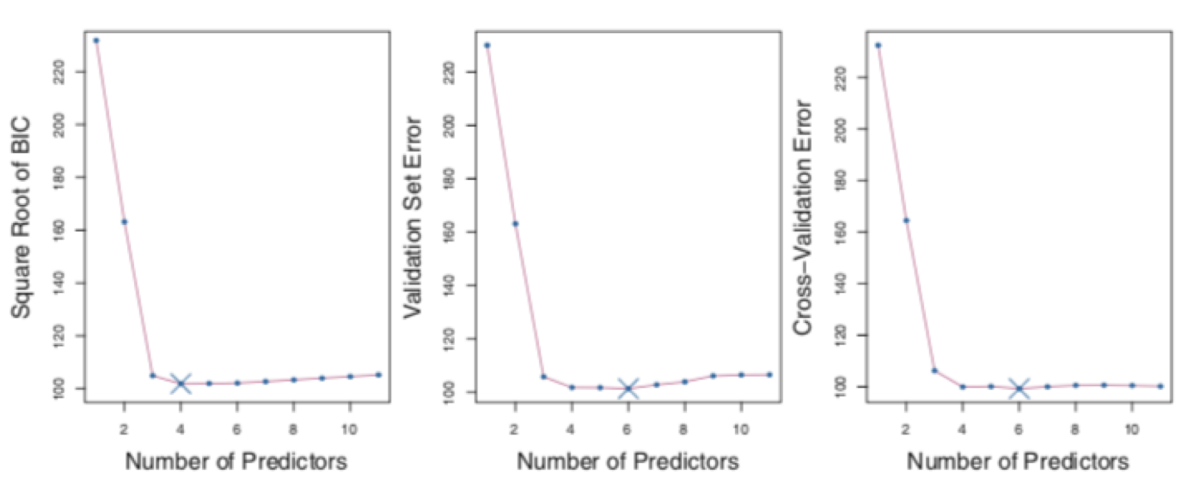


Linear Model Selection And Regularization

Machine Learning

Jay Urbain, PhD

Credits: G. James et al., An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics.

Standard Linear Model

- In the regression setting, the standard linear model is commonly used to describe the relationship between a response Y and a set of variables $X_1, X_2, ..., X_p$.

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

- Dataset with n examples, p variables, where $n >> p$.
- Consider approaches for extending the linear model framework:
  - Prediction accuracy when the number of samples is not much larger than the number of variables: $n \sim p$.
  - Prediction accuracy when $n < p$.
  - Model interpretability

Data has inherent variance that does not have predictive value.

$$
\begin{aligned}
E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\
&= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}} ,
\end{aligned}
$$

Necessitates the need for training, validation, and test sets.

- Training set – Learn model
- Validation set – tune model
- Test set – evaluate tuned model

Assessing the accuracy of regression model coefficients

Linear regression with residual term. Represents what we can't explain with our model.

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

RSS measures the amount of variability that is left unexplained after performing the regression

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

TSS (Total sum of squares) measures the total variance when measuring the response y.

$$\text{TSS} = \sum (y_i - \bar{y})^2$$

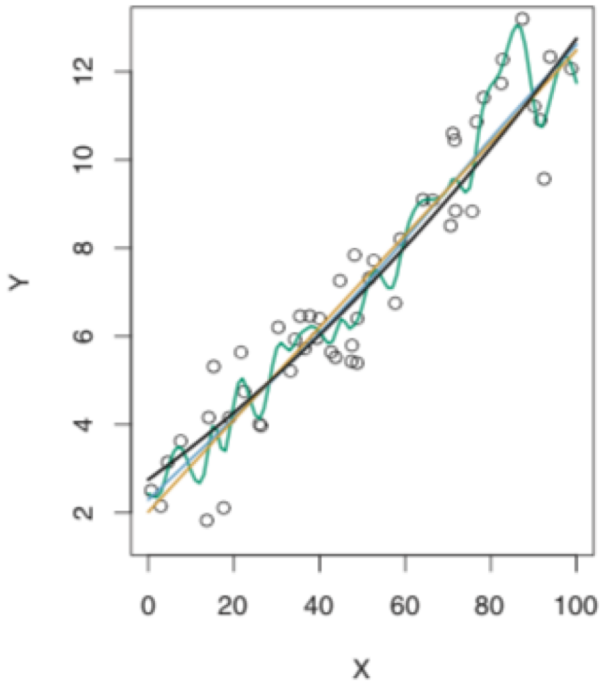
R^2 amount of variance explained by our model

