6.1 Subset Selection

Lauren Perry

Linear Model Selection and Regularization

- Chapter 6 extends the linear model framework.
- Chapters 7 and 8 generalize the linear model and consider some general non-linear models.
- In this section, we focus on methods for selecting a subset of predictors.

Why Linear Models?

- Linear models may seem too simple, but…
  - they are highly interpretable.
  - they often have good predictive capabilities.
- In Chapter 6, we will discuss ways to improve the basic least squares fit by using some alternate fitting procedures.

Why consider alternatives to least squares?

- Prediction accuracy
  - especially important when $p > n$
- Model interpretablity
  - Removing irrelevant factors to simplify models can help with interpretability.
  - We will consider some approaches for automating *feature selection*

A Preview of Ch 6

- Subset selection (6.1)
  - Identify a subset of the p predictors to be used in the final model, which we fit using least squares.
- Shrinkage (6.2)
  - Fit a model using all p predictors, with some of the coefficients shrunken toward zero. This can help reduce variance and perform variable selection.
- Dimension Reduction (6.3)
  - Project the p predictors into an M-dimensional subspace, $M < p$ and use these as predictors in a linear regression model.

Best Subset Selection

To access the *best* subset, we need to fit a separate linear model to each possible combination of the predictors.

Best Subset Selection

Basic algorithm:

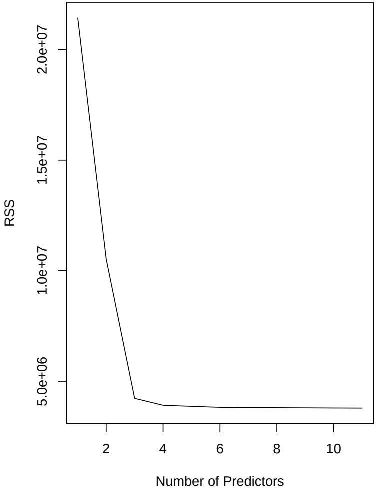
1. Let Ω_0 denote the *null model*, which contains no predictors: $\hat{y} = \bar{y}$.
2. For $k = 1, 2, \ldots, p$:
   Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   Select the best among these $\binom{p}{k}$ models and call it Ω_k. Here, "best" means smallest RSS (or largest R^2).
3. Select a single best model from among $\Omega_0, \ldots, \Omega_p$ using cross-validated prediction error, AIC, BIC, or Adjusted R^2.

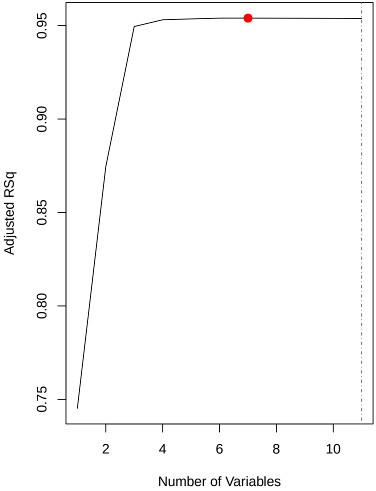
Example: Credit Data

```r
data("Credit")
library(leaps)
bestset <- regsubsets(Balance ~ ., Credit, nvmax=11)
reg.sumry <- summary(bestset)

par(mfrow = c(1, 2))
mx = which.max(reg.sumry$adjr2)
plot(reg.sumry$rss, xlab = "Number of Predictors",
    ylab = "RSS", type = "l")
plot(reg.sumry$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
points(mx, reg.sumry$adjr2[mx], col = "red", cex = 2, pch =
abline(v=11, lty=4, col="purple")
```

Example: Credit Data





Extentions to Other Models

- The best subsets approach can be used for other types of models.
- Where RSS is not available, we instead use the *deviance*
  - Deviance $= -2\times$ maximized log-likelihood

Stepwise Selection

Best subset selection has some significant drawbacks

- Cannot be applied with very large p.
- Larger search spaces increase the chances of finding models that look good on training data, but perform poorly on future data.
  - Huge search space can lead to *overfitting*.
- These can also be computationally intensive.

Forward Stepwise Selection

- We begin with a null model and then add predictors to the model one at a a time, until all predictors are included in the model.
- The predictor added at each step is selected based on which one results in the greatest additional improvement to the existing model.

Forward Stepwise Selection

The basic algorithm

1. Let Ω_0 denote the *null model*, which contains no predictors: $\hat{y} = \bar{y}$.
2. For $k = 1, 2, \ldots, p - 1$:
   Consider all $p - k$ models that augment the predictors in Ω_k with one additional predictor.
   Select the best among these models and call it Ω_{k+1}.
3. Select a single best model from among $\Omega_0, \ldots, \Omega_p$ using cross-validated prediction error, AIC, BIC, or Adjusted R^2.

