PSTAT 126

Regression Analysis

Laura Baracaldo

Lecture 10
Model Selection

Model Selection

We consider the model:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon$$

The purpose of this lecture is to outline methods to select predictors, and the regressors derived from them (interactions, polynomial terms), to use in a regression problem of interest.

We consider the problem of selecting the "best" set of predictors. What is the best model?

Occam's Razor Principle

We aim for the model that fits observations sufficiently well in the least complex way

Why? Because we want a balance between accuracy and precision:

- We might obtain better predictions by using more complex models so although simpler models might be appealing, we do not want to compromise on predictive accuracy.
- Unnecessary predictors will add noise to the estimation which leads to imprecise predictions. Simpler models are more precise. Additionally, removing irrelevant variables can greatly enhance interpretability.

Occam's Razor Principle

We aim for the model that fits observations sufficiently well in the least complex way

We need to define criteria to quantify two things:

- How well a model fits the data.
- Level of complexity of a model.

Variable Selection Trade-offs

- The likelihood (or SSR) provides a measure of goodness-of-fit.
- The number of parameters p provides a measure of model complexity.
- The negative likelihood (or SSR) and p are two opposite aspects of a model: the negative likelihood (or SSR) decreases as p increases
- The goal of variable selection is to find a balance between these two conflicting aspects.

The Occam's razor principle suggests that a simple model should be preferred over a more complicated one, provided they have similar goodness of fit.

Variable Selection Schemes

Least squares estimation does not perform linear model selection as it is incapable to estimate an irrelevant predictor coefficient $\hat{\beta}_j$ as exactly equal to zero. We consider two classes of schemes that drive variable selection:

1. Model Comparison: Identifying a subset of predictors among all possible predictors which are relevant when explaining the response, by using particular comparison criteria.
2. Regularization: Shrinkage of non significant coefficients towards zero by imposing some restrictions and penalties.

Model Comparison Criteria - Nested Models

To test if model M with p parameters can be reduced to a sub-model, say M_0, with $q < p$ parameters we can use the F-test with F Statistic:

$$F = \frac{(SSR_{M_0} - SSR_M)/(p-q)}{SSR_M/(n-p)} \sim F_{(p-q,n-p)}$$

If we reject H_0 we opt for model M, otherwise we opt for model M_0.

Model Comparison Criteria - Non-Nested Models

Criteria for comparing various model candidates are based on the lack of fit of the model and its complexity. Lack of fit is meassured by the SSR and complexity is measured by the number of predictors k (including the intercept).

- **Akaike Information Criterion**: For linear regression models under normality assumptions *Sakamoto et al* defined:

$$AIC = n\log(SSR/n) + 2k$$

- **Bayes Information Criterion**: Under Normality assumption, an alternative criterion was defined by *Schwarz*:

$$BIC = n\log(SSR/n) + k\log(n)$$

Model Comparison Criteria - Non-Nested Models

Both criteria provide a balance between lack of fit and complexity. Small values of AIC and BIC are preferred, so a better candidate will have a smaller SSR and smaller p^*.

Another commonly used criterion is the **Adjusted R^2 (R^2_A)**. Recall that $R^2 = 1 - SSR/SST$, therefore: Large model $\Rightarrow$ Small SSR $\Rightarrow$ Large R^2. Hence R^2 by itself is not a good criterion, because we would always choose the largest possible model, instead we use:

$$R^2_A = 1 - \frac{SSR/(n-k)}{SST/(n-1)}$$

Model Comparison Criteria - Non-Nested Models

Our final criterion is **Mallow's C_p Statistic**/ A good model should predict well, so the total mean squared error (TMSE) of prediction might be a good criterion. The $TMSE$ is written:

$$TMSE = \sum_{i=1}^{n} E(\hat{y}_i - E(y_i))^2 = \sum_{i=1}^{n} \{bias(\hat{y}_i)\}^2 + Var(\hat{y}_i)$$

Ideally we want to minimize the $TMSE$, however this is impossible since $E(y_i)$ is unknown. What we can do is to find an estimate the $TMSE$ and then use this estimate as a criterion. We define the Mallow's C_k criterion is defined as:

$$C_k = SSR_k/\sigma_p^2 + 2k - n$$

When $\sigma_{p^*}^2$ is unknown we can use $\hat{\sigma}_{p^*}^2$ which is the estimate of the error variance when using all the predictors. It can be proved that $E(C_k) = TMSE/\sigma^2$.