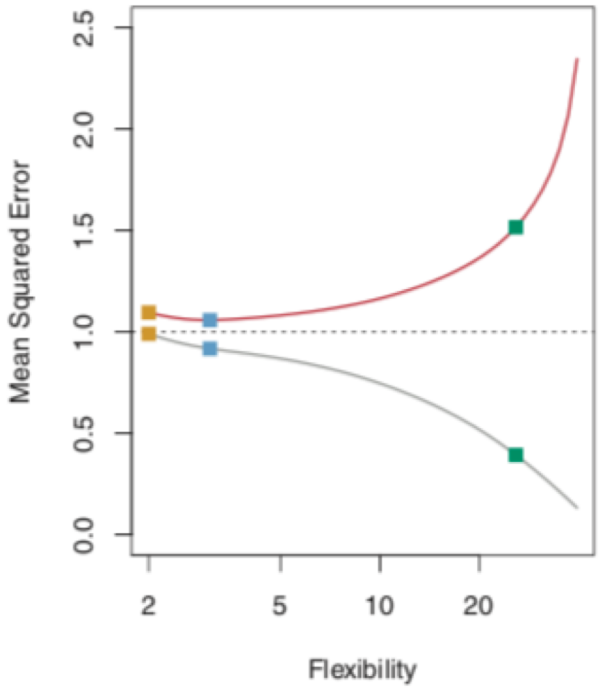
$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

The RSE is an estimate of the standard deviation of ε. It is basically the average amount that the response will deviate from the true regression line.

$$\text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}.$$

Bias-Variance Tradeoff





Subset Selection

- Fit a separate least squares regression for each possible combination of the predictors.
- Fit all *p* models that contain one predictor, all p models $\binom{p}{2} = p(p-1)/2$ contain two predictors, and so forth. Select best model of 2^p possibilities.
- 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 $k = 1, 2, \ldots p$:

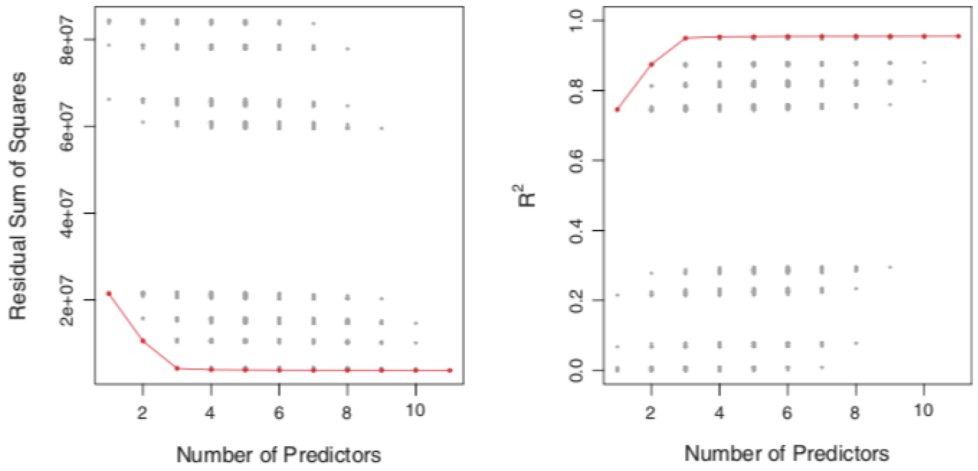
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (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.

