STA 314: Statistical Methods for Machine Learning I

Lecture 4 - Cross-validation and subset selection under linear models

Xin Bing

Department of Statistical Sciences
University of Toronto

Review

Example

$$\text{Model 1:} \qquad Y = \alpha_0 + \alpha_1 X_1 + \epsilon$$
$$\text{Model 2:} \qquad Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

lead to different predictors at $X = x$

$$\hat{y}_1 = \hat{\alpha}_0 + \hat{\alpha}_1 x_1 \qquad \text{v.s.} \qquad \hat{y}_2 = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2.$$

- When we have $\mathcal{D}_{test}$, we compare the test MSE errors

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_i^{(T)} - \hat{\alpha}_0 - \hat{\alpha}_1 x_{i1}^{(T)}\right)^2$$

$$\text{v.s.} \quad \frac{1}{m}\sum_{i=1}^{m}\left(y_i^{(T)} - \hat{\beta}_0 - \hat{\beta}_1 x_{i1}^{(T)} - \hat{\beta}_2 x_{i2}^{(T)}\right)^2$$

Review

When we don't have $\mathcal{D}_{test}$, there are two common approaches for model selection

- We can avoid estimating the expected MSE by making an adjustment to the training error to account for the model complexity:
  - Mallow's C_p
  - AIC
  - BIC
  - adjusted R^2

- We can directly estimate the expected MSE by manually creating a "test set" using data-splitting techniques:
  - validation set approach
  - cross-validation approach

Direct estimation of the expected MSE via data-splitting techniques

We *randomly* split the available data to create a validation set that functions as a test set.

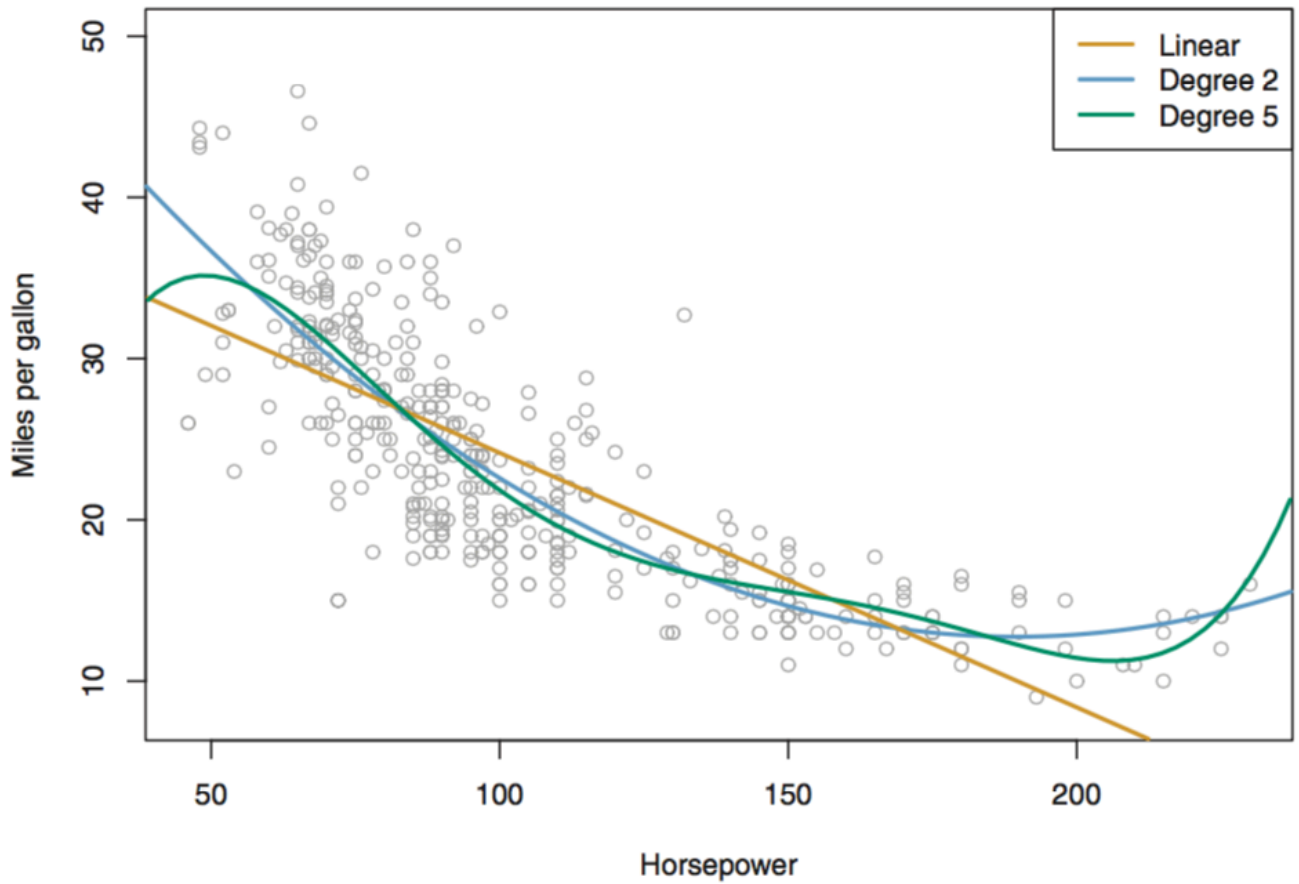
- Validation set approach: one-time data splitting
- Cross-validation approach: multiple-time data splitting