Forward Stepwise Selection

Pro: computational advantage (Searches through only $1 + p(p+1)/2$ models)

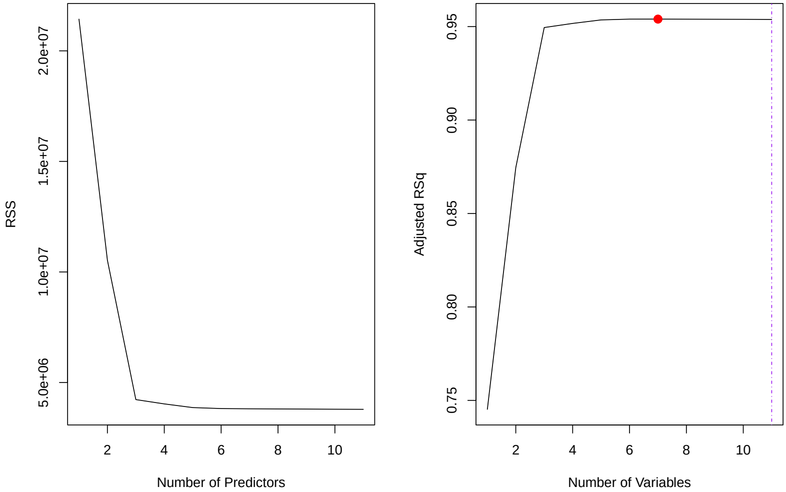
Con: not guaranteed to result in the best possible model

Example: Credit Data

```r
regfit.fwd <- regsubsets(Balance ~ ., Credit, nvmax = 11,
                         method = "forward")
regfit.smry <- summary(regfit.fwd)

par(mfrow = c(1, 2))
mx = which.max(regfit.smry$adjr2)
plot(regfit.smry$rss, xlab = "Number of Predictors",
    ylab = "RSS", type = "l")
plot(regfit.smry$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
points(mx, regfit.smry$adjr2[mx], col = "red", cex = 2,
    pch = 20)
abline(v=11, lty=4, col="purple")
```

Example: Credit Data



Example: Credit Data

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	**cards**, income, student, limit	**rating**, income, student, limit

Backward Stepwise Selection

- Similar to forward stepwise selection, but starts with the full model.
- Iteratively removes one predictor at a time.

Backward Stepwise Selection

The basic algorithm

1. Let Ω_p denote the *full model*, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   Consider all k models contain all but one of the predictors in Ω_k.
   Select the best among these models and call it Ω_{k-1}.
3. Select a single best model from among $\Omega_0, \ldots, \Omega_p$ using cross-validated prediction error, AIC, BIC, or Adjusted R^2.

Backward Stepwise Selection

Pro: computational advantage (Searches through only $1 + p(p + 1)/2$ models)

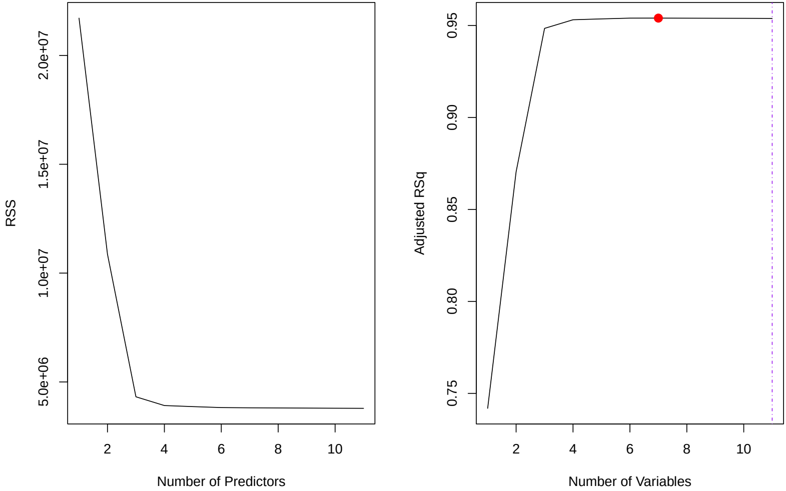
Cons: - not guaranteed to result in the best possible model - requires $n > p$ to run