Model Comparison Criteria - Non-Nested Models

- When a model fits (i.e. $\hat{y}_i$ is unbiased), which implies $E(SSR_k) = (n-k)\sigma^2$, and then $E(C_k) \approx k$. A model with bad fit will have C_k much larger than k.
- The plot of k vs C_k provides a way to check for large bias ($C_k >> k$).
- We desire models with small k and with $C_k \leq k$.

States Example

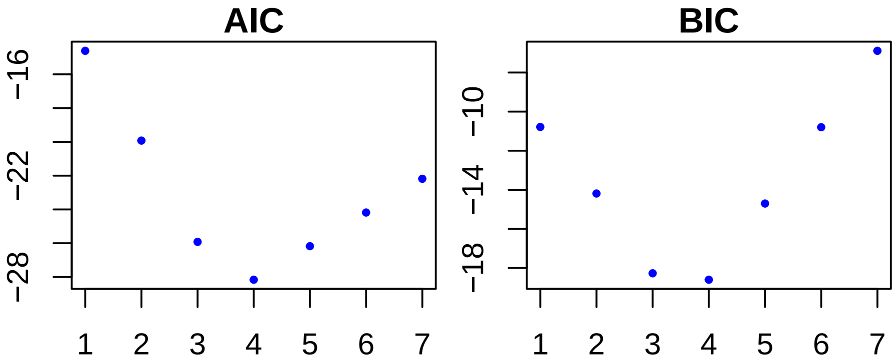
The data set comprises information on the 50 states from the 1970's collected by the U.S. Bureau of the Census. We take *Life Expectancy* as the response and the remaining variables (population, Income, Area, Illiteracy, etc) as predictors. The *leaps* package exhaustively searches all possible combination of the predictors. For each model size k it finds the variables that produce the minimum SSR:

```r
data(state)
statedata<- data.frame(state.x77, row.names =state.abb)
models<- regsubsets(Life.Exp ~ ., statedata)
rs<- summary(models)
rs$which
```

```
##   (Intercept) Population Income Illiteracy Murder HS.Grad Frost  Area
## 1        TRUE      FALSE  FALSE      FALSE   TRUE   FALSE FALSE FALSE
## 2        TRUE      FALSE  FALSE      FALSE   TRUE    TRUE FALSE FALSE
## 3        TRUE      FALSE  FALSE      FALSE   TRUE    TRUE  TRUE FALSE
## 4        TRUE       TRUE  FALSE      FALSE   TRUE    TRUE  TRUE FALSE
## 5        TRUE       TRUE   TRUE      FALSE   TRUE    TRUE  TRUE FALSE
## 6        TRUE       TRUE   TRUE       TRUE   TRUE    TRUE  TRUE FALSE
## 7        TRUE       TRUE   TRUE       TRUE   TRUE    TRUE  TRUE  TRUE
```

