3.1 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 = SSE, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p, AIC, BIC, or adjusted R^2.

3.1 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 = SSE, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p, AIC, BIC, or adjusted R^2.

3.1 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 = SSE, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p, AIC, BIC, or adjusted R^2.

3.1 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 = SSE, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p, AIC, BIC, or adjusted R^2.

3.2 Stepwise Selection: Forward

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 $k = 0, 1, \ldots, p - 1$:
   (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 C_p, AIC, BIC, or adjusted R^2.

3.2 Stepwise Selection: Forward

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 $k = 0, 1, \ldots, p - 1$:
   (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 C_p, AIC, BIC, or adjusted R^2.

3.2 Stepwise Selection: Forward

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 $k = 0, 1, \ldots, p - 1$:
   (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 C_p, AIC, BIC, or adjusted R^2.

3.2 Stepwise Selection: Forward

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 $k = 0, 1, \ldots, p-1$:
   (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 C_p, AIC, BIC, or adjusted R^2.

3.3 Stepwise Selection: Backward

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the best among these 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 C_p, AIC, BIC, or adjusted R^2.

3.3 Stepwise Selection: Backward

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the best among these 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 C_p, AIC, BIC, or adjusted R^2.

3.3 Stepwise Selection: Backward

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the best among these 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 C_p, AIC, BIC, or adjusted R^2.

3.3 Stepwise Selection: Backward

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the best among these 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 C_p, AIC, BIC, or adjusted R^2.

Data example

- 50 states data collected by U.S. Bureau of the Census
- Response: life expectancy

```r
#read data and load package
library(faraway)
data(state)
statedata <- data.frame(state.x77,row.names=state.abb)
head(statedata)
```

```
##    Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## AL       3615   3624        2.1    69.05   15.1    41.3    20  50708
## AK        365   6315        1.5    69.31   11.3    66.7   152 566432
## AZ       2212   4530        1.8    70.55    7.8    58.1    15 113417
## AR       2110   3378        1.9    70.66   10.1    39.9    65  51945
## CA      21198   5114        1.1    71.71   10.3    62.6    20 156361
## CO       2541   4884        0.7    72.06    6.8    63.9   166 103766
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

R package: leaps

```r
library(leaps)
```

regsubsets: R function for model selection

- method: exhaustive search, forward or backward stepwise

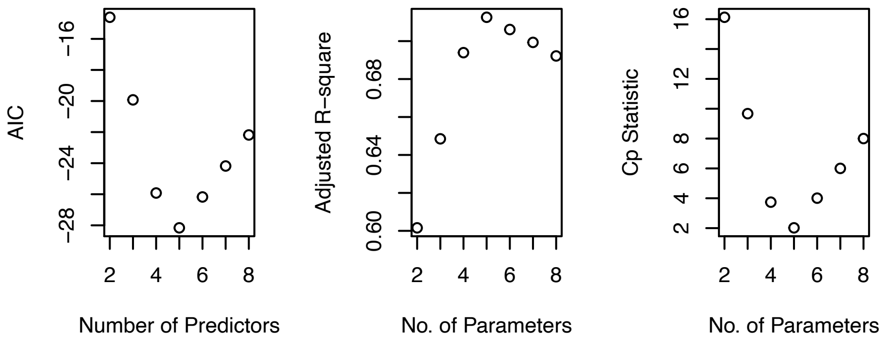
```r
b <- regsubsets(Life.Exp~.,data=statedata, method="exhaustive")
rs <- summary(b)
```

- for each size of model p, it finds the variables that produce the minumum RSS .

```r
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
```

```r
AIC <- 50*log(rs$rss/50) + (2:8)*2
par(mfrow=c(1,3))
plot(AIC ~ c(2:8), ylab="AIC", xlab="Number of Predictors")
plot(2:8, rs$adjr2, xlab="No. of Parameters", ylab="Adjusted R-square")
plot(2:8,rs$cp,xlab="No. of Parameters",ylab="Cp Statistic")
```



[MS 4] Validation and Cross-Validation

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \| Y_{\text{new}} - \hat{Y}_{\text{new}} \|^2$$

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

Prediction Error

- Arguably, the goal of a regression analysis is to "build" a model that predicts well.
- One way to measure this is in the expected prediction error of the model.
  - Estimate model parameters $\hat{\beta}$ from training data.
  - Consider future data $(X_{\text{new}}, Y_{\text{new}})$
  - Given X_{new}. Predict Y_{new} by $\hat{Y}_{\text{new}} = X_{\text{new}}\hat{\beta}$.
  - Prediction Error is

$$\text{PE} = \text{E}_{Y_{\text{new}}} \|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\mathrm{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Model Validation

Model validation refers to checking a selected model against independent data.

1. Collect new data as validation data set.
2. Split data into training and validation set.

- Estimate model by a training set.
- Evaluate Mean Squared Prediction Error by

$$\text{MSPE} = \frac{\sum_{i \in \mathcal{V}} (Y_i - \hat{Y}_i)^2}{|\mathcal{V}|}$$

  - $|\mathcal{V}|$ is the sample size of the validation data set.
  - Y_i is the i th observed response in the validation data set.
  - $\hat{Y}_i$ is the i th predicted response in the validation data set.

Leave-One-Out Cross-Validation

- Suppose we have n observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- For $i = 1, \ldots, n$
  - Fit a model with observations exclusing i-th observation.
  - Make a prediciton $\hat{y}_i$ using the fitted model.
  - Define $\mathrm{MSE}_i = (y_i - \hat{y}_i)^2$ (prediction error).
- Define LOOCV estimate

$$\mathrm{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{MSE}_i$$

Leave-One-Out Cross-Validation

- Suppose we have n observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- For $i = 1, \ldots, n$
  - Fit a model with observations exclusing i-th observation.
  - Make a prediciton $\hat{y}_i$ using the fitted model.
  - Define $\text{MSE}_i = (y_i - \hat{y}_i)^2$ (prediction error).
- Define LOOCV estimate

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i$$

Leave-One-Out Cross-Validation

- Suppose we have n observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- For $i = 1, \ldots, n$
  - Fit a model with observations exclusing i-th observation.
  - Make a prediciton $\hat{y}_i$ using the fitted model.
  - Define $\text{MSE}_i = (y_i - \hat{y}_i)^2$ (prediction error).
- Define LOOCV estimate

$$\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \text{MSE}_i$$

Leave-One-Out Cross-Validation

- Suppose we have n observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- For $i = 1, \ldots, n$
  - Fit a model with observations exclusing i-th observation.
  - Make a prediciton $\hat{y}_i$ using the fitted model.
  - Define $\text{MSE}_i = (y_i - \hat{y}_i)^2$ (prediction error).
- Define LOOCV estimate

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i$$

Leave-One-Out Cross-Validation

- Suppose we have n observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- For $i = 1, \ldots, n$
  - Fit a model with observations exclusing i-th observation.
  - Make a prediciton $\hat{y}_i$ using the fitted model.
  - Define $\text{MSE}_i = (y_i - \hat{y}_i)^2$ (prediction error).
- Define LOOCV estimate

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i$$

k-Fold Cross-Validation

- Split data randomly into K roughly equal parts.
- For $k = 1, \ldots, K$, fit the model using all but the k th part of the data and obtain predicted values $\hat{Y}_{ki}$
- Compute the prediction error mean sum of squares

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \left(Y_{ki} - \hat{Y}_{ki} \right)^2$$

- Compute a K-fold cross-validation estimate

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

k-Fold Cross-Validation

- Split data randomly into K roughly equal parts.
- For $k = 1, \ldots, K$, fit the model using all but the k th part of the data and obtain predicted values $\hat{Y}_{ki}$
- Compute the prediction error mean sum of squares

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \left(Y_{ki} - \hat{Y}_{ki} \right)^2$$

- Compute a K-fold cross-validation estimate

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

k-Fold Cross-Validation

- Split data randomly into K roughly equal parts.
- For $k = 1, \ldots, K$, fit the model using all but the k th part of the data and obtain predicted values $\hat{Y}_{ki}$
- Compute the prediction error mean sum of squares

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \left(Y_{ki} - \hat{Y}_{ki} \right)^2$$

- Compute a K-fold cross-validation estimate

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

k-Fold Cross-Validation

- Split data randomly into K roughly equal parts.
- For $k = 1, \ldots, K$, fit the model using all but the k th part of the data and obtain predicted values $\hat{Y}_{ki}$
- Compute the prediction error mean sum of squares

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \left(Y_{ki} - \hat{Y}_{ki} \right)^2$$

- Compute a K-fold cross-validation estimate

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
```

```r
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by cv.glm in the package boot.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)

cv.err$delta[1] #LOOCV estimate

## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
```

```r
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
```

```r
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
```

```r
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on LOOCV

```r
library(ISLR)
library(boot)
```

- Auto Data: Including MPG, horsepower, and other information for 392 vehicles.
- LOOCV: done by `cv.glm` in the package `boot`.