Validation set approach

- *Randomly* divide the available set of samples into two parts: a **training set** and a **validation** or **hold-out** set.
  - What is the proportion? Depends.
- The model is fit on the training set, and the fitted model is used to predict the response for the observations in the validation set.
- The resulting validation-set MSE provides an estimate of the expected MSE.

Example: Auto Data

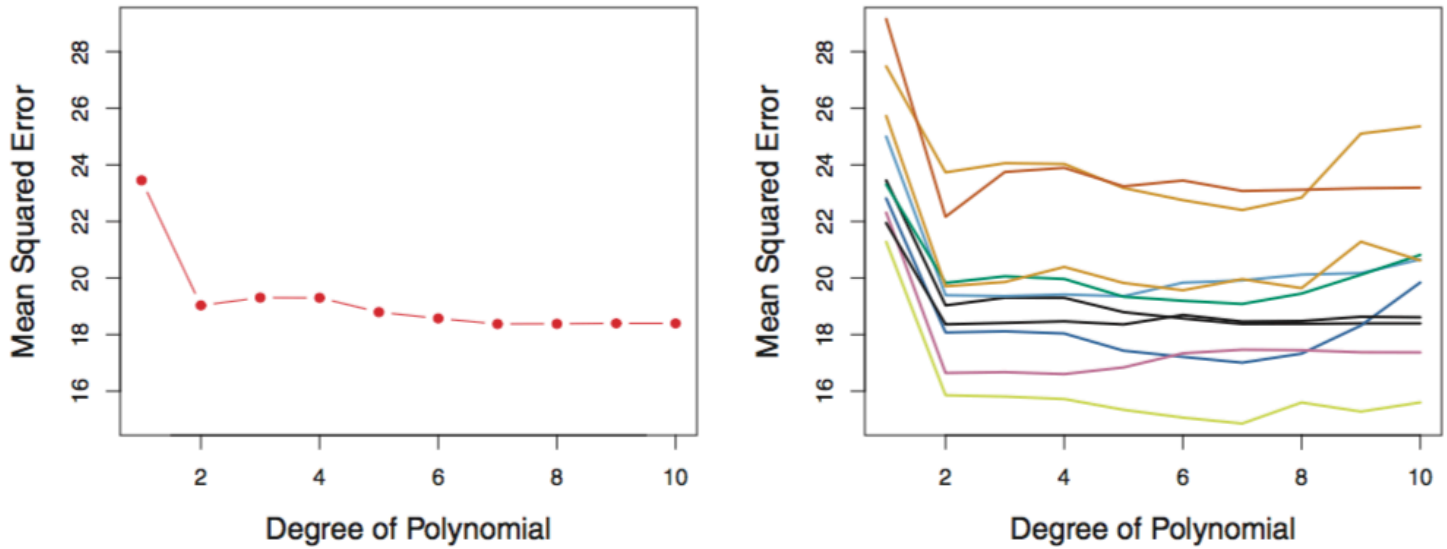
In Chapter 3, we find there appears to be a non-linear relationship between **mpg** and **horsepower**.



Whether a cubic or higher-order fit might provide a better fit?

