# Statistics 452: Statistical Learning and Prediction

## Chapter 6, Part 1: Linear Model Selection

Brad McNeney

2018-10-10

# Introduction

# Alternatives to Least Squares

- We have used least squares to fit the linear model

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon. \tag{1}$$

- In this chapter we consider alternative methods of fitting the model, with the goal of better prediction accuracy and model interpretability when $p$ is large.
  - Prediction accuracy: Unless $n$ is much larger than $p$ there is a tendancy to overfit, leading to poor predictions on the test set. In case $p > n$ there is no unique least squares solution.
  - Model interpretability: It is often the case that only a small subset of the predictors is truly associated with the response. The model is more interpretable without irrelevant variables.

# Approaches in this Chapter

- Each of the following can be thought of as a strategy to reduce variance, with (hopefully) minimal increase in bias.
- Subset selection: Forward, backward, stepwise and all subsets selection to identify truly associated model terms.
- Shrinkage (regularization): Shrink estimated coefficients toward zero.
- Dimension reduction: Find a low-dimension representation of the predictors, and use these as predictors.

# Subset Selection

# Best (All) Subset Selection

- Straightforward idea: Consider all $2^p$ possible models ($p$ with one predictor, $\binom{p}{2} = p(p-1)/2$ with two predictors, etc.) and choose the one with the best estimated test set error.
  - Can use cross validation to estimate test set error, or computationally cheaper alternatives ($C_p$, BIC – to be discussed).
- Break the exhaustive search for the best of all models into two steps:
  1. Fit all $\binom{p}{k}$ models with $k$ predictors and select the one, call it $\mathcal{M}_k$, with the smallest RSS.
  2. Select the best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ based on estimated test set error.
- See Algorithm 6.1 in test for a complete algorithm.

# Drawback of All Subsets

- Computational: $2^p$ becomes very large as $p$ increases.

```
p<-10; 2^p
```

```
## [1] 1024
```

```
p<-20; 2^p
```

```
## [1] 1048576
```

# Example of All Subsets

```r
uu <- url("http://www-bcf.usc.edu/~gareth/ISL/Credit.csv")
Credit <- read.csv(uu,row.names=1)
head(Credit,n=3)
```

```
##    Income Limit Rating Cards Age Education Gender Student Married
## 1  14.891  3606    283     2  34        11   Male      No     Yes
## 2 106.025  6645    483     3  82        15 Female     Yes     Yes
## 3 104.593  7075    514     4  71        11   Male      No      No
##   Ethnicity Balance
## 1 Caucasian     333
## 2     Asian     903
## 3     Asian     580
```

```r
library(leaps) # contains regsubsets()
cfits <- regsubsets(Balance ~ ., data=Credit,nvmax=11)
cfits.sum <- summary(cfits)
```

```r
cfits.sum$which
```

```
##    (Intercept) Income Limit Rating Cards   Age Education GenderFemale
## 1         TRUE  FALSE FALSE   TRUE FALSE FALSE     FALSE        FALSE
## 2         TRUE   TRUE FALSE   TRUE FALSE FALSE     FALSE        FALSE
## 3         TRUE   TRUE FALSE   TRUE FALSE FALSE     FALSE        FALSE
## 4         TRUE   TRUE  TRUE  FALSE  TRUE FALSE     FALSE        FALSE
## 5         TRUE   TRUE  TRUE   TRUE  TRUE FALSE     FALSE        FALSE
## 6         TRUE   TRUE  TRUE   TRUE  TRUE  TRUE     FALSE        FALSE
## 7         TRUE   TRUE  TRUE   TRUE  TRUE  TRUE     FALSE         TRUE
## 8         TRUE   TRUE  TRUE   TRUE  TRUE  TRUE     FALSE         TRUE
## 9         TRUE   TRUE  TRUE   TRUE  TRUE  TRUE     FALSE         TRUE
## 10        TRUE   TRUE  TRUE   TRUE  TRUE  TRUE     FALSE         TRUE
## 11        TRUE   TRUE  TRUE   TRUE  TRUE  TRUE      TRUE         TRUE
##    StudentYes MarriedYes EthnicityAsian EthnicityCaucasian
## 1       FALSE      FALSE          FALSE              FALSE
## 2       FALSE      FALSE          FALSE              FALSE
## 3        TRUE      FALSE          FALSE              FALSE
## 4        TRUE      FALSE          FALSE              FALSE
## 5        TRUE      FALSE          FALSE              FALSE
## 6        TRUE      FALSE          FALSE              FALSE
## 7        TRUE      FALSE          FALSE              FALSE
## 8        TRUE      FALSE           TRUE              FALSE
## 9        TRUE       TRUE           TRUE              FALSE
## 10       TRUE       TRUE           TRUE               TRUE
## 11       TRUE       TRUE           TRUE               TRUE
```

```r
cfits.sum$rss
```

```
##  [1] 21435122 10532541  4227219  3915058  3866091  3821620  3810759
##  [8]  3804746  3798367  3791345  3786730
```