Example Code: Credit Data

```r
regfit.bwd <- regsubsets(Balance ~ ., Credit, nvmax = 11,
                         method = "backward")
regfit.smry <- summary(regfit.bwd)

par(mfrow = c(1, 2))
mx = which.max(regfit.smry$adjr2)
plot(regfit.smry$rss, xlab = "Number of Predictors",
    ylab = "RSS", type = "l")
plot(regfit.smry$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
points(mx, regfit.smry$adjr2[mx], col = "red", cex = 2, pch
abline(v=11, lty=4, col="purple")
```

Example Code: Credit Data



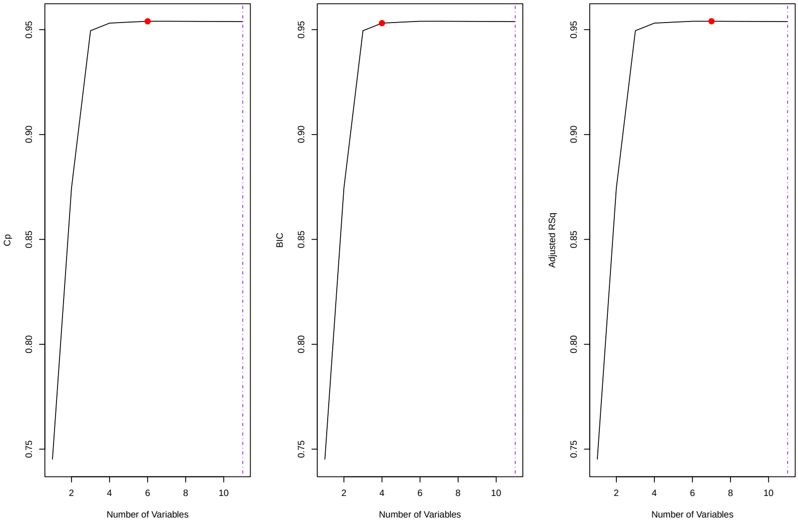
Choosing the Optimal Model

- The model contianing all the predictors will always have the smallest RSS and the largest R^2
- We want to choose a model with low test error, not low training error
- For all three approaches, we ended by selecting a single best model from among $\Omega_0, \ldots, \Omega_p$ using cross-validated prediction error, AIC, BIC, or Adjusted R^2.
  - These allow us to either *directly* estimate test error or to *adjust* test error to account for bias.

C_p, AIC, BIC, Adjusted R^2

- These techniques adjust the training error for the model size (penalize for complexity) and can be used to select among models with different numbers of variables.
  - These models do not need to be subsets of each other.

Credit Data Example (Best Subsets)



Some Details: Adjusted R^2

We saw these last semester, but we'll recap.

- R^2 is strictly decreasing as the number of variables in the model increases.
- Adjusted R^2 basically penalizes the R^2 statistic directly.

$$R^2_{adj} = 1 - \frac{\mathrm{RSS}/(n-d-1)}{\mathrm{TSS}/(n-1)}$$

- Maximizing R^2_{adj} is equivalent to minimizing $\mathrm{RSS}/(n-d-1)$
- We select the model with the greatest R^2_{adj}

Some Details: Mallow's C_p

- Mallow's C_p
$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$$
where d is the total number of parameters in the model and $\hat{\sigma}^2$ is the variance of the error associated with each response
- We select the model with the lowest C_p

Some Details: AIC

- AIC is defined for any models fit by maximum likelihood

$$\mathrm{AIC} = -2 \log L + 2d$$

  where L is the maximized likelihood for the estimated model
- For the linear model with Gaussian errors, maximum likelihood and least squares are the same (so C_p and AIC are equivalent)
- We select the model with the lowest AIC

Some Details: BIC

$$\mathrm{BIC} = -2 \log L + d \log(n)$$

- Similar to Mallow's C_p and AIC, but uses a different penalty term.
  - Replaces the 2 in AIC by $\log n$
  - Since $\log n > 2$ for $n > 7$, BIC tends to penalize largest variables more heavily than AIC and Mallow's C_p.

Validation and Cross-Validation

- Each of the procedures (best subset, stepwise) returns a sequence of models Ω_k, $k = 0, 1, 2, \ldots$
- We need to select k to determine our final model

Validation and Cross-Validation

- Compute the (cross) validation set error for each model under consideration, then select the k for which the resulting estimated error is smallest.
- Pros:
  - provides a direct estimate of test error
  - does not require estimation of σ
  - can be used in a wider range of model selection tasks, since we do not need to pinpoint degrees of freedom or calculate a likelihood
- Con: computationally intensive
- You will do this in the Chapter 6 Lab.

Selecting a Final Model

- It is often the case that each approach gives a slightly different result.
- We will select a model using the *one standard error rule*.
    - Calculate the standard error of the estimated test MSE for each model size.
    - Select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve.