Chapter 12

Model Building

Chanwoo Yoo, Division of Advanced Engineering,
Korea National Open University

Contents

1. Variable Selection

1. Variable(Feature) Selection Methods

- Stepwise Selection
- Forward Selection
- Backward Selection
- Best Subset Selection

2. Outcome of Model Formulation 1

- **A regression model is correctly specified** if the regression equation contains all of the relevant predictors, including any necessary transformations and interaction terms. That is, there are no missing, redundant or extraneous predictors in the model.

2. Outcome of Model Formulation 1

- A correctly specified regression model yields unbiased regression coefficients and unbiased predictions of the response. And, the mean squared error (MSE) — which appears in some form in every hypothesis test we conduct or confidence interval we calculate — is an unbiased estimate of the error variance σ^2.

3. Unbiased Estimates

- An estimate is unbiased if the average of the values of the statistics determined from all possible random samples equals the parameter you're trying to estimate.

4. Outcome of Model Formulation 2

- **A regression model is underspecified** if the regression equation is missing one or more important predictor variables.
- An underspecified model yields biased regression coefficients and biased predictions of the response.

4. Outcome of Model Formulation 2

- In using the model, we would consistently underestimate or overestimate the population slopes and the population means.
- The mean square error MSE tends to overestimate σ^2, thereby yielding wider confidence intervals than it should.

5. Outcome of Model Formulation 3

- **A regression model can contain one or more extraneous variables.**
- Such a model does yield unbiased regression coefficients, unbiased predictions of the response, and an unbiased SSE.
- MSE has fewer degrees of freedom because we have more parameters in our model.
  - Confidence intervals tend to be wider and our hypothesis tests tend to have lower power.

6. Outcome of Model Formulation 4

- **A regression model is overspecified** if the regression equation contains one or more redundant predictor variables.
- Redundant predictors lead to problems such as inflated standard errors for the regression coefficients.

6. Outcome of Model Formulation 4

- Regression models that are overspecified yield unbiased regression coefficients, unbiased predictions of the response, and an unbiased SSE. Such a regression model can be used, with caution, for prediction of the response, but should not be used to describe the effect of a predictor on the response.

2. Stepwise Selection

1. Procedure

- Decide when to enter a predictor into the stepwise model.
  - Alpha-to-Enter: α_E
- Decide when to remove a predictor into the stepwise model.
  - Alpha-to-Remove: α_R

2. Step #1

- Fit each of the one-predictor models — that is, regress y on x_1, regress y on x_2 , ..., and regress y on x_{p-1}.
- Of those predictors whose t-test or F-test P-value is less than α_E, the first predictor put in the stepwise model is the predictor that has the smallest P-value.
- If no predictor has a P-value less than α_E, stop.

2. Step #2

- Suppose x_1 had the smallest P-value below α_E and therefore was deemed the "best" single predictor arising from the the first step.
- Fit each of the two-predictor models that include as a predictor — that is, regress y on x_1 and x_2, regress y on x_1 and x_3, …, and y on x_1 and x_{p-1}.

3. Step #2

- Of those predictors whose P-value is less than α_E, the second predictor put in the stepwise model is the predictor that has the smallest P-value.
- If no predictor has a P-value less than α_E, stop.
- But, suppose instead that x_2 was deemed the "best" second predictor and it is therefore entered into the stepwise model.

3. Step #2

- Now, since x_1 was the first predictor in the model, step back and see if entering x_2 into the stepwise model somehow affected the significance of the x_1 predictor. That is, check the P-value for testing $\beta_1 = 0$. If the P-value for $\beta_1 = 0$ has become not significant — that is, the P-value is greater than α_R — remove x_1 from the stepwise model.

4. Step #3

- Suppose both x_1 and x_2 made it into the two-predictor stepwise model and remained there.
- Now, fit each of the three-predictor models that include x_1 and x_2 as predictors — that is, regress y on x_1, x_2 and x_3, regress y on x_1, x_2 and x_4, …, and regress y on x_1, x_2 and x_{p-1}.

4. Step #3

- Of those predictors whose P-value is less than α_E, the third predictor put in the stepwise model is the predictor that has the smallest P-value.
- If no predictor has a P-value less than α_E, stop.
- But, suppose instead that x_3 was deemed the "best" third predictor and it is therefore entered into the stepwise model.

4. Step #3

- Now, since x_1 and x_2 was the first predictors in the model, step back and see if entering x_3 into the stepwise model somehow affected the significance of the x_1 and x_2 predictors. That is, check the P-value for testing $\beta_1 = 0$ and $\beta_2 = 0$. If the P-value for either $\beta_1 = 0$ and $\beta_2 = 0$ has become not significant — that is, the P-value is greater than α_R — remove the predictor from the stepwise model.

5. Stopping the Procedure

- Continue the steps as described above until adding an additional predictor does not yield a P-value below α_E.

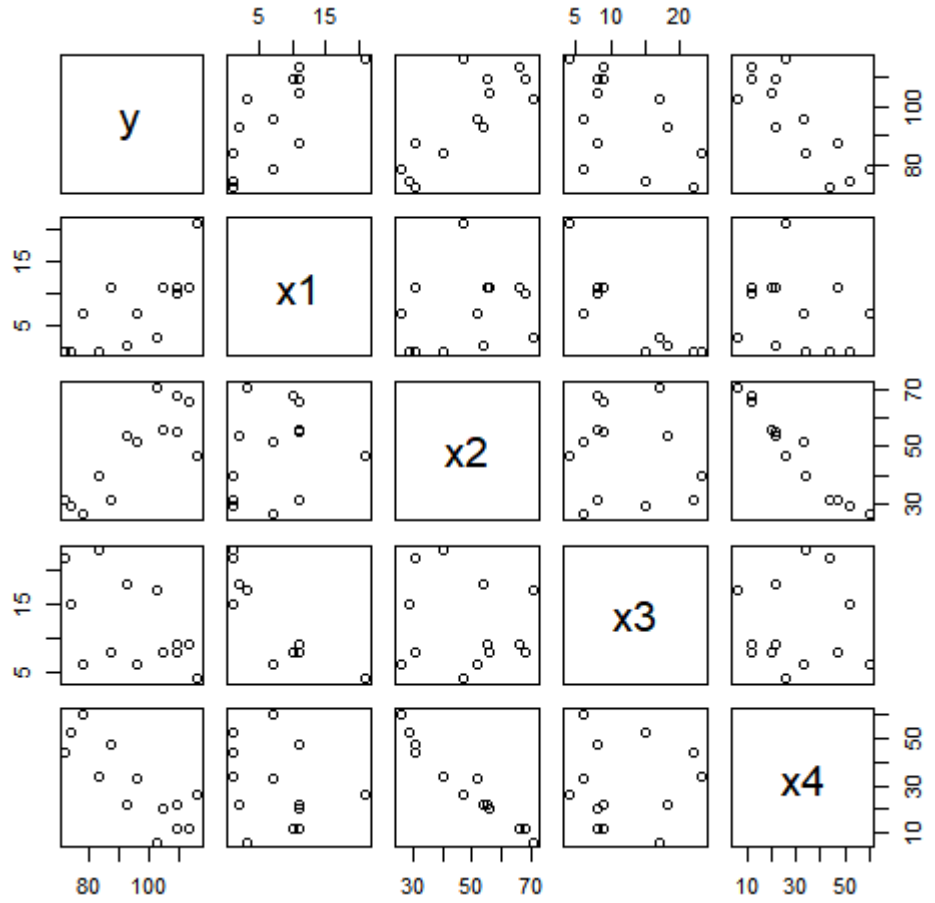
6. Data: Cement

- Data: Cement
  - y (y): heat evolved in calories during hardening of cement on a per gram basis
  - x_1 (x1): % of tricalcium aluminate
  - x_2 (x2): % of tricalcium silicate
  - x_3 (x3): % of tetracalcium alumino ferrite
  - x_4 (x4): % of dicalcium silicate

7. Data Load

```
> cement <- read.table("cement.txt", header=T)
> attach(cement)
> head(cement)
      y x1 x2 x3 x4
1  78.5  7 26  6 60
2  74.3  1 29 15 52
3 104.3 11 56  8 20
4  87.6 11 31  8 47
5  95.9  7 52  6 33
6 109.2 11 55  9 22
```

8. Scatter Plot Matrix



- It looks as if the strongest relationship exists between either y and x_2 or between y and x_4.
- A strong correlation also exists between the predictors x_2 and x_4.