Example: Auto Data – Compare linear vs higher-order polynomial terms in a linear regression.

We randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 data.



- Left: Validation error estimates for a single split into training and validation data sets.
- Right: Validation method repeated 10 times with each time using a different random split of the observations into a training set and a validation set.
- We can see the one-time data splitting is not stable

Drawbacks of Validation Set Approach

- The validation estimate of the test error can be highly unstable, depending on which observations are included in the training set and which are in the validation set.
- Only a subset of the observations – those in the training set rather than in the validation set – are used to fit the model.
  The resulting estimate or classifier is worse!
- How to remedy these drawbacks?

Leave-One-Out Cross-Validation (LOOCV)

- First split the data into two parts by leaving out the **first** observation:
  - a validation set: (x_1, y_1)
  - a training set: the remaining observations $(x_2, y_2), ..., (x_n, y_n)$
  - using the training set, we fit the model and predict y_1 as $\hat{y}_1$ using the value x_1. The test error could be approximated by

$$MSE_1 = (y_1 - \hat{y}_1)^2.$$

  - not good enough!

Leave-One-Out Cross-Validation

- Repeat the procedure by leaving out the **second** observation:
  - a validation set: (x_2, y_2),
  - a training set: the remaining observations $(x_1, y_1), (x_3, y_3), ..., (x_n, y_n)$
  - using the training set, we fit the model and predict y_2 as $\hat{y}_2$ using the value x_2. Compute

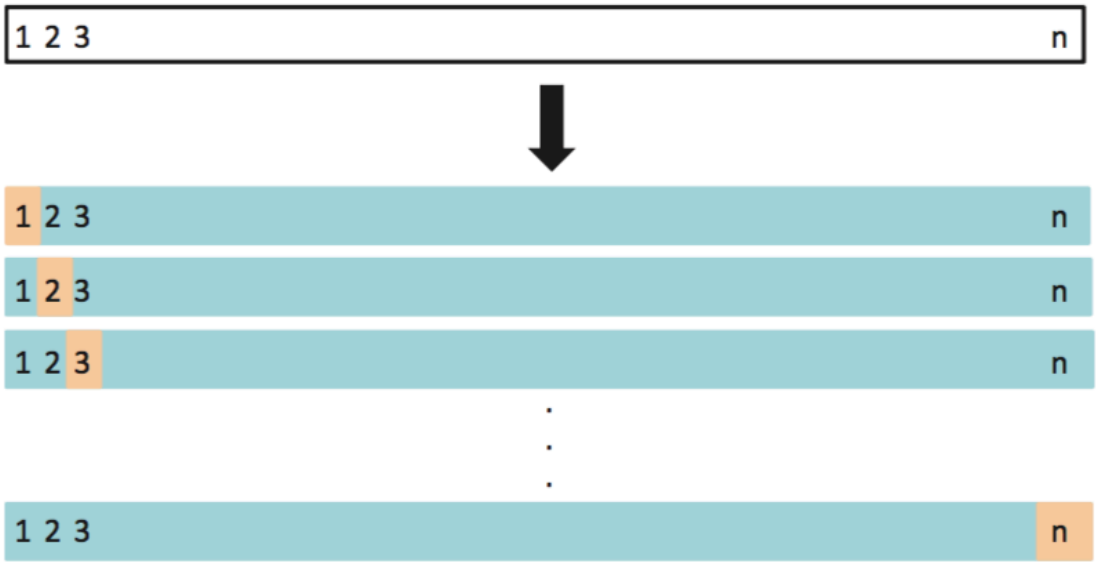
$$MSE_2 = (y_2 - \hat{y}_2)^2.$$

- Repeating the approach n times by leaving out **each** observation to obtain $MSE_1, ..., MSE_n$.

- The LOOCV estimate for the test MSE is the average of these n test error estimates:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} MSE_i.$$

Leave-One-Out Cross-Validation



Validation data sets in beige, and training sets in blue.

LOOCV vs Validation Set Approach

LOOCV has the following advantage over the validation set approach.

- The training set of LOOCV is almost the same as the entire data set. The fitted model is almost as good as that based on the entire data set.
- The validation approach yields different results when applied repeatedly, because the training/validation set is randomly divided. LOOCV has no randomness in the splitting.

