

## An Illustration on Model Selection

Consider an artificial data set, which consists of  $n = 248$  independent observations on  $\{Y, X_1, X_2, X_3, X_4\}$ . Here  $Y$  and  $X_1-X_3$  are continuous while  $X_4$  is categorical with 3 levels: I, II, and III.

To identify and assess the ‘best’ model that fit the data, one may follow the following steps. First, split the data into two parts, the learning (or training) sample and the test sample, with a ratio about 1:1. Listed below are some data from the learning sample.

```
> learning
      y      x1      x2      x3  x4
1 115.7548865 10.25221911 3.09049450 3.86343133  I
2  98.6595986 15.04006617 2.76906953 2.27600358  II
3 111.2153157 12.46567294 2.53745340 3.11288934  III
4  58.3598448  5.54610509 2.55064307 2.42591879  III
5 115.1793537 16.75727865 2.42151837 2.37153843  III
.....
121 102.8861333  8.26141183 3.31631143 3.89813704  III
122  45.6598231  5.59376954 2.19834591 2.43815588   I
123 122.9809779 14.84289132 3.76720792 2.83239960  III
124  96.3418221 11.75499959 3.27246330 2.74722342   I
```

### 1 Variable Selection

First we select the best subset of variables.

#### 1.1 All Possible Regressions Procedure

To measure the performance of a multiple linear regression model,

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i,$$

where  $\varepsilon_i \sim N(0, \sigma^2)$ , Akaiki Information Criterion (AIC) is given by, up to a constant,

$$AIC = n \cdot \log(SSE) + 2 \times (k + 2).$$

Here  $k + 2$  is the total number of *parameters* in the above model including  $\sigma^2$ . A smaller AIC corresponds to a more favorable model.

Table 1: All possible regression procedures by AIC

Model	Variable Included	AIC
1	$X_1$	1035.27
2	$X_2$	1178.55
3	$X_3$	1149.46

4	$X_4$	1184.51
5	$X_1, X_2$	1026.10
6	$X_1, X_3$	902.08
7	$X_1, X_4$	1038.73
8	$X_2, X_3$	1142.29
9	$X_2, X_4$	1181.31
10	$X_3, X_4$	1153.3
<b>11</b>	$X_1, X_2, X_3$	<b>841.51</b>
12	$X_1, X_2, X_4$	1029.24
13	$X_1, X_3, X_4$	905.12
14	$X_2, X_3, X_4$	1145.57
15	$X_1, X_2, X_3, X_4$	845.34

## 1.2 Backward Deletion

**Step 1** : Start with the largest model that includes all predictors. Note that categorical variables needs special handling.

```
> options(contrasts=c("contr.treatment", "contr.poly"))
> fit <- lm(y ~ x1 + x2 + x3 + factor(x4))
> summary(fit)
```

	Value	Std. Error	t value	Pr(> t )
(Intercept)	-61.517	5.033	-12.223	0.000
x1	6.416	0.183	35.103	0.000
x2	9.529	1.092	8.730	0.000
x3	21.264	1.050	20.259	0.000
factor(x4)1	0.320	0.797	0.401	0.689
factor(x4)2	0.002	0.450	0.003	0.997

To get the overall  $p$ -value for  $X_4$ , use the general  $F$  test:

```
fit <- lm(y~x1 + x2 + x3 + factor(x4)); summary(fit)
fit1 <- lm(y~x1 + x2 + x3); anova(fit, fit1)
```

It can be found that the  $F$  test statistic is 0.0808466 and corresponds to a  $p$ -value of 0.922386. Obviously,  $X_4$  should be dropped.

**Step 2** : fit model with only  $X_1$ ,  $X_2$ , and  $X_3$  included.

```
> fit1 <- lm(y ~ x1 + x2 + x3)
> summary(fit1)
```

Coefficients:

	Value	Std. Error	t value	Pr(> t )
(Intercept)	-61.548	4.983	-12.351	0.000
x1	6.420	0.181	35.486	0.000
x2	9.488	1.069	8.874	0.000
x3	21.296	1.039	20.503	0.000

Since all the above  $p$ -values are very small ( $< .05$ ), the backward deletion procedure stops. So the best subset of predictors is  $(X_1, X_2, X_3)$ .

