Lecture 13: Model selection and regularization

Reading: Sections 6.1-6.2.1

STATS 202: Data mining and analysis

Jonathan Taylor, 10/22
Slide credits: Sergio Bacallado

What do we know so far

- In linear regression, adding predictors always decreases the training error or RSS.

What do we know so far

- In linear regression, adding predictors always decreases the training error or RSS.
- However, adding predictors does not necessarily improve the test error.

What do we know so far

- In linear regression, adding predictors always decreases the training error or RSS.
- However, adding predictors does not necessarily improve the test error.
- Selecting significant predictors is hard when n is not much larger than p.

What do we know so far

- In linear regression, adding predictors always decreases the training error or RSS.
- However, adding predictors does not necessarily improve the test error.
- Selecting significant predictors is hard when n is not much larger than p.
- When $n < p$, there is no least squares solution:

$$\hat{\beta} = \underbrace{(\mathbf{X}^T\mathbf{X})}_{\text{Singular}}{}^{-1}\mathbf{X}^T y.$$

So, we must find a way to select fewer predictors.

Best subset selection

- Simple idea: let's compare all models with k predictors.

Best subset selection

- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.

Best subset selection

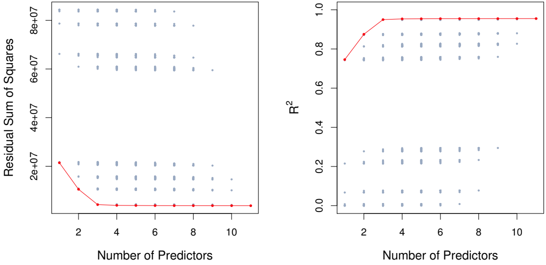
- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.
- Choose the model with the smallest RSS.

Best subset selection

- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.
- Choose the model with the smallest RSS. Do this for every possible k.

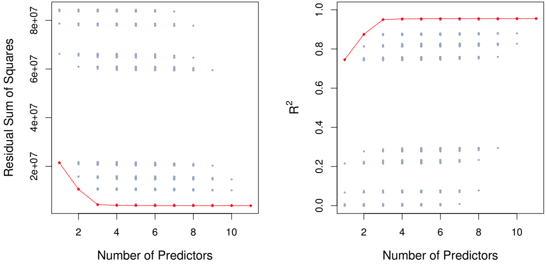
Best subset selection

- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.
- Choose the model with the smallest RSS. Do this for every possible k.



Best subset selection

- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.
- Choose the model with the smallest RSS. Do this for every possible k.



- Naturally, the RSS and R^2 improve as we increase k.

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**,

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. **Akaike Information Criterion (AIC):**

$$\frac{1}{n\hat{\sigma}^2}(\mathsf{RSS} + 2k\hat{\sigma}^2)$$

   where $\hat{\sigma}^2$ is an estimate of the irreducible error.

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. **Akaike Information Criterion (AIC) or C_p:**

$$\frac{1}{n}(\mathsf{RSS} + 2k\hat{\sigma}^2)$$

   where $\hat{\sigma}^2$ is an estimate of the irreducible error.

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. **Akaike Information Criterion (AIC) or C_p:**
2. **Bayesian Information Criterion (BIC):**

$$\frac{1}{n}(\mathsf{RSS} + \log(n)k\hat{\sigma}^2)$$

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. **Akaike Information Criterion (AIC) or C_p:**
2. **Bayesian Information Criterion (BIC):**
3. **Adjusted R^2:**

$$R^2 = 1 - \frac{\mathsf{RSS}}{\mathsf{TSS}}$$

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. **Akaike Information Criterion (AIC) or C_p:**
2. **Bayesian Information Criterion (BIC):**
3. **Adjusted R^2:**

$$R^2_{\text{adj}} = 1 - \frac{\text{RSS}/(n-k-1)}{\text{TSS}/(n-1)}$$

Best subset selection

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

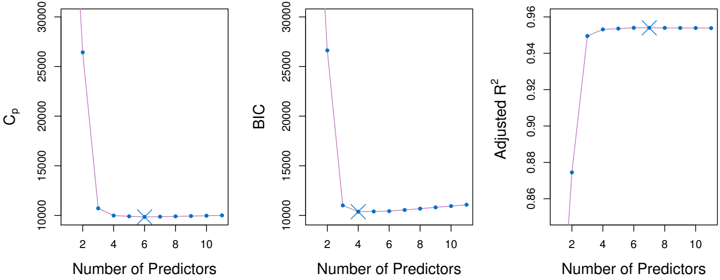
1. **Akaike Information Criterion (AIC) or C_p:**
2. **Bayesian Information Criterion (BIC):**
3. **Adjusted R^2:**

How do they compare to cross validation:

- They are much less expensive to compute.
- They are motivated by asymptotic arguments and rely on model assumptions (eg. normality of the errors).
- Equivalent concepts for other models (e.g. logistic regression).