Best Model Selection

- To select a single best model, we choose among these *p + 1* options.
- Must be careful, the *RSS* of these *p + 1* models decreases monotonically, and the R^2 increases monotonically, as the number of features included in the models increases.
- Use C_p, ***Adjusted R^2***, *BIC.*



Forward Stepwise Selection

- Best subset selection procedure considers all 2^p possible models containing subsets of the p predictors.
- Forward Stepwise Selection: Go from *2^p to (p +1) models*
  - Add predictors to the model, one-at-a-time, until all of the predictors are in the model.
  - At each step the variable that gives the greatest additional improvement to the fit is added to the model.

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.

Backward Stepwise Selection

- Begin with the full least squares model containing all *p* predictors, and then iteratively removes the least useful predictor, one-at-a-time.
- *1+p(p+1)/2* models

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.

Selecting Best Model

- MSE is generally an under-estimate of the *test* MSE (*MSE = RSS/n*).
- When we fit a model to the training data using least squares, we specifically estimate the regression coefficients such that the *training* RSS (but not the *test* RSS) is as small as possible.
- Therefore the training error will decrease as more variables are included in the model, but the test error may not.
- A number of techniques for adjusting the training error for the model size are available:
  - Cp, Akaike information criterion (AIC), Bayesian information criterion(BIC), and adjusted R^2.
  - Add penalty to training RSS to adjust for the fact that the training error tends to underestimate test error.

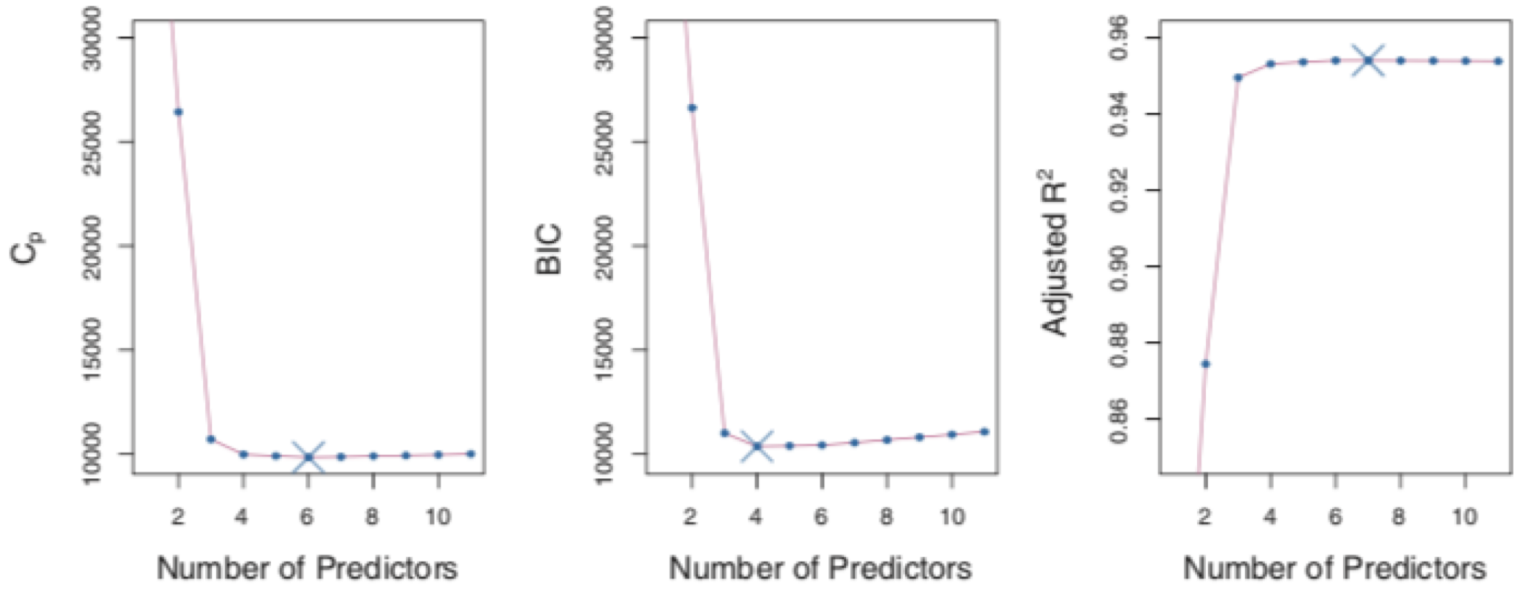
Adjusted R^2

- $R^2 = 1 - RSS/TSS$
- RSS always decreases as more variables are added to the model, therefore R^2 always increases as more variables are added.
- For a least squares model with *d* variables, the adjusted R^2 statistic is calculated as:

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}.$$

- Once all of the *correct* variables have been included in the model, adding additional noise variables will lead to only a very small decrease in RSS.
- Since adding noise variables leads to an increase in d, such variables will lead to an increase in RSS, and consequently a decrease in the adjusted R2.

Selecting the Best Model



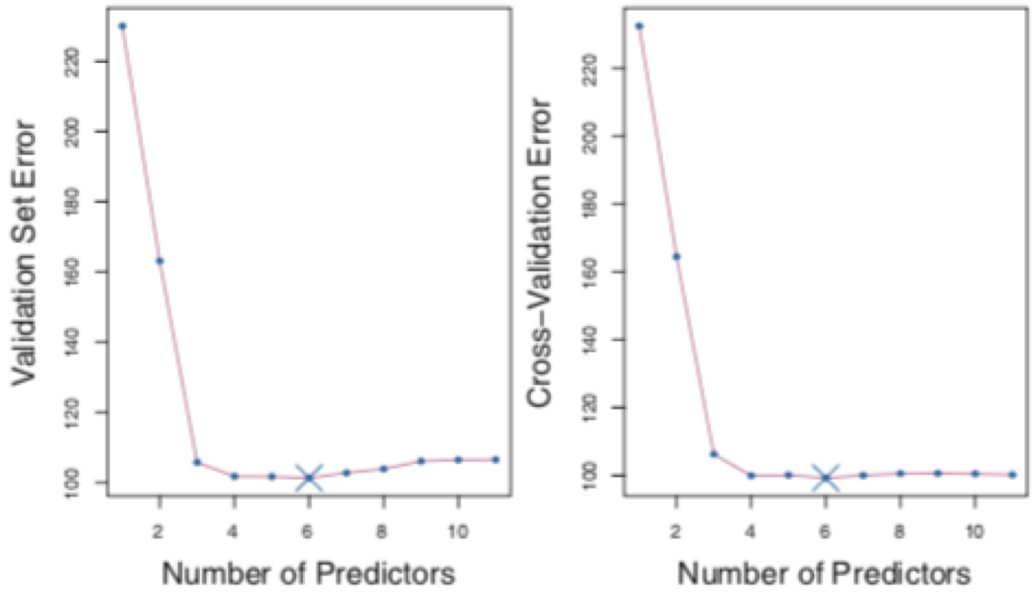
Feature Selection Implementations

sklearn.feature_selection https://scikit-learn.org/stable/modules/feature_selection.html

- **VarianceThreshold** – keep features with a minimal variance
- **SelectKBest** - removes all but the highest scoring features
- **SelectPercentile** - removes all but a user-specified highest scoring percentage of features
- **RFE** (Recursive feature elimination) - select features by recursively considering smaller and smaller sets of features.
- *Note: Use with cross-validation.*

Validation and Cross-Validation

- Alternative or complementary method: directly estimate the test error using validation set and cross-validation methods.



Review: Test error versus Training Error

Distinction between test error rate and training error rate:

- The *test* error is the average error that results from using a statistical learning method to predict the response on a *new observation*.
- So you should not tune your model on the test data.