```r
glm.fit = glm(mpg ~ horsepower, data =Auto)
```

- glm gives the same fit as lm but can be input for cv.glm

```r
cv.err = cv.glm( Auto, glm.fit)
```

```r
cv.err$delta[1] #LOOCV estimate
```

```
## [1] 24.23151
```

Example on K-fold CV

- Set K option in cv.glm

```
cv.glm( Auto, glm.fit, K=10)$delta[1]

## [1] 24.14184
```

- Similar value to LOOCV.
- K-fold CV can be less computationally demanding compared to LOOCV under general models.

Example on K-fold CV

- Set K option in cv.glm

```
cv.glm( Auto, glm.fit, K=10)$delta[1]
```

```
## [1] 24.14184
```

- Similar value to LOOCV.
- K-fold CV can be less computationally demanding compared to LOOCV under general models.

Example on K-fold CV

- Set K option in cv.glm

```r
cv.glm( Auto, glm.fit, K=10)$delta[1]
```

```
## [1] 24.14184
```

- Similar value to LOOCV.
- K-fold CV can be less computationally demanding compared to LOOCV under general models.

Example on K-fold CV

- Set K option in cv.glm

```
cv.glm( Auto, glm.fit, K=10)$delta[1]
```

```
## [1] 24.14184
```

- Similar value to LOOCV.
- K-fold CV can be less computationally demanding compared to LOOCV under general models.

[MS 5] Bias-Variance Tradeoff

Decomposing PE

- Expected prediction error/ Mean squared error

$$\mathrm{MSE} = \mathrm{E}(\mathrm{PE}) = \mathrm{E}\,\|Y_{\mathrm{new}} - \hat{Y}_{\mathrm{new}}\|^2$$

- We have

$$\begin{aligned}\mathrm{MSE} &= \|\,\mathrm{E}(Y_{\mathrm{new}}) - \mathrm{E}(\hat{Y}_{\mathrm{new}})\|^2 + \mathrm{tr}\{\mathrm{var}(Y_{\mathrm{new}} - \hat{Y}_{\mathrm{new}})\} \\ &= \mathrm{bias}^2 + \mathrm{variance}\end{aligned}$$

  - $\hat{Y}_{\mathrm{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \mathrm{tr}\{\mathrm{var}(\epsilon_{\mathrm{new}}) + \mathrm{var}(\hat{Y}_{\mathrm{new}})\}$
    - $\mathrm{tr}\{\mathrm{var}(\epsilon_{\mathrm{new}})\}$ is the irreducible variane while $\mathrm{tr}\{\mathrm{var}(\hat{Y}_{\mathrm{new}})\}$ depends on model.

Decomposing PE

- Expected prediction error/ Mean squared error

$$\mathrm{MSE} = \mathrm{E}(\mathrm{PE}) = \mathrm{E}\,\|Y_{\text{new}} - \hat{Y}_{\text{new}}\|^2$$

- We have

$$\begin{aligned}\mathrm{MSE} &= \|\,\mathrm{E}(Y_{\text{new}}) - \mathrm{E}(\hat{Y}_{\text{new}})\|^2 + \mathrm{tr}\{\mathrm{var}(Y_{\text{new}} - \hat{Y}_{\text{new}})\} \\ &= \mathrm{bias}^2 + \mathrm{variance}\end{aligned}$$

  - $\hat{Y}_{\text{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \mathrm{tr}\{\mathrm{var}(\epsilon_{\text{new}}) + \mathrm{var}(\hat{Y}_{\text{new}})\}$
    - $\mathrm{tr}\{\mathrm{var}(\epsilon_{\text{new}})\}$ is the irreducible variane while $\mathrm{tr}\{\mathrm{var}(\hat{Y}_{\text{new}})\}$ depends on model.

Decomposing PE

- Expected prediction error/ Mean squared error

$$\mathrm{MSE} = \mathrm{E(PE)} = \mathrm{E}\,\| Y_{\mathrm{new}} - \hat{Y}_{\mathrm{new}} \|^2$$