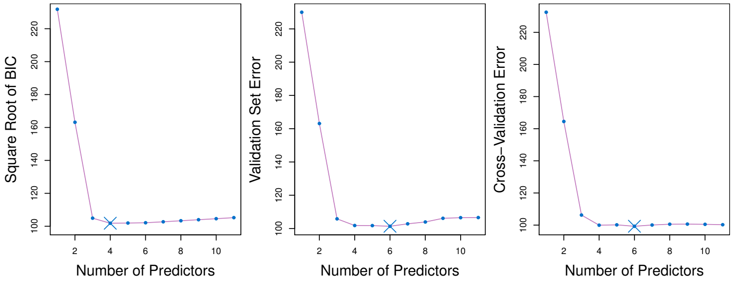
Example

Best subset selection for the Credit dataset.



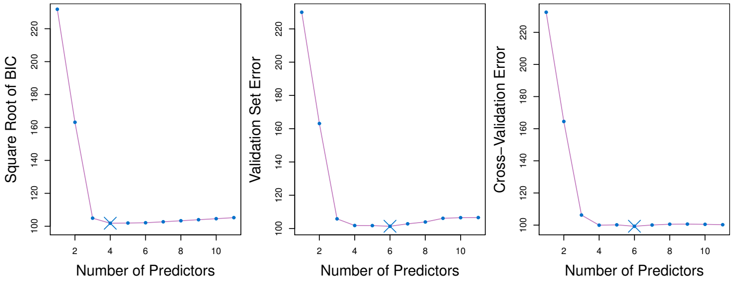
Example

Cross-validation vs. the BIC.



Example

Cross-validation vs. the BIC.



Recall: In k-fold cross validation, we can estimate a standard error or accuracy for our test error estimate. Then, we apply the **one standard-error rule**.

Stepwise selection methods

Best subset selection has 2 problems:

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting.

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting. The model selected has *high variance*.

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting. The model selected has *high variance*.

In order to mitigate these problems, we can restrict our search space for the best model.

This reduces the variance of the selected model at the expense of an increase in bias.

Forward 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.

Forward selection vs. best subset

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

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* Credit *data set. The first three models are identical but the fourth models differ.*

Backward 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.

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.
- Although it seems like they should, they need not produce the same sequence of models.

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.
- Although it seems like they should, they need not produce the same sequence of models.

  Example. $X_1, X_2 \sim \mathcal{N}(0, \sigma)$ independent.

  $$X_3 = X_1 + 3X_2$$
  $$Y = X_1 + 2X_2 + \epsilon$$

  Regress Y onto X_1, X_2, X_3.

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.
- Although it seems like they should, they need not produce the same sequence of models.

  Example. $X_1, X_2 \sim \mathcal{N}(0, \sigma)$ independent.

  $$X_3 = X_1 + 3X_2$$
  $$Y = X_1 + 2X_2 + \epsilon$$

  Regress Y onto X_1, X_2, X_3.

  - Forward: $\{X_3\} \to \{X_3, X_2\} \to \{X_3, X_2, X_1\}$

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.
- Although it seems like they should, they need not produce the same sequence of models.

  Example. $X_1, X_2 \sim \mathcal{N}(0, \sigma)$ independent.

  $$\begin{aligned} X_3 &= X_1 + 3X_2 \\ Y &= X_1 + 2X_2 + \epsilon \end{aligned}$$

  Regress Y onto X_1, X_2, X_3.

  - Forward: $\{X_3\} \to \{X_3, X_2\} \to \{X_3, X_2, X_1\}$
  - Backward: $\{X_1, X_2, X_3\} \to \{X_1, X_2\} \to \{X_2\}$

Other stepwise selection strategies

- **Mixed stepwise selection:** Do forward selection, but at every step, remove any variables that are no longer "necessary".