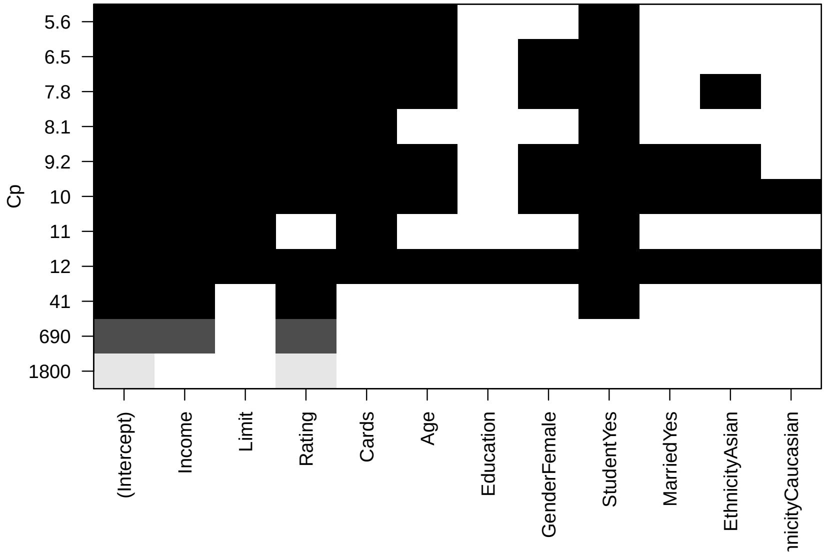
```r
cfits.sum$rsq
```

```
##  [1] 0.7458484 0.8751179 0.9498788 0.9535800 0.9541606 0.9546879 0.9548167
##  [8] 0.9548880 0.9549636 0.9550468 0.9551016
```

```r
cfits.sum$cp
```

```
##  [1] 1800.308406  685.196514   41.133867   11.148910    8.131573
##  [6]    5.574883    6.462042    7.845931    9.192355   10.472883
## [11]   12.000000
```

```r
plot(cfits,scale="Cp")
```



