Linear Regression: Variable Selection

36-600

The Setting

- Linear regression is an inferential (read: inflexible) model in which we assume that Y is related to the predictor variables $\mathbf{x}$ via the model

$$Y|\mathbf{x} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

  - ϵ represents the scatter of data around the regression line

- Today, we begin by pointing out a potentially obvious truism:

  - just because a predictor variable exists in your data table doesn't mean that it has predictive power!

- In variable selection, we attempt to select a subset s out of the p overall predictors in a linear model

  - this will *improve model interpretability*: eliminating uninformative predictors is obviously a good thing when your goal is to tell the story of how your predictors are associated with your response

  - this can *improve prediction accuracy*: eliminating uninformative predictors can lead to lower model variance, at the expense of a slight increase in bias, leading to lower test-set mean-squared error values

- Note that variable selection is useful and/or necessary if, e.g., $n \lesssim p$ (the sample size is roughly the same as, or less than, the number of predictor variables), but can still be helpful if $n > p$

Best Subset Selection

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

(Algorithm 6.1, *Introduction to Statistical Learning* by James et al.)

- Note that:

$$\binom{p}{k} = \frac{p!}{k!(p-k)!}$$

- For multiple linear regression, BSS works for $p \lesssim 25$; for larger p, computer memory becomes an issue

Best Subset Selection: Metrics

- The functional forms of the metrics given in Step 3 are

$$C_p = \frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2)$$
$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2k\hat{\sigma}^2) = \frac{C_p}{\hat{\sigma}^2}$$
$$\text{BIC} = \frac{1}{n}(\text{RSS} + \log(n)k\hat{\sigma}^2)$$

- RSS denotes the "residual sum-of-squares"
- **KEY:** the additive terms are penalty terms that increase with k and thus act to prevent overfitting
- $\hat{\sigma}$ is an estimate of the standard deviation of the error term ϵ, i.e., the magnitude of the scatter of data around the regression line

Best Subset Selection: Metrics

- Typically, $\log(n) > 2$, so BIC (or "Bayesian Information Criterion") imposes a larger penalty relative to C_p (or "Mallow's C_p") or AIC (or "Akaike Information Criterion")
    - BIC tends to underfit (i.e., it will select as optimal those models that have *fewer* variables)
    - AIC (and C_p) tend to overfit (i.e., they will select models with *more* variables)
- Which metric you choose is up to you; the choice should be motivated by your inferential goals
    - if you use BIC, then you can be confident that every selected variable is informative, but other informative variables might have been left out of the final list
    - if you use AIC, then you can be confident that your selected variables include all the informative ones, but the final list may also include some uninformative ones as well

Forward and Backward Stepwise Selection

- What if BSS is computationally infeasible? I
  - we might use either *forward* or *backward stepwise selection*
  - for instance:

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

(Algorithm 6.2, *Introduction to Statistical Learning* by James et al.)

Forward and Backward Stepwise Selection

- In words:
  - forward stepwise selection starts with no predictor variables and adds one at a time
  - backward stepwise selection starts with the full set of predictors and takes one out at a time
- Forward and backward stepwise selection are examples of *greedy algorithms*: they make locally optimally choices that may collectively not yield a globally optimal solution
  - BSS is always to be preferred, if applying it is computationally feasible
- Note that both forward and backward stepwise selection should, when applied to a given dataset, yield similar but not necessarily identical results

Regression Example

- Below, `df` is a data frame with 3,419 rows and 17 columns

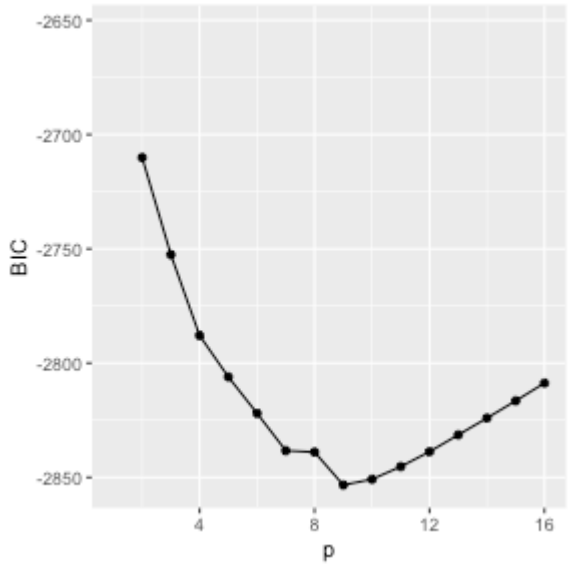
```
suppressMessages(library(bestglm))
set.seed(404)
train    <- sample(nrow(df),0.7*nrow(df)) # Perform 70-30 data splitting
df.train <- df[train,]
df.test  <- df[-train,]
lm.out   <- lm(y~.,data=df.train)         # Response variable is called "y" for bestglm
mse.full <- mean((predict(lm.out,newdata=df.test)-df.test$y)^2)
bg.out   <- bestglm(df.train,family=gaussian,IC="BIC")
bg.out$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)        mag.i       col.iJ       col.JH          J.G       J.size
##      1.4113       0.4540      -0.7420       0.4833       4.1496      -1.1839
##         H.G        H.M20          H.C       H.size
##     -3.1846       0.3445       0.2248       1.6232
```

- We observe that 9 (or 11) of 16 predictor variables are retained when using BIC (or AIC) as the penalizing criterion

Regression Example

```
library(ggplot2)
df.bg        <- data.frame(1:16,bg.out$Subsets$BIC[-1]) # the -1 gets rid of the BIC for a no-variable model
names(df.bg) <- c("p","BIC")
g            <- ggplot(data=df.bg,mapping=aes(x=p,y=BIC)) +
                  geom_point() + geom_line() + ylim(min(df.bg$BIC),min(df.bg$BIC)+200)
suppressWarnings(print(g))
```



Regression Example

- The output of `bestglm()` contains the element `BestModel`
  - `BestModel` is "[a]n lm-object representing the best fitted algorithm"
  - we can pass it to `predict()` in order to generate predicted response values

```
resp.pred <- predict(bg.out$BestModel,newdata=df.test)
mean((df.test$y-resp.pred)^2)
```

```
## [1] 0.2643661
```

```
# The saved value for the full predictor set:
mse.full
```

```
## [1] 0.2658115
```

- In this case, removing the seven least important variables improves our predictive accuracy, albeit slightly

Regression Example

- Let's repeat the analysis, but with forward-stepwise selection:

```
bg.out    <- bestglm(df.train,family=gaussian,IC="BIC",method="forward")
bg.out$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)        mag.i       col.iJ       col.JH          J.G        J.M20
##      1.4043       0.4518      -0.7446       0.4803       4.1214       0.3194
##         J.C       J.size          H.G       H.size
##      0.2168      -1.1651      -3.1460       1.5903
```

- This is a separate set of nine variables relative to the set produced by BSS

```
resp.pred <- predict(bg.out$BestModel,newdata=df.test)
mean((df.test$y-resp.pred)^2)  # Worse than for BSS or even the full set!
```

```
## [1] 0.267722
```