Other stepwise selection strategies

- **Mixed stepwise selection:** Do forward selection, but at every step, remove any variables that are no longer "necessary".
- **Forward stagewise selection:** ...

Other stepwise selection strategies

- **Mixed stepwise selection:** Do forward selection, but at every step, remove any variables that are no longer "necessary".
- **Forward stagewise selection:** ...
- ...

Shrinkage methods

A mainstay of modern statistics

The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.

Shrinkage methods

A mainstay of modern statistics

The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.

Why would shrunk coefficients be better?

Shrinkage methods

A mainstay of modern statistics

The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.

Why would shrunk coefficients be better?

- This introduces *bias*, but may significantly decrease the *variance* of the estimates. If the latter effect is larger, this would decrease the test error.

Shrinkage methods

A mainstay of modern statistics

The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.

Why would shrunk coefficients be better?

- This introduces *bias*, but may significantly decrease the *variance* of the estimates. If the latter effect is larger, this would decrease the test error.
- Extreme example: set $\hat{\beta}$ to 0 – variance is 0!

Shrinkage methods

A mainstay of modern statistics

The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.

Why would shrunk coefficients be better?

- This introduces *bias*, but may significantly decrease the *variance* of the estimates. If the latter effect is larger, this would decrease the test error.
- Extreme example: set $\hat{\beta}$ to 0 – variance is 0!
- There are Bayesian motivations to do this: the prior tends to shrink the parameters.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β, or $\|\beta\|_2^2$.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β, or $\|\beta\|_2^2$.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β, or $\|\beta\|_2^2$.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

We find an estimate $\hat{\beta}_\lambda^R$ for many values of λ and then choose it by cross-validation.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β, or $\|\beta\|_2^2$.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

We find an estimate $\hat{\beta}^R_\lambda$ for many values of λ and then choose it by cross-validation. Fortunately, this is no more expensive than running a least-squares regression.

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c, ie. after doing this we have the same RSS.

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c, ie. after doing this we have the same RSS.

In ridge regression, this is not true.

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c, ie. after doing this we have the same RSS.

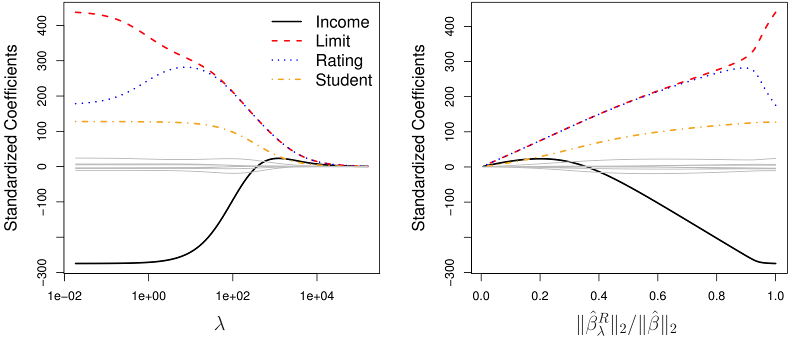
In ridge regression, this is not true.

In practice, what do we do?

- Scale each variable such that it has sample variance 1 before running the regression.
- This prevents penalizing some coefficients more than others.

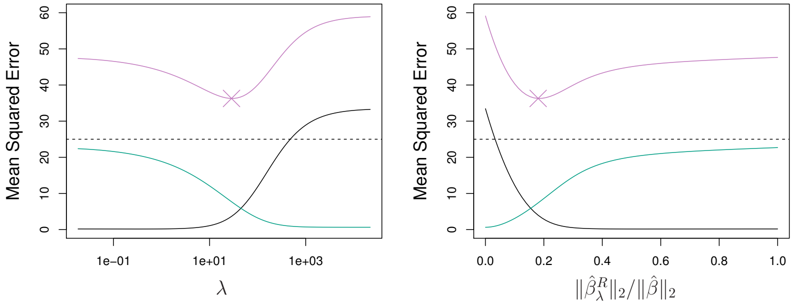
Example. Ridge regression

Ridge regression of `default` in the Credit dataset.



Bias-variance tradeoff

In a simulation study, we compute bias, variance, and test error as a function of λ.



Cross validation would yield an estimate of the test error.

Selecting λ by cross-validation