## 2 Interactions and Transformations

Now consider possible interactions and transformations of the selected predictors. Again, one may apply the same idea of backward deletion. The backward deletion can be done one-by-one or in a chunkwise manner.

**Step 1** : fit the model with all second-order interaction.

```
> fit.1 <- lm(y ~ (x1 + x2 + x3)^2)
> summary(fit.1)
```

Coefficients:

	Value	Std. Error	t value	Pr(> t )
(Intercept)	11.978	17.732	0.676	0.501
x1	-0.901	0.996	-0.904	0.368
x2	9.450	4.840	1.953	0.053
x3	-0.461	4.966	-0.093	0.926
x1:x2	0.109	0.219	0.500	0.618
x1:x3	2.243	0.224	10.012	0.000
x2:x3	-0.627	1.330	-0.472	0.638

**Step 2** : delete  $X_3$ .

```
> fit.2 <- lm(y ~ x1 + x2 + x1:x2 + x1:x3 + x2:x3)
> summary(fit.2)
```

Coefficients:

	Value	Std. Error	t value	Pr(> t )
(Intercept)	10.487	7.476	1.403	0.163
x1	-0.856	0.870	-0.984	0.327
x2	9.799	3.041	3.222	0.002
x1:x2	0.108	0.217	0.496	0.620
x1:x3	2.231	0.178	12.499	0.000
x2:x3	-0.734	0.654	-1.122	0.264

**Step 3** : delete  $X_1X_2$ .

```
> fit.3 <- lm(y ~ x1 + x2 + x1:x3 + x2:x3)
> summary(fit.3)
```

Coefficients:

	Value	Std. Error	t value	Pr(> t )
(Intercept)	7.028	2.704	2.600	0.011
x1	-0.529	0.565	-0.935	0.352
x2	10.906	2.061	5.291	0.000
x1:x3	2.228	0.178	12.530	0.000
x2:x3	-0.713	0.651	-1.095	0.276

**Step 4** : delete  $X_1$ .

```
> fit.4 <- lm(y ~ x2 + x1:x3 + x2:x3)
> summary(fit.4)
```

```

Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept)  6.555   2.654    2.470  0.015
           x2   9.295   1.132    8.210  0.000
          x1:x3  2.066   0.042   49.679  0.000
          x2:x3 -0.170   0.296   -0.576  0.565

```

**Step 5** : delete  $X_2X_3$ .

```

> fit.5 <- lm(y ~ x2 + x1:x3)
> summary(fit.5)

```

```

Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept)  6.838   2.601    2.629  0.010
           x2   8.821   0.775   11.380  0.000
          x1:x3  2.053   0.035   58.487  0.000

```

```

Residual standard error: 5.12 on 121 degrees of freedom
Multiple R-Squared: 0.967
F-statistic: 1780 on 2 and 121 degrees of freedom, the p-value is 0

```

Therefore, the best model is

$$y_i = \beta_0 + \beta_1 x_{i2} + \beta_2 \cdot x_{i1}x_{i3} + \varepsilon_i,$$

which explains  $R^2 = 96.7\%$  of the total variation in  $y$ .

### 3 Model Validity

To assess the model stability, apply the fitted linear fit to the test sample.

```

> R2 <- summary(fit.4)$r.squared; R2
[1] 0.967175

```

```

# The Cross Validation Correlation.

```

```

> pred <- predict(fit.4, newdata = test)
> R2.test <- (cor(test$y, pred))^2; R2.test
[1] 0.957864

```

```

# Shrinkage on Cross-Validation

```

```

> shrinkage <- R2 - R2.test; shrinkage
[1] 0.00931072

```

Since the *shrinkage on cross-validation* is only .00931 ( $< .10$ ), the model seems very reliable.