# RSS and $R^2$ for Model Selection

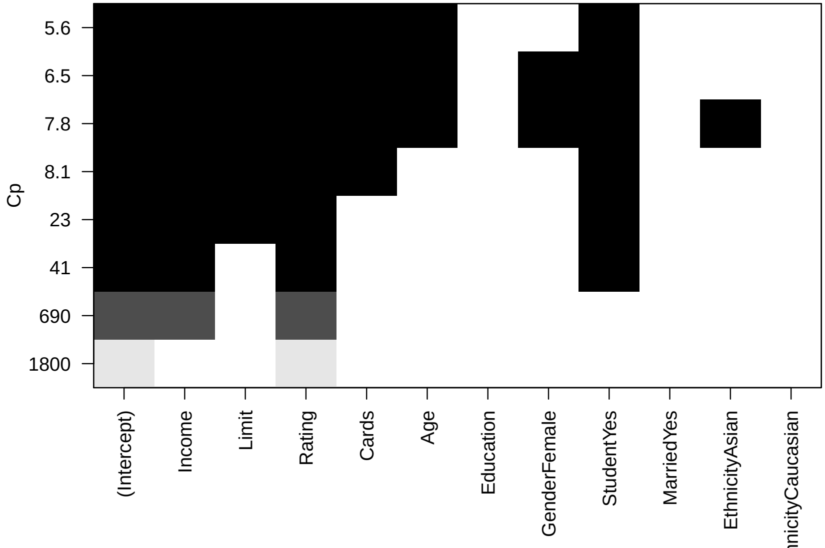
- RSS always decreases when we add predictors, even if the added predictors are, in fact, unrelated to the response.
  - $k$ predictors: Least squares finds the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ that minimize RSS.
  - $k+1$ predictors: Least squares can reduce RSS compared to coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k, 0$.
- Similarly, $R^2 = 1 - RSS/TSS$ always increases.
- Neither is useful for comparing models of different size.
  - Will define $C_p$ and other measures soon.

# Forward Selection

- Select the best model of each size through the following restricted search:
  - Start with the null model, $\mathcal{M}_0$, that contains no predictors.
  - Consider the best model, $\mathcal{M}_1$ with 1 predictor.
  - Consider the best model, $\mathcal{M}_2$ obtained by adding one of the $p-1$ terms **not** in $\mathcal{M}_1$.
  - Consider the best model, $\mathcal{M}_3$ obtained by adding one of the $p-2$ terms **not** in $\mathcal{M}_2$.
  - And so on.
- Then use the estimated test set error to select the best from $\mathcal{M}_0, \ldots, \mathcal{M}_p$.
- See Algorithm 6.2.

# Example Forward Selection

```r
cfits.fwd <- regsubsets(Balance ~ .,data=Credit,
                        method="forward")
plot(cfits.fwd,scale="Cp")
```



# Advantages and Disadvantages of Forward Selection

- Advantages:
  - Far less computation. Can show forward selection only fits $1 + p(p+1)/2$ models. With $p = 20$, $2^p = 1048686$ while $\$1{+}\mathrm{p}(\mathrm{p}{+}1)/2 = 211$.
  - Can be applied even when $p > n$.
- Disadvantage:
  - Not guaranteed to find the best model.

# Backward Selection

- Reverse of forward selection: Start with the largest model and remove the least predictive predictor one at a time.
  - Start with the full model $\mathcal{M}_p$.
  - Consider the best model, $\mathcal{M}_{p-1}$, obtained by removing one of the $p$ terms in $\mathcal{M}_p$.
  - Consider the best model, $\mathcal{M}_{p-2}$ obtained by removing one of the $p-1$ terms in $\mathcal{M}_{p-1}$.
  - And so on.
- Then use the estimated test set error to select the best from $\mathcal{M}_0, \ldots, \mathcal{M}_p$.
- See Algorithm 6.3.

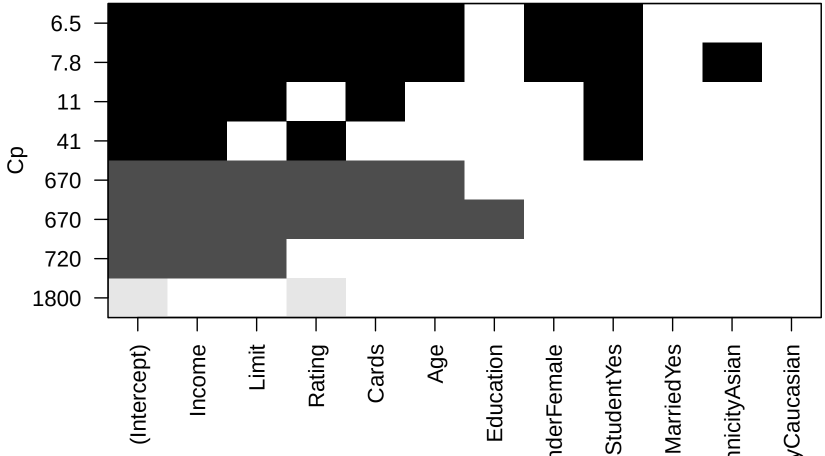
# Advantages and Disadvantages of Backward Selection