However, LOOCV can be computationally expensive. (In linear model, the computation can be simplified, the formula is shown in page 180 of the textbook).

k-Fold Cross-Validation

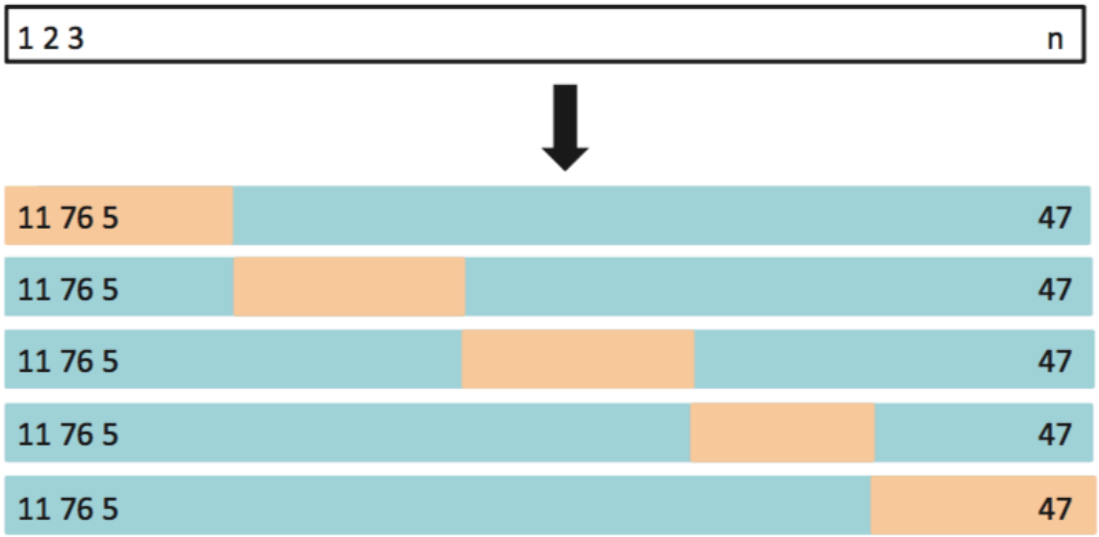
- **k-fold CV** is to randomly divide the data into k (roughly) equal-sized groups or folds.
- The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds. We compute the mean squared error, MSE_1, for the observations in the first fold.
- Then we repeat the procedure to fold 2, fold 3,..., fold k, and get MSE_2, MSE_3,...,MSE_k.
- The k-fold CV estimate is computed by averaging these values,

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i.$$

Remark

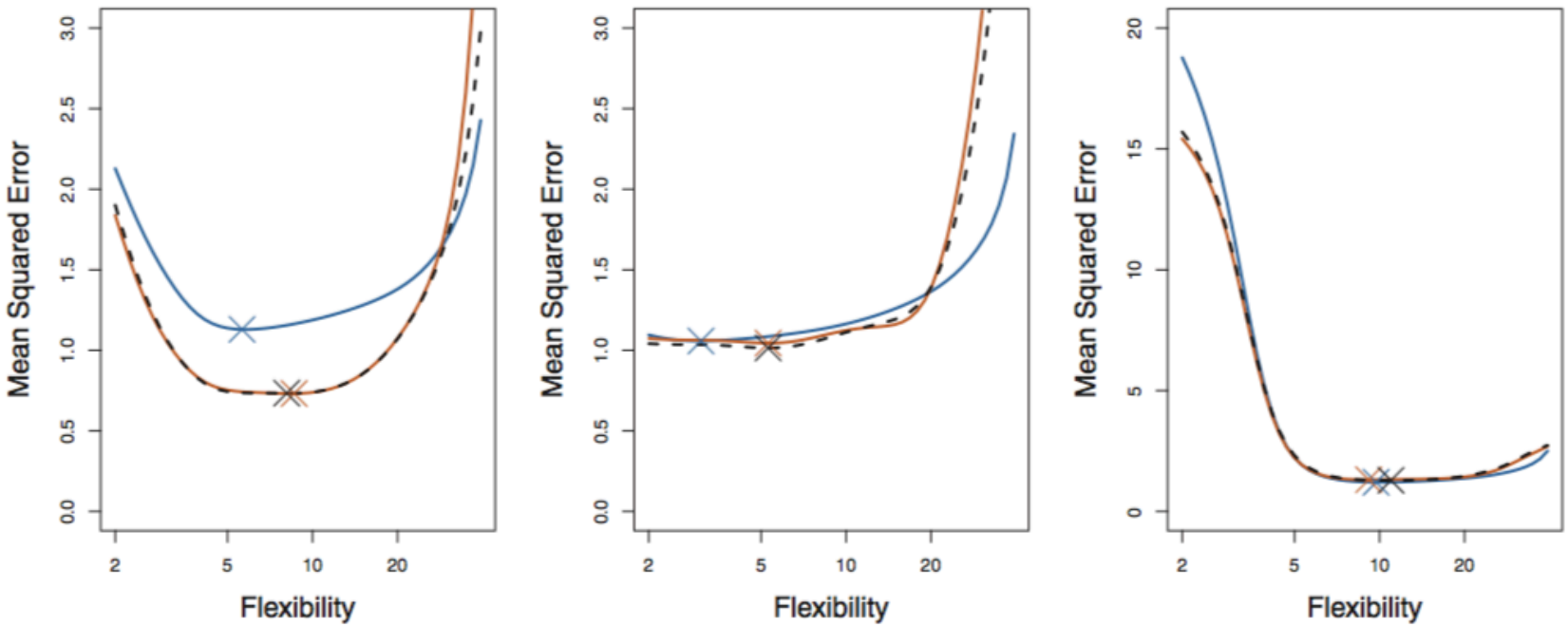
- LOOCV is a special case of n-fold CV.
- 5-fold or 10-fold is commonly used in practice.