Validation Set Approach

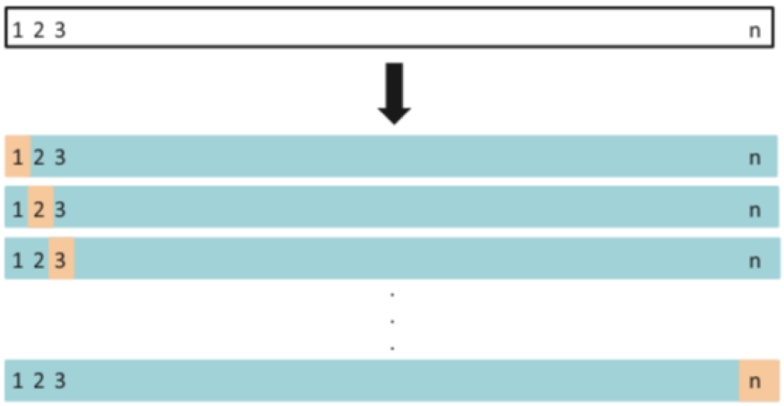
- We would like to estimate the test error associated with fitting a particular statistical learning method on a set of observations.
- Randomly divide the available set of *training observations* into two parts, a *training set* and a *validation set* or hold-out set.
- The model is fit on the *training set*, and the fitted model is used to predict the responses for the observations in the validation set as we tune the model.
- The resulting *validation set error rate*—typically assessed using MSE in the case of a quantitative response—provides an estimate of the test error rate.

Cross Validation Set Approach

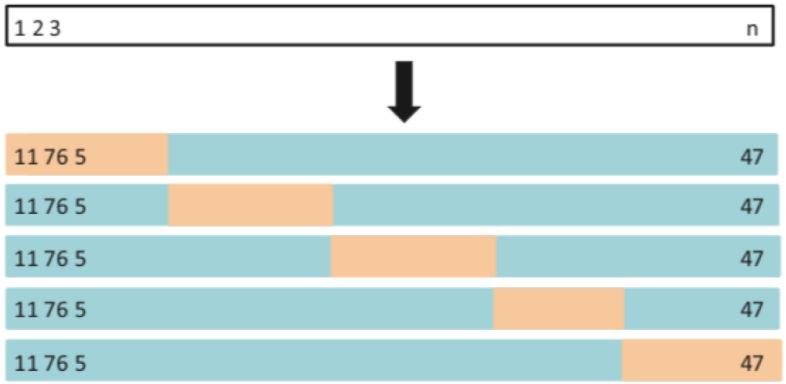
- Instead of splitting the set of observations into two parts, identify *n* parts.
- Evaluate MSE n times, once for each of n parts. Average the results.