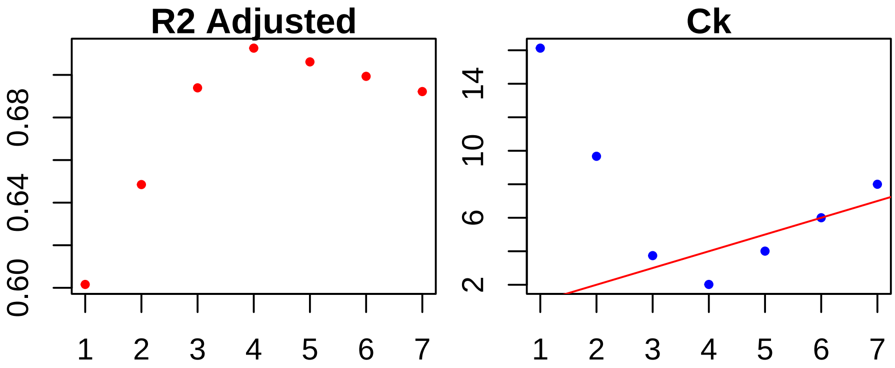
States Example - AIC & BIC

```
n<- dim(statedata)[1]
AIC<- n*log(rs$rss/n) + 2*seq(2,8,1)
BIC<- n*log(rs$rss/n) + log(n)*seq(2,8,1)
par(mar = c(2, 2, 1.2, 0.5), mfrow=c(1,2))
plot(AIC~I(1:7), main="AIC", xlab="# Predictors", pch=20, col="blue", cex=0.7)
plot(BIC~I(1:7), main="BIC", xlab="# Predictors", pch=20, col="blue", cex=0.7)
```



States Example - R^2 adjusted & C_k

```
par(mar = c(2, 2, 1, 0.5), mfrow=c(1,2))
r2Ad<- rs$adjr2
Ck<- rs$cp
plot(r2Ad~I(1:7), main="R2 Adjusted", xlab="# Predictors", pch=20, col="red", cex=0.8)
plot(Ck~I(1:7), main="Ck", xlab="# Predictors", pch=20, col="blue", cex=0.8)
abline(0,1, col="red")
```



Large p

- For p predictors, the total number of possible models is 2^p
- For small p we can compare all possible models and select the best according to some criterion.
- For large p, this is impractical. Alternative procedures should be used.

Stepwise Procedures

- **Forward Selection:**

1. Start with no variables
2. Add one predictor according to some criterion (e.g. lowest p-value$< \alpha$)
3. Stop when no variables can be added.

- **Backwards Elimination:**

1. Start with full model (all predictors).
2. Remove one predictor according to some criterion (largest p-value$> \alpha$)
3. Stop when no variables need to be dropped.

- **Stepwise Regression:** is a combination of forward addition and backward elimination. At each step a variable can be removed or added.

States Example - Backwards elimination

```
lmod<- lm(Life.Exp ~ ., statedata);summary(lmod)$coefficients
```

```
##                  Estimate   Std. Error     t value     Pr(>|t|)
## (Intercept)  7.094322e+01 1.747975e+00 40.58594017 2.510609e-35
## Population   5.180036e-05 2.918703e-05  1.77477309 8.318351e-02
## Income      -2.180424e-05 2.444256e-04 -0.08920603 9.293422e-01
## Illiteracy   3.382032e-02 3.662799e-01  0.09233464 9.268712e-01
## Murder      -3.011232e-01 4.662073e-02 -6.45899735 8.679582e-08
## HS.Grad      4.892948e-02 2.332328e-02  2.09788176 4.197175e-02
## Frost       -5.735001e-03 3.143230e-03 -1.82455682 7.518682e-02
## Area        -7.383166e-08 1.668163e-06 -0.04425927 9.649075e-01
```

```
lmod<- update(lmod ,.~. - Area);summary(lmod)$coefficients
```

```
##                  Estimate   Std. Error     t value     Pr(>|t|)
## (Intercept) 70.9893185176 1.387454e+00 51.16515405 3.694989e-40
## Population   0.0000518827 2.878768e-05  1.80225346 7.851808e-02
## Income      -0.0000244403 2.342908e-04 -0.10431609 9.174036e-01
## Illiteracy   0.0284588124 3.416329e-01  0.08330231 9.339978e-01
## Murder      -0.3018231392 4.334432e-02 -6.96338357 1.453868e-08
## HS.Grad      0.0484723220 2.066727e-02  2.34536620 2.369166e-02
## Frost       -0.0057757582 2.970228e-03 -1.94455035 5.838883e-02
```

