*Age*Lrn, data = quine, lambda = seq(-0.05, 0.45, len = 20))
quineBC <- quine
quineBC$Days <- (quineBC$Days^0.2 - 1)/0.2
plot(lm(Days ~ Eth*Sex*Age*Lrn, data=quineBC),which=1:4)
Shrinkage Methods

• Subset selection is a discrete process – individual variables are either in or out

• This method can have high variance – a different dataset from the same source can result in a totally different model

• Shrinkage methods allow a variable to be partly included in the model. That is, the variable is included but with a shrunken co-efficient.
Ridge Regression

$$\hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

subject to: $$\sum_{j=1}^{p} \beta_j^2 \leq s$$

Equivalently:

$$\hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \left( \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

This leads to:

$$\hat{\beta}_{\text{ridge}} = \left( X^T X + \lambda I \right)^{-1} X^T y$$

Choose $\lambda$ by cross-validation. Predictors should be centered. Works even when $X^T X$ is singular.
effective number of $X$'s
The Lasso

$$
\hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2
$$

subject to: \( \sum_{j=1}^{p} |\beta_j| \leq s \)

Quadratic programming algorithm needed to solve for the parameter estimates. Choose \( s \) via cross-validation.

$$
\tilde{\beta} = \arg \min_{\beta} \left( \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right)
$$

\( q=0: \) var. sel.  
\( q=1: \) lasso  
\( q=2: \) ridge

Learn \( q \)?
library(glmnet)
bank <- read.table("/Users/dbm/Documents/W2025/BankSortedMissing.TXT",header=TRUE)
names(bank)
bank <- as.matrix(bank)
x <- bank[1:200,-1]
y <- bank[1:200,1]
fit1 <- glmnet(x,y)

xTest <- bank[201:233,-1]
predict(fit1,newx=xTest,s=c(0.01,0.005))
predict(fit1,type="coef")
plot(fit1,xvar="lambda")
MyCV <- cv.glmnet(x,y)