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

LOOCV

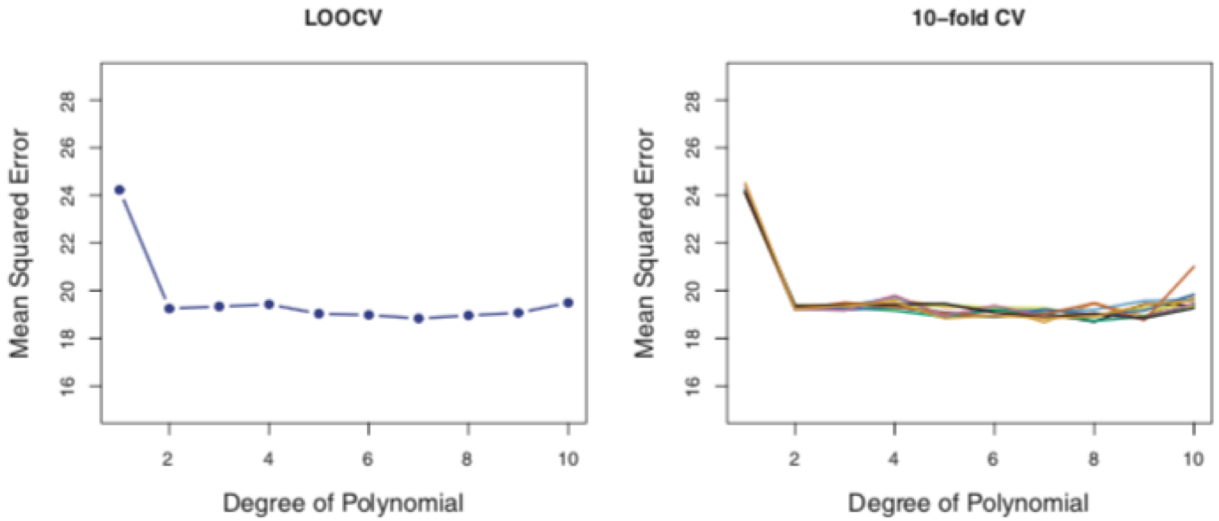


5-fold CV



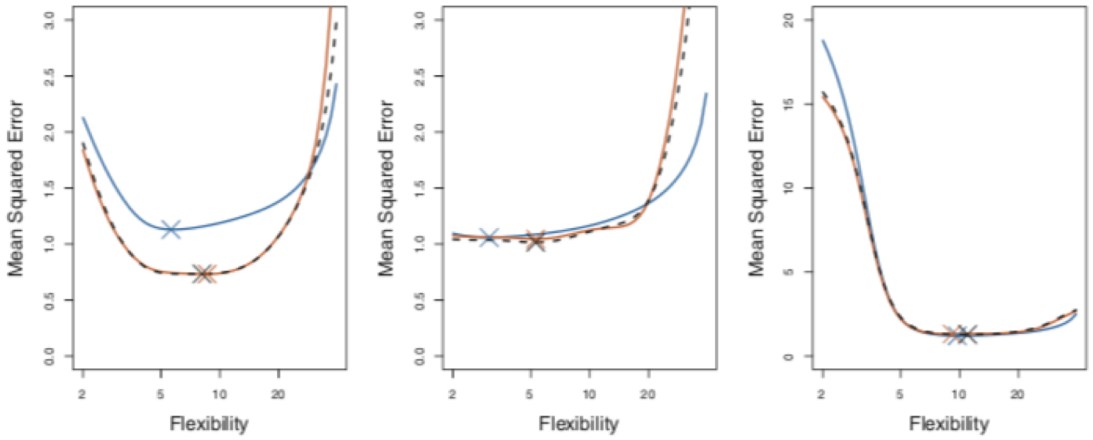
Cross-validation used on the Auto data set

- Left: The LOOCV (Leave One Out CV) error curve.
- Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.



Bias Variance Tradeoff

- Cross-validation estimates and true test error rates that result from applying smoothing splines
- The true test MSE is displayed in blue. The black dashed and orange solid lines respectively show the estimated LOOCV and 10-fold CV estimates.



CV in scikit-learn

- https://scikit-learn.org/stable/modules/cross_validation.html

```
from sklearn.model_selection import cross_val_score
clf = svm.SVC(kernel='linear', C=1)
scores = cross_val_score(clf, iris.data, iris.target, cv=5)
scores
```

Shrinkage Methods

- Alternative or complement to subset feature selection.
- Fit a model containing all *p* predictors using a technique that *constrains* or *regularizes* the coefficient estimates.
- Or, shrinks the coefficient estimates towards zero.
- Shrinking the coefficient estimates can significantly reduce their variance.

L1 and L2 Regularization (Shrinkage) Methods

- A regression model that uses **L1** regularization technique is called ***Lasso Regression*** and model which uses **L2** is called ***Ridge Regression***.
- Lasso and Ridge regression will produce a different set of coefficient estimates for each value of a tuning parameter λ.

Ridge Regression

- Least squares fitting procedure:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- Ridge regression: Add shrinkage penalty to penalize model complexity.
- Tuning parameters lambda controls tradeoff.
- Not applied to intercept.

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

Ridge regression

- **Ridge regression** adds "*squared magnitude*" of coefficient as penalty term to the loss function. Here the *highlighted* part represents **L2** regularization element.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Cost function

- if *lambda* is zero you are back to OLS.
- If *lambda* is very large then it will add too much weight and it will lead to under-fitting. So it's important how *lambda* is chosen.
- This technique works very well to avoid over-fitting issue.