- Advantage:
  - Same computation as forward selection. Only fits $1 + p(p+1)/2$ models.
- Disadvantage:
  - Not guaranteed to find the best model.

# Hybrid Stepwise Selection

- Iterate between adding and deleting model terms in the search for a best model.

```r
cfit.hybrid <- regsubsets(Balance ~ ., data=Credit,
                          method="seqrep")
plot(cfit.hybrid,scale="Cp")
```



# Model Comparisons and Estimated Test Error

- Estimated test error is a basis for model comparison.
- Methods for estimating test error are classified as indirect or direct.
- Indirect methods estimate the "optimism", which is roughly the difference between the test and training errors.
  - That is, test error = training error + optimism
    and estimated test error = training error + estimated optimism
- Direct methods use validation or cross-validation.

# Indirect methods

- $C_p$, AIC and BIC are in this class.
- $C_p$ for a model with $d$ (subset of $p$) predictors is defined as

$$C_p = \frac{1}{n}(\mathrm{RSS} + 2d\hat{\sigma}^2)$$

or (Mallow's definition)

$$C_p' = \frac{\mathrm{RSS}}{\hat{\sigma}^2} + 2d - n)$$

where $\hat{\sigma}^2$ is an estimate of $\sigma^2$ from a low-bias model.
- Ignoring scalings and constants that are the same for all models being compared, $C_p$ is essentially RSS plus a penalty that increases with $d$.

# AIC

- AIC stands for Akaike Information Criterion.
- AIC can be defined for many models fit by maximum likelihood.
- For linear regression with Gaussian errors AIC is essentially $C_p'$ up to scale and constant factors.
  - A difference is that $\hat{\sigma}^2$ in AIC is usually taken to be the estimate from the current model, rather than a fixed low-bias model.
  - For model selection with models fit by least squares, we usually report $C_p$ (or $C_p'$).

# BIC

- BIC stands for Bayesian Information Criterion and is a.k.a Schwartz's criterion.
- BIC is defined in the text as

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}(\mathrm{RSS} + \log_e(n) d\hat{\sigma}^2)$$

  to highlight similarities with their definition of $C_p$.
  - For BIC, replace the factor 2 by $\log(n)$ in the penalty term of $C_p$.
  - What matters is that BIC is essentially RSS plus a penalty that depends on $d$ and grows faster with $d$ because of the $\log(n)$.
  - $\log_e(n) > 2$ for $N > 7$.

# Aside: AIC and BIC in R

- R uses the formulas $AIC = -2\ell(\hat{\beta}, \hat{\sigma}^2) + 2p$ and $BIC = -2\ell(\hat{\beta}, \hat{\sigma}^2) + 2p$, where $\ell$ is the log-likelihood.
  - The likelihood is the probability of the data, considered as a function of the parameters.
  - $p$ is the number of model parameters that have been estimated, **including** $\sigma^2$.

```
set.seed(1); x <- 1:100; y <- x + rnorm(100)
ff <- lm(y~x)
logLik(ff)
```

```
## 'log Lik.' -130.6444 (df=3)
```

```
AIC(ff) # -2*logLik(ff) + 2*3
```

```
## [1] 267.2888
```

```
BIC(ff) # -2*logLik(ff) + log(100)*3
```

```
## [1] 275.1043
```

# Direct Methods

- Can use validation or cross-validation to directly estimate the test error.
  - Takes a little programming – see week 6 exercises.