k-Fold Cross-Validation



Validation data sets in orange, and training sets in blue.

k-Fold Cross-Validation



True test MSE (in blue), the LOOCV estimate (black dashed line), and the 10-fold CV estimate (in orange) for three simulated data sets.

Cross-Validation on Classification Problems

- Cross-validation also works for classification problems.
- For LOOCV, we split the data in the same way as before. We compute the error on the validation set, $Err_1 = 1\{y_1 \neq \hat{y}_1\}$.
- Then we repeat the procedure n times, and get Err_2, Err_3,...,Err_n.
- The LOOCV estimate is computed by averaging these values,

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_n.$$

Cross-Validation is sometimes tricky!

Independence between the fitted model and the validation set is the key!

Example

Consider a simple two-step approach applied to some data $\mathcal{D}^{train}$.

- Starting with 5000 predictors and 100 samples, find the 10 predictors having the largest correlation with the outcome.
- We then apply the OLS using only these 10 predictors.

How do we estimate the expected MSE of the fitted model from this approach?

Discussion / recommendation on these two approaches

- The data-splitting technique has two advantages relative to AIC, BIC, C_p, and adjusted R^2:
  - it provides a direct estimate of the test error
  - It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g. the number of predictors in the model) or hard to estimate the error variance.

- The data-splitting technique also has a couple of drawbacks comparing to the other approach:
  - it requires a relatively large sample size
  - it is difficult to have guarantees for the model selected by using CV.
  - when the distribution is specified and the error of variance can be consistently estimated, the first approach is preferred.

Application to the model selection in linear models

Recall that we have the following alternatives to the OLS using all predictors:

- **Subset Selection**. We identify a subset of the p predictors that we believe to be related to the response. We then fit a model using the OLS approach on the identified set of predictors.
  - **Best Subset Selection**
  - **Stepwise Selection**
- **Shrinkage**. Next lecture.
- **Dimension Reduction**. Later after PCA.

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.

For best subset selection, we need to fit and compare

$$\binom{p}{0} + \binom{p}{1} + \binom{p}{2} + \cdots + \binom{p}{p} = 2^p$$

models.

Best Subset Selection

- Step 2 identifies the best model (on the training data) for each subset size. After this step, we have reduced the problem of choosing one from 2^p possible models to that of choosing one from $p + 1$ possible models.
- In Step 3, we should not use RSS or R^2, because we want a model with small test error rather than small training error.
- The same approach can be used for other types of models, such as logistic regression (RSS replaced by deviance).

Forward Stepwise Selection

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.

For forward stepwise selection, in the kth iteration, we fit and compare $(p - k)$ models. So, in total we choose one from

$$1 + \sum_{k=0}^{p-1} (p - k) = 1 + \frac{p(p+1)}{2}$$

models, *much fewer* than 2^p models.