9. Stepwise Selection

- Let's perform a stepwise selection with $\alpha_E = 0.15$ and $\alpha_R = 0.15$.

9. Stepwise Selection

```
> model.0 <- lm(y ~ 1)
> summary(model.0)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   95.423      4.172   22.87  2.9e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.04 on 12 degrees of freedom
```

9. Stepwise Selection

```
> add1(model.0, ~ x1 + x2 + x3 + x4, test="F")
Single term additions

Model:
y ~ 1
       Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>              2715.76 71.444
x1      1   1450.08 1265.69 63.519 12.6025 0.0045520 **
x2      1   1809.43  906.34 59.178 21.9606 0.0006648 ***
x3      1    776.36 1939.40 69.067  4.4034 0.0597623 .
x4      1   1831.90  883.87 58.852 22.7985 0.0005762 ***
```

9. Stepwise Selection

```
> model.4 <- lm(y ~ x4)
> add1(model.4, ~ . + x1 + x2 + x3, test="F")
Single term additions

Model:
y ~ x4
       Df Sum of Sq    RSS    AIC  F value    Pr(>F)
<none>              883.87 58.852
x1      1    809.10  74.76 28.742 108.2239 1.105e-06 ***
x2      1     14.99 868.88 60.629   0.1725    0.6867
x3      1    708.13 175.74 39.853  40.2946 8.375e-05 ***
```

9. Stepwise Selection

```
> model.14 <- lm(y ~ x1 + x4)
> drop1(model.14, ~ ., test="F")
Single term deletions

Model:
y ~ x1 + x4
       Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>               74.76 28.742
x1      1     809.1  883.87 58.852  108.22 1.105e-06 ***
x4      1    1190.9 1265.69 63.519  159.30 1.815e-07 ***
```

9. Stepwise Selection

```
> add1(model.14, ~ . + x2 + x3, test="F")
Single term additions

Model:
y ~ x1 + x4
       Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>               74.762 28.742
x2      1    26.789 47.973 24.974  5.0259 0.05169 .
x3      1    23.926 50.836 25.728  4.2358 0.06969 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

9. Stepwise Selection

```
> model.124 <- lm(y ~ x1 + x2 + x4)
> drop1(model.124, ~ ., test="F")
Single term deletions

Model:
y ~ x1 + x2 + x4
       Df Sum of Sq    RSS    AIC  F value    Pr(>F)
<none>               47.97 24.974
x1      1    820.91 868.88 60.629 154.0076 5.781e-07 ***
x2      1     26.79  74.76 28.742   5.0259   0.05169 .
x4      1      9.93  57.90 25.420   1.8633   0.20540
```

9. Stepwise Selection

```
> model.12 <- lm(y ~ x1 + x2)
> add1(model.12, ~ . + x3 + x4, test="F")
Single term additions

Model:
y ~ x1 + x2
       Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>              57.904 25.420
x3      1    9.7939 48.111 25.011  1.8321 0.2089
x4      1    9.9318 47.973 24.974  1.8633 0.2054
```

10. Forward Selection

- Start with a model including no variables.
- Test whether the addition of each variable improves the model significantly. If a variable significantly improves the model, include the variable in the model.
- Repeat this process until there is no improvement of the model.

11. Backward Selection

- Start with a model including all variables.
- Test whether the deletion of each variable decreases the performance of the model significantly. If a deletion of a variable does not decrease the performance of the model significantly, exclude the variable from the model.
- Repeat this process until no further variable can be excluded without significant performance loss.

3. Best Subset Selection

1. Best Subset Selection

- We select the subset of predictors that do the best at meeting some well-defined objective criterion, such as having the largest R^2 or the smallest MSE.

1. Best Subset Selection

```
install.packages("leaps")
library(leaps)
# After package install, restart R session

cement <- read.table("cement.txt", header=T)
attach(cement)
```

1. Best Subset Selection