- We have

$$\begin{aligned}\mathrm{MSE} &= \| \mathrm{E}(Y_{\mathrm{new}}) - \mathrm{E}(\hat{Y}_{\mathrm{new}}) \|^2 + \mathrm{tr}\{\mathrm{var}(Y_{\mathrm{new}} - \hat{Y}_{\mathrm{new}})\} \\ &= \mathrm{bias}^2 + \mathrm{variance}\end{aligned}$$

  - $\hat{Y}_{\mathrm{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \mathrm{tr}\{\mathrm{var}(\epsilon_{\mathrm{new}}) + \mathrm{var}(\hat{Y}_{\mathrm{new}})\}$
    - $\mathrm{tr}\{\mathrm{var}(\epsilon_{\mathrm{new}})\}$ is the irreducible variane while $\mathrm{tr}\{\mathrm{var}(\hat{Y}_{\mathrm{new}})\}$ depends on model.

Decomposing PE

- Expected prediction error/ Mean squared error

$$\mathrm{MSE} = \mathrm{E}(\mathrm{PE}) = \mathrm{E}\,\| Y_{\text{new}} - \hat{Y}_{\text{new}} \|^2$$

- We have

$$\begin{aligned} \mathrm{MSE} &= \| \mathrm{E}(Y_{\text{new}}) - \mathrm{E}(\hat{Y}_{\text{new}}) \|^2 + \mathrm{tr}\{\mathrm{var}(Y_{\text{new}} - \hat{Y}_{\text{new}})\} \\ &= \text{bias}^2 + \text{variance} \end{aligned}$$

  - $\hat{Y}_{\text{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \mathrm{tr}\{\mathrm{var}(\epsilon_{\text{new}}) + \mathrm{var}(\hat{Y}_{\text{new}})\}$
    - $\mathrm{tr}\{\mathrm{var}(\epsilon_{\text{new}})\}$ is the irreducible variane while $\mathrm{tr}\{\mathrm{var}(\hat{Y}_{\text{new}})\}$ depends on model.

Decomposing PE

- Expected prediction error/ Mean squared error

$$\text{MSE} = \text{E}(\text{PE}) = \text{E}\,\| Y_{\text{new}} - \hat{Y}_{\text{new}} \|^2$$

- We have

$$\begin{aligned} \text{MSE} &= \| \text{E}(Y_{\text{new}}) - \text{E}(\hat{Y}_{\text{new}}) \|^2 + \text{tr}\{\text{var}(Y_{\text{new}} - \hat{Y}_{\text{new}})\} \\ &= \text{bias}^2 + \text{variance} \end{aligned}$$

  - $\hat{Y}_{\text{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \text{tr}\{\text{var}(\epsilon_{\text{new}}) + \text{var}(\hat{Y}_{\text{new}})\}$
    - $\text{tr}\{\text{var}(\epsilon_{\text{new}})\}$ is the irreducible variane while $\text{tr}\{\text{var}(\hat{Y}_{\text{new}})\}$ depends on model.

Decomposing PE

- Expected prediction error/ Mean squared error

$$\text{MSE} = \text{E(PE)} = \text{E}\,\| Y_{\text{new}} - \hat{Y}_{\text{new}} \|^2$$

- We have

$$\begin{aligned}\text{MSE} &= \| \text{E}(Y_{\text{new}}) - \text{E}(\hat{Y}_{\text{new}}) \|^2 + \text{tr}\{\text{var}(Y_{\text{new}} - \hat{Y}_{\text{new}})\} \\ &= \text{bias}^2 + \text{variance}\end{aligned}$$

  - $\hat{Y}_{\text{new}}$ is from old (training) data.
  - Y_{new} is from new data.
    - When independent, variance $= \text{tr}\{\text{var}(\epsilon_{\text{new}}) + \text{var}(\hat{Y}_{\text{new}})\}$
    - $\text{tr}\{\text{var}(\epsilon_{\text{new}})\}$ is the irreducible variane while $\text{tr}\{\text{var}(\hat{Y}_{\text{new}})\}$ depends on model.