Forward Stepwise Selection

- It has computational advantage over best subset selection.
- It can be used in high-dimensional setting with $n < p$.
  - Why the best subset selection cannot be used for $n < p$?
- It is a greedy procedure!
  So not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.

The Credit Card Data

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income`, `student`, `limit`	`rating`, `income`, `student`, `limit`

Backward Stepwise Selection

Algorithm 6.3 *Backward stepwise selection*

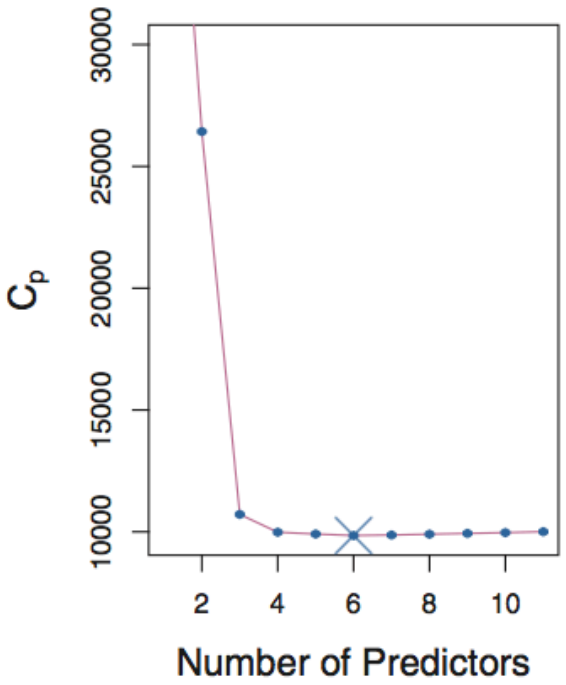
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$.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

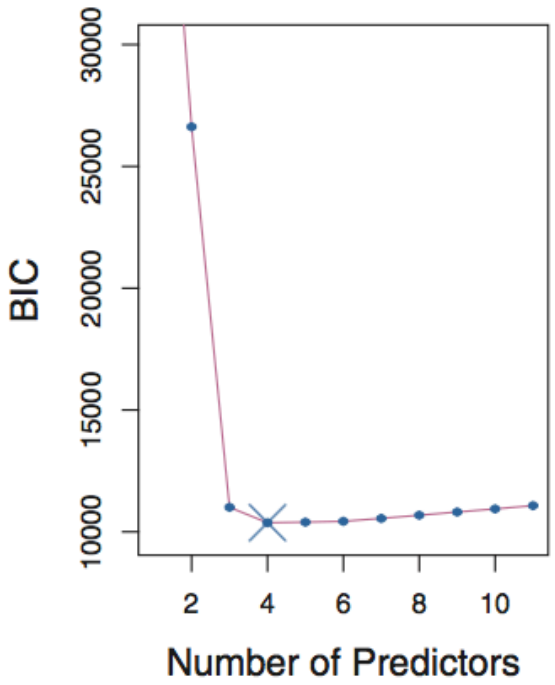
- For backward stepwise selection, we also compare $1 + p(p+1)/2$ models, much fewer than 2^p models.
- It only works when $n > p$. (Why?)
- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.

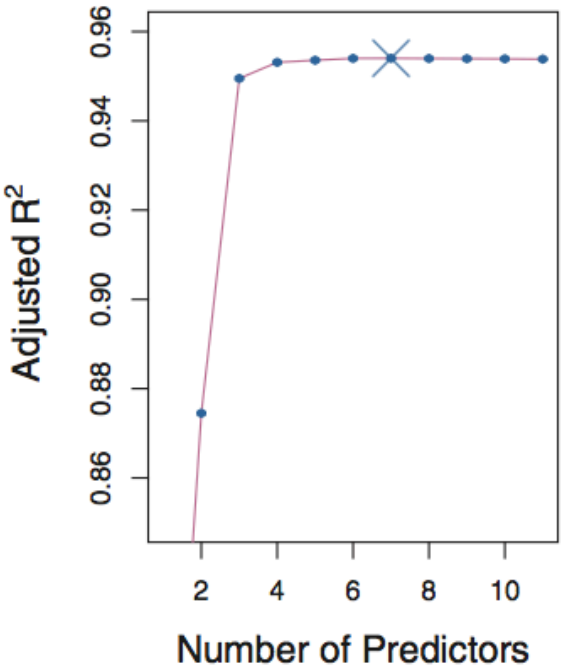
Choosing the Optimal Model

- The model containing all of the predictors will always have the smallest RSS and the largest R^2, since these quantities are related to the training error.
- We wish to choose a model with low test error, not a model with low training error. Recall that training error is usually a poor estimate of test error.
- Therefore, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors.
- Instead, we should use C_p (AIC), BIC and adjusted R^2.