States Example - Backwards elimination

```
lmod<- update(lmod ,.~. - Illitiracy);summary(lmod)$coefficients
```

```
##                  Estimate   Std. Error     t value     Pr(>|t|)
## (Intercept) 70.9893185176 1.387454e+00 51.16515405 3.694989e-40
## Population   0.0000518827 2.878768e-05  1.80225346 7.851808e-02
## Income      -0.0000244403 2.342908e-04 -0.10431609 9.174036e-01
## Illiteracy   0.0284588124 3.416329e-01  0.08330231 9.339978e-01
## Murder      -0.3018231392 4.334432e-02 -6.96338357 1.453868e-08
## HS.Grad      0.0484723220 2.066727e-02  2.34536620 2.369166e-02
## Frost       -0.0057757582 2.970228e-03 -1.94455035 5.838883e-02
```

```
lmod<- update(lmod ,.~. - Income);summary(lmod)$coefficients
```

```
##                  Estimate   Std. Error     t value     Pr(>|t|)
## (Intercept)  7.094960e+01 1.319111e+00 53.78591393 8.774013e-42
## Population   5.090342e-05 2.690641e-05  1.89186952 6.510101e-02
## Illiteracy   2.906222e-02 3.377228e-01  0.08605349 9.318143e-01
## Murder      -3.020008e-01 4.282127e-02 -7.05258939 9.573445e-09
## HS.Grad      4.732776e-02 1.731633e-02  2.73312927 9.000725e-03
## Frost       -5.805885e-03 2.922739e-03 -1.98645360 5.323324e-02
```

States Example - Backwards elimination

```
lmod<- update(lmod ,.~. - Population);summary(lmod)$coefficients
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 71.519958469 1.320487491 54.1617841 1.283585e-42
## Illiteracy  -0.181607775 0.327846089 -0.5539422 5.823608e-01
## Murder      -0.273117563 0.041137761 -6.6390965 3.501046e-08
## HS.Grad      0.044969709 0.017759471  2.5321536 1.489583e-02
## Frost       -0.007678224 0.002827792 -2.7152715 9.358724e-03
```

States Example - Backwards elimination

```r
lmod<-lm(Life.Exp ~ ., statedata)
step(lmod)
```

```
Start:  AIC=-22.18
Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
    Frost + Area

             Df Sum of Sq    RSS     AIC
- Area        1    0.0011 23.298 -24.182
- Income      1    0.0044 23.302 -24.175
- Illiteracy  1    0.0047 23.302 -24.174
<none>                    23.297 -22.185
- Population  1    1.7472 25.044 -20.569
- Frost       1    1.8466 25.144 -20.371
- HS.Grad     1    2.4413 25.738 -19.202
- Murder      1   23.1411 46.438  10.305

Step:  AIC=-24.18
Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
    Frost

             Df Sum of Sq    RSS     AIC
- Illiteracy  1    0.0038 23.302 -26.174
- Income      1    0.0059 23.304 -26.170
<none>                    23.298 -24.182
- Population  1    1.7599 25.058 -22.541
- Frost       1    2.0488 25.347 -21.968
- HS.Grad     1    2.9804 26.279 -20.163
- Murder      1   26.2721 49.570  11.569
```

States Example - Backwards elimination

```
Step:  AIC=-26.17
Life.Exp ~ Population + Income + Murder + HS.Grad + Frost

             Df Sum of Sq    RSS     AIC
- Income      1     0.006 23.308 -28.161
<none>                    23.302 -26.174
- Population  1     1.887 25.189 -24.280
- Frost       1     3.037 26.339 -22.048
- HS.Grad     1     3.495 26.797 -21.187
- Murder      1    34.739 58.041  17.456

Step:  AIC=-28.16
Life.Exp ~ Population + Murder + HS.Grad + Frost

             Df Sum of Sq    RSS     AIC
<none>                    23.308 -28.161
- Population  1     2.064 25.372 -25.920
- Frost       1     3.122 26.430 -23.877
- HS.Grad     1     5.112 28.420 -20.246
- Murder      1    34.816 58.124  15.528
NULL
```