```
subset <- regsubsets(y ~ x1 + x2 + x3 + x4,
                     method="exhaustive",
                     nbest=2,
                     data=cement)
cbind(summary(subset)$outmat,
      round(summary(subset)$adjr2, 3))
```

1. Best Subset Selection

```
> cbind(summary(subset)$outmat,
+       round(summary(subset)$adjr2, 3))
          x1  x2  x3  x4
1  ( 1 ) " " " " " " "*" "0.645"
1  ( 2 ) " " "*" " " " " "0.636"
2  ( 1 ) "*" "*" " " " " "0.974"
2  ( 2 ) "*" " " " " "*" "0.967"
3  ( 1 ) "*" "*" " " "*" "0.976"
3  ( 2 ) "*" "*" "*" " " "0.976"
4  ( 1 ) "*" "*" "*" "*" "0.974"
```

2. Model Evaluation Criteria

- Akaike's Information Criterion (AIC)
  - $AIC = n\ln(SSE) - n\ln(n) + 2p$
- Bayesian Information Criterion (BIC)
  - $BIC = n\ln(SSE) - n\ln(n) + p\ln(n)$
- Amemiya's Prediction Criterion (APC)
  - $APC = \frac{(n+p)}{n(n-p)}SSE$

2. Model Evaluation Criteria

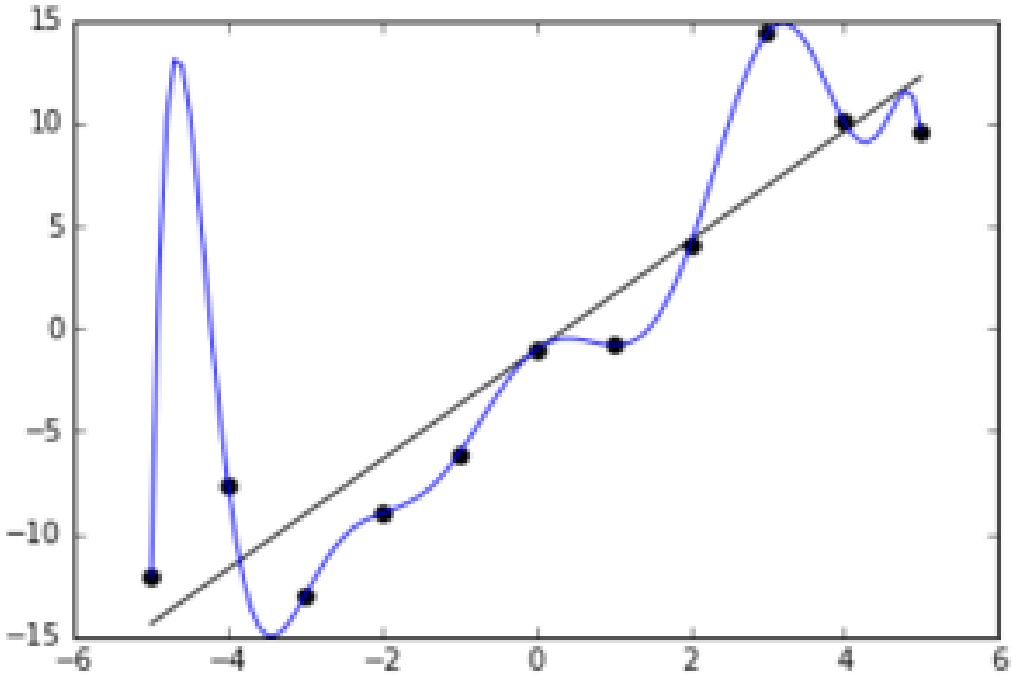
- The BIC places a higher penalty on the number of parameters in the model so will tend to reward more parsimonious (smaller) models. This stems from one criticism of AIC in that it tends to overfit models.

4. Cross Validation

1. Overfitting & Generalization

- Overfitting occurs when a model has too many parameters for the given data. The overfitted model tries to represent the given data too accurately, so it describes the errors rather than the overall trend.
- Generalization refers to the ability to make good predictions about unseen data.

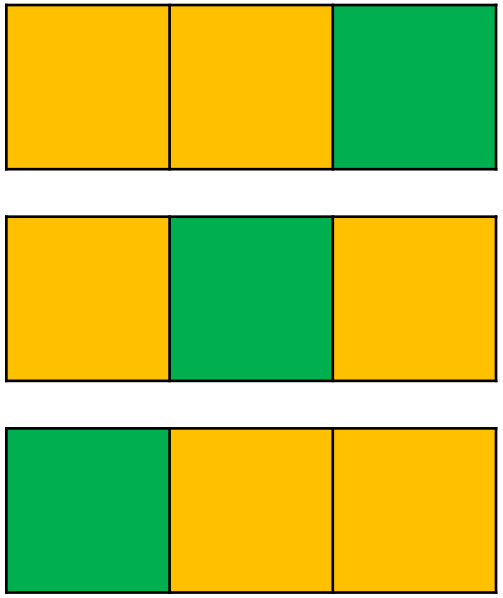
1. Overfitting & Generalization



2. Cross Validation

- Partition the sample data into a training (or model-building) set, which we can use to develop the model, and a validation (or prediction) set, which is used to evaluate the predictive ability of the model.

3. K-fold Cross Validation



$K = 3$

training set:

validation set:

3. K-fold Cross Validation

