The Trade-Off Between Prediction Accuracy and Model Interpretability

- linear regression: fairly inflexible
- splines: considerably more flexible (can fit a much wider range of possible shapes to estimate $f$)

Inference:
- linear model: easy to understand the relationship between $Y$ and $X_1, X_2, \ldots, X_p$

Very flexible approaches (splines, SVM, etc)
- can lead to such complicated estimates of $f$
- hard to understand how any individual predictor is associated with the response (less interpretable)

LASSO:
- less flexible
- linear model + sparsity of $[\beta_0, \beta_1, \ldots, \beta_p]$
- more interpretable; only a small subset of predictors matter
Flexibility vs. Interpretability

Figure: Pretty clear pattern...
\[ R^2 = \frac{TSS - RSS}{TSS} \]

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

\[ TSS = \sum_{i} (y_i - \bar{y})^2 \]
Variable selection

Which predictors are associated with the response? (in order to fit a single model involving only those $d$ predictors)

- **Note:** $R^2$ always increase as you add more variables to the model.
- **Mallow’s:** $C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$
- **adjusted $R^2$:** $1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$
- **Akaike Information criterion AIC:** $\frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2)$

Cannot consider all $2^p$ models...

- **Best Subset Selection:** fit a separate least squares regression for each possible $k$-combination of the $p$ predictors, and select the best one
- **Forward selection:** start with the null model and keep adding predictors one by one
- **Backward selection:** start with all variables in the model, and remove the variable with the largest $p$-value
- **Mixed selection**
Prediction Accuracy

$$\text{MSE} = \mathbb{E}[(h(x^*) - \bar{h}(x^*))^2] + [f(x^*) - \bar{h}(x^*)]^2 + \text{Var}[\epsilon],$$

$x^*$: new data point, $f$: ground truth, $h$: our estimator

$$\text{MSE} = \text{Var}[h(x^*)] + \text{Bias}(h(x^*))^2 + \text{Var}[\epsilon]$$

- if true relationship is $\approx$ linear, the OLS will have low bias
- if $n \gg p$: OLS also has low variance, and will perform well on $X_{test}$
- if $n \sim p$: OLS has high variability, leads to overfitting/poor predictions on $X_{test}$
- if $n < p$: OLS estimate is no longer unique!

Today:

- by shrinking the estimated coefficients, we can often substantially reduce the variance at the cost of a negligible increase in bias
- can lead to substantial improvements in the accuracy with which we can predict the response for $X_{test}$
Model Interpretability

- some or most of the variables used in a multiple linear regression may not be associated with the response
- excluding them from the fit leads to a model that is more easily interpreted

Shrinkage/Regularization:
- by setting the corresponding coefficient estimates to zero — we can obtain a model that is more easily interpreted
- approach for automatically performing feature/variable selection and thus excluding irrelevant variables from a multiple regression model
Variable selection

- **Subset Selection**: identify a subset of $p$ predictors that best relate to the response, and perform OLS on them.
- **Shrinkage/Regularization**: fit a model involving all $p$ predictors, but the estimated coefficients are shrunken towards zero, or end up even equal to zero.
- **Dimension Reduction**: first project the $p$ predictors into a $d$-dimensional subspace, with $d < p$. The $d$ linear combinations, or projections are subsequently used as predictors in OLS.
Shrinkage Methods

- fit a model containing all $p$ predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero
- shrinking the coefficient estimates can significantly reduce their variance
- the two best-known techniques for shrinking the regression coefficients towards zero are
  - ridge regression
  - lasso regression

See Section 6.2 in the textbook.
Ridge Regression

Recall OLS estimates $\beta_0, \beta_1, \ldots, \beta_p$ such that it minimizes

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

Ridge regression shrinks $(\beta_0), \beta_1, \ldots, \beta_p$ towards zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$

\[
\hat{\beta}^{(\text{ridge})} = \arg \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2
\]

\[
= \arg \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left( y_i - x_i^T \beta \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2
\]

\[
= \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\| y - X\beta \|_2^2}_{\text{Loss}} + \underbrace{\lambda \| \beta \|_2^2}_{\text{Penalty}}
\]
\[ \hat{\beta}^{\text{ridge}} = \arg \min_{\beta \in \mathbb{R}^p} \left\| y - X\beta \right\|_2^2 + \lambda \left\| \beta \right\|_2^2 \]

Loss \hspace{1cm} \text{Penalty}

\[ \hat{\beta}^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y \]

Here \( \lambda \geq 0 \) is a tuning parameter

- controls the strength of the penalty term
- \( \lambda = 0 \) recovers the linear regression estimate
- \( \lambda = \infty \) leads to \( \hat{\beta}^{\text{ridge}} = 0 \)
- \( \lambda \in (0, \infty) \) trades-off two ideas: fitting a linear model of \( y \) on \( X \) versus shrinking the coefficients
Experimental setup

Given fixed covariates $x_i \in \mathbb{R}^p, i = 1, \ldots, n$
We observe:

- $y_i = f(x_i) + \epsilon_i, i = 1, \ldots, n$,
- for a linear model $f(x_i) = x_i^T \beta$
- $\epsilon_i \in \mathbb{R}$
- $\mathbb{E}[\epsilon_i] = 0$
- $\text{Var}[\epsilon_i] = \sigma^2$
- $\text{Cov}(\epsilon_i, \epsilon_j) = 0$
Experimental setup