Lasso Regression

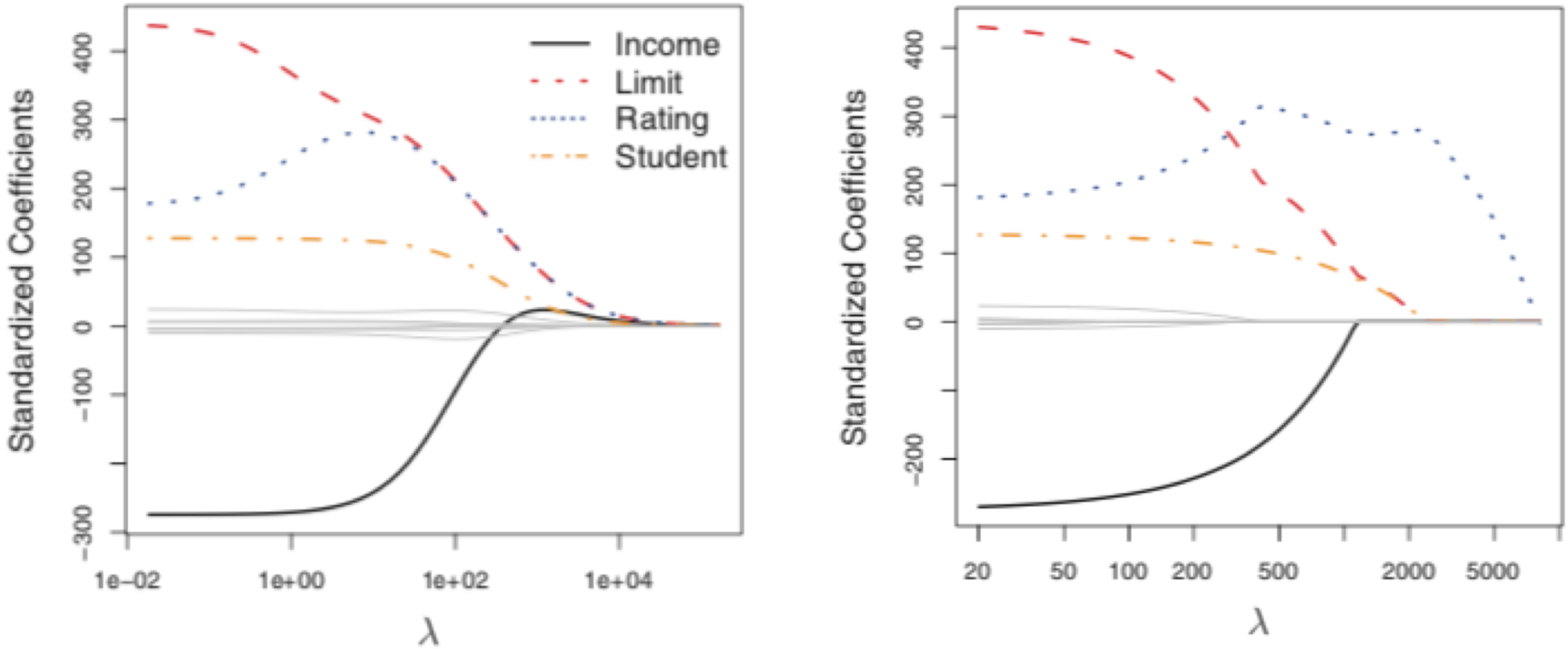
- Ridge regression will include all *p* predictors in the final model.
- **Lasso Regression** (Least Absolute Shrinkage and Selection Operator) adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Cost function

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- However, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when λ is sufficiently large.
- If *lambda* is zero then we will get back OLS whereas very large value will make coefficients zero hence it will under-fit.

Standardized ridge regression (left), lasso regression (right) coefficients as a function of λ. ISLR Credit card dataset.



Ridge vs. Lasso Regression

- **Key difference:** Lasso shrinks the less important feature's coefficient to zero thus, removing some features altogether. So, this works well for **feature selection** in case we have a huge number of features.

L2 loss function	L1 loss function
Not very robust	Robust
Stable solution	Unstable solution
Always one solution	Possibly multiple solutions

$$P = \alpha \Sigma_{n=1}^{N} \theta_n^2 \qquad P = \alpha \Sigma_{n=1}^{N} |\theta_n|$$