The Credit Card Data



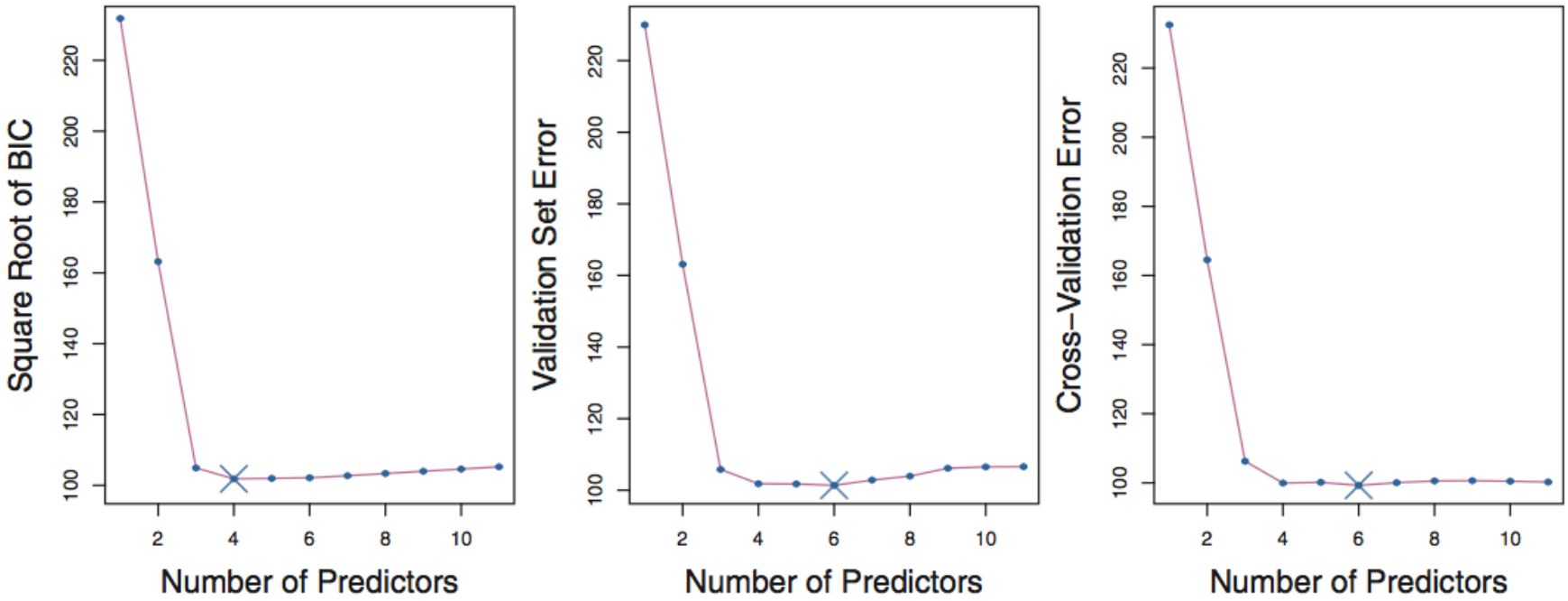




What about the validation set approach and cross-validation?

- We compute the validation set error or the cross-validation error for each model M_k, $k = 1, 2, \ldots,$ under consideration
- Then we select the model for which the computed error is the smallest.

The Credit Card Data



Practical recommendation

- The cross-validation approach is generally applicable to all supervised learning problems.
  - It typically requires at least a moderate sample size, relative to the model complexity.
  - When the sample size is large, it is safe to stick to the CV approach.

- The AIC/BIC approach is suitable when the likelihood is specified (mainly for parametric approach).
  - In such case, its performance could be better than the CV approach when the sample size is limited. In practical problems, we often find that the selected models by both approaches are close.