```
install.packages("caret")
library(caret)
set.seed(1)
cv <- trainControl(method="cv", number=5)
model.0 <- train(y ~ x4,
                 data=cement, trControl=cv, method='lm')
model.0$results

model.1 <- train(y ~ x1 + x2,
                   data=cement, trControl=cv, method='lm')
model.1$results
```

3. K-fold Cross Validation

```
> model.0$results
  intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD
MAESD
1      TRUE 9.702026 0.7195476 8.542292 3.027772  0.3669648
2.223049
```

- $RMSE = \sqrt{MSE}$
- $MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$

3. K-fold Cross Validation

```
> model.0$resample
       RMSE  Rsquared       MAE Resample
1  8.394758 1.0000000  8.287072    Fold1
2  4.941976 1.0000000  4.780567    Fold2
3 11.580299 0.9493671  9.724346    Fold3
4 11.814470 0.4133647  9.698464    Fold4
5 11.778626 0.2350061 10.221009    Fold5

> mean(model.0$resample$RMSE)
[1] 9.702026
```

3. K-fold Cross Validation

```
> model.0$results
  intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD
MAESD
1      TRUE 9.702026 0.7195476 8.542292 3.027772  0.3669648
2.223049

> model.1$results
  intercept     RMSE  Rsquared     MAE    RMSESD RsquaredSD
MAESD
1      TRUE 2.441475 0.7745286 2.31647 0.5695182  0.4184268
0.50983
```

4. Leave-One-Out Cross Validation

```
loocv <- trainControl(method="loocv")
model.2 <- train(y ~ x4,
                 data=cement, trControl=loocv, method='lm')
model.2$results

model.3 <- train(y ~ x1 + x2,
                 data=cement, trControl=loocv, method='lm')
model.3$results
```

4. Leave-One-Out Cross Validation

```
> model.2$resample
         RMSE Rsquared        MAE Resample
1   7.9675674       NA  7.9675674   Fold01
2   6.2680312       NA  6.2680312   Fold02
3   1.6738500       NA  1.6738500   Fold03
4   5.6451827       NA  5.6451827   Fold04
...
10 19.0830177       NA 19.0830177   Fold10
11  9.4416434       NA  9.4416434   Fold11
12  5.5521698       NA  5.5521698   Fold12
13  0.8346507       NA  0.8346507   Fold13
```

- $r^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$
- $SSTO = \sum_{i=1}^{n}(y_i - \bar{y})^2$

4. Leave-One-Out Cross Validation

```
> model.2$results
  intercept     RMSE Rsquared      MAE   RMSESD RsquaredSD
MAESD
1      TRUE 8.126361      NaN 8.126361 5.289356          NA
5.289356

> model.3$results
  intercept     RMSE Rsquared      MAE   RMSESD RsquaredSD
MAESD
1      TRUE 2.485171      NaN 2.485171 1.064331          NA
1.064331
```

Next

Chapter 13
Influential Points