- $n = 50$, $p = 30$, and $\sigma^2 = 1$
- The true model is linear with
  - 10 large coefficients (between 0.5 and 1) and
  - 20 small ones (between 0 and 0.3)
- Histogram of true coefficients

Source: R. Tibshirani
Experimental setup

- $n = 50$, $p = 30$, and $\sigma^2 = 1$
- The true model is linear with
  - 10 large coefficients (between 0.5 and 1) and
  - 20 small ones (between 0 and 0.3)
- Histogram

- the linear regression fit yields:
  - Squared bias $\approx 0.006$
  - Variance $\approx 0.627$
  - Pred. error $\approx 1 + 0.006 + 0.627 \approx 1.633$
**Improved prediction via shrinking**

![Graph showing prediction error vs. amount of shrinkage](Image)

<table>
<thead>
<tr>
<th></th>
<th>Linear Regression</th>
<th>Ridge Reg. (at its best)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squared bias</td>
<td>≈ 0.006</td>
<td>≈ 0.077</td>
</tr>
<tr>
<td>Variance</td>
<td>≈ 0.627</td>
<td>≈ 0.403</td>
</tr>
<tr>
<td>Pred. error</td>
<td>1 + 0.006 + 0.627</td>
<td>1 + 0.077 + 0.403</td>
</tr>
<tr>
<td></td>
<td>≈ 1.633</td>
<td>≈ 1.48</td>
</tr>
</tbody>
</table>
Ridge regression in R

- the function `lm.ridge` in the package MASS:
  - `lambdas = seq(0, 25, length = 100)`
  - `aa = lm.ridge(y ~ x + 0, lambda = lambdas)`
  - `b.ridge = coef(aa)`
  - `fit.ridge = b.ridge %*% t(x)`

- the `glmnet` function and package
Bias and variance of ridge regression

\[
\hat{\beta}^{\text{ridge}} = \arg \min_{\beta \in \mathbb{R}^p} \left( \| y - X\beta \|^2 + \lambda \| \beta \|^2 \right)
\]

Bias and variance:
- not as simple to derive for ridge regression as they are for linear regression
- but closed-form expressions are still possible

The general trend is:
- The bias increases as \( \lambda \) increases
- The variance decreases as \( \lambda \) increases

Questions:
- What is the bias at \( \lambda = 0 \)?
- What is the variance at \( \lambda = \infty \)?
Bias and variance of ridge regression
Mean squared error (MSE), bias and variance
Recap: ridge regression

- minimizes the usual regression criterion plus a penalty term on the squared $l_2$ norm of the coefficient vector
- shrinks the coefficients towards zero
- introduces some bias
- but can greatly reduce the variance
- overall, it results in a better mean-squared error
- the amount of shrinkage is controlled by $\lambda$
- performs particularly well when there is a subset of true coefficients that are small or even zero
- not as great when all of the true coefficients are moderately large (can still outperform OLS over a pretty narrow range of (small) $\lambda$ values)
- does NOT set coefficients to zero exactly, and therefore cannot perform variable selection in the linear model
LASSO

Recall OLS estimates $\beta_0, \beta_1, \ldots, \beta_p$ such that it minimizes

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

LASSO sets some of the coefficients $\beta_1, \ldots, \beta_p$ to zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$

$$\hat{\beta}^{(\text{lasso})} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

$$= \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} \left( y_i - x_i^T \beta \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

$$= \arg\min_{\beta \in \mathbb{R}^p} \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$
arg \min_{\beta \in \mathbb{R}^p} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda \underbrace{||\beta||_1}_{\text{Penalty}}

The tuning parameter $\lambda$ controls the strength of the penalty, and (like ridge regression), we get

- $\hat{\beta}^{(\text{lasso})}$ = the usual OLS estimator, whenever $\lambda = 0$
- $\hat{\beta}^{(\text{lasso})} = 0$, whenever $\lambda = \infty$

For $\lambda \in (0, \infty)$, we are balancing the trade-offs:

- fitting a linear model of $y$ on $X$
- shrinking the coefficients; but the nature of the $l_1$ penalty causes some coefficients to be shrunken to zero exactly

LASSO (vs. RIDGE):

- LASSO performs variable selection in the linear model
- has no closed form solution (quadratic programming from convex optimization)
- as $\lambda$ increases, more coefficients are set to zero (less variables are selected), and among the nonzero coefficients, more shrinkage is employed
Ridge: coefficient paths
LASSO: coefficient paths
Fitting LASSO models in R with the glmnet package

- Lasso and Elastic-Net Regularized Generalized Linear Models
- Fits a wide variety of models (linear models, generalized linear models, multinomial models) with LASSO penalties
- The syntax is fairly straightforward, though it differs from `lm` in that it requires you to form your own design matrix:
  ```
  fit = glmnet(X, y)
  ```
- The package also allows you to conveniently carry out cross-validation:
  ```
  cvfit = cv.glmnet(X, y); plot(cvfit);
  ```
- Prediction with cross validation. Example:
  ```
  X = matrix(rnorm(100*20), 100, 20)
  y = rnorm(100)
  cv.fit = cv.glmnet(X, y)
  yhat = predict(cv.fit, newx=X[1:5,])
  coef(cv.fit)
  coef(cv.fit, s = "lambda.min")
  ```