# ECON3389 Machine Learning in Economics

## Module 4
## Feature Selection in Linear Models

Alberto Cappello

Department of Economics, Boston College

Fall 2024

# Overview

Agenda:

- Ridge regression.
- Lasso regression.

Readings:

- ISLR Chapter 6, sections 6.1 and 6.2

# Shrinkage Methods

- The subset selection methods fit a linear model that contains only a subset of the predictors. This is equivalent to setting the coefficients on excluded predictors to zero prior to running the estimation algorithm.

# Shrinkage Methods

- The subset selection methods fit a linear model that contains only a subset of the predictors. This is equivalent to setting the coefficients on excluded predictors to zero prior to running the estimation algorithm.
- As an alternative, one can fit a model containing all $p$ predictors using a technique that *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero as part of the estimation algorithm.
- It may not be immediately obvious why such a constraint should improve the fit or if the algorithm will work in the first place, but it turns out that shrinking the coefficient estimates can significantly reduce their variance at a cost of a minor increase in bias.

# Shrinkage Methods

- The subset selection methods fit a linear model that contains only a subset of the predictors. This is equivalent to setting the coefficients on excluded predictors to zero prior to running the estimation algorithm.
- As an alternative, one can fit a model containing all $p$ predictors using a technique that *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero as part of the estimation algorithm.
- It may not be immediately obvious why such a constraint should improve the fit or if the algorithm will work in the first place, but it turns out that shrinking the coefficient estimates can significantly reduce their variance at a cost of a minor increase in bias.
- Two most common shrinkage/regularization methods are *ridge regression* and *lasso regression*.

# Ridge Regression

- Standard least squares regression fits the model by picking values of $\beta_0, \beta_1, \ldots, \beta_p$ that minimize

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

# Ridge Regression

- Standard least squares regression fits the model by picking values of $\beta_0, \beta_1, \ldots, \beta_p$ that minimize

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

- Ridge regression instead picks coefficient values $\widehat{\beta}^R$ that minimize

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

- Parameter $\lambda$ is the *tuning parameter*, to be determined separately.

# Ridge Regression

- The nature of ridge regression is similar to that of OLS: seek coefficient values that make the model fit the data well (by making RSS small).
- However, now we can no longer set values of coefficients to just significantly decrease RSS. This is because the second term $\lambda \sum_{j=1}^{p} \beta_j^2$, called a *shrinkage penalty*, will increase our loss function if values of $\beta_0, \beta_1, \ldots, \beta_p$ are far away from zero.

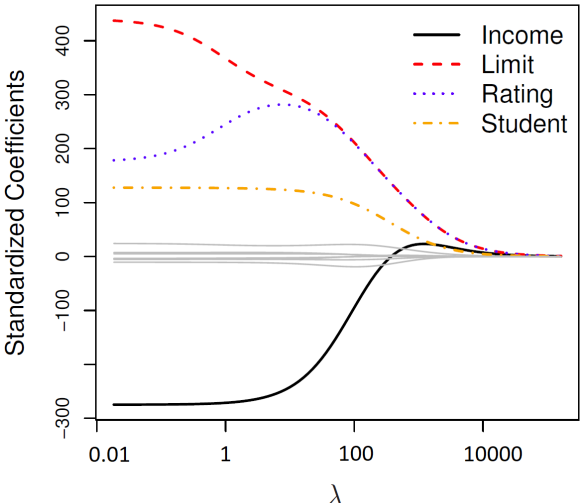
# Ridge Regression

- The nature of ridge regression is similar to that of OLS: seek coefficient values that make the model fit the data well (by making RSS small).
- However, now we can no longer set values of coefficients to just significantly decrease RSS. This is because the second term $\lambda \sum_{j=1}^{p} \beta_j^2$, called a *shrinkage penalty*, will increase our loss function if values of $\beta_0, \beta_1, \ldots, \beta_p$ are far away from zero.
- Because loss function now has two terms to balance out, the extra second term has the effect of *shrinking* the estimates of $\beta_j$ towards zero.

# Ridge Regression

- The nature of ridge regression is similar to that of OLS: seek coefficient values that make the model fit the data well (by making RSS small).
- However, now we can no longer set values of coefficients to just significantly decrease RSS. This is because the second term $\lambda \sum_{j=1}^{p} \beta_j^2$, called a *shrinkage penalty*, will increase our loss function if values of $\beta_0, \beta_1, \ldots, \beta_p$ are far away from zero.
- Because loss function now has two terms to balance out, the extra second term has the effect of *shrinking* the estimates of $\beta_j$ towards zero.
- The tuning parameter $\lambda$ serves to control the relative impact of these two terms on the regression coefficient estimates. Selecting a good value for $\lambda$ is critical, and is done via cross-validation.

# Credit Card Data Example




- As can be seen from the picture, it is always possible to set $\lambda$ to a value that will shrink all coefficients arbitrary close to zero.
- As such, we need to perform cross-validation testing to see which value of $\lambda$ achieves minimal total value of ridge loss function.
- The process is usually done via a grid search algorithm (more on that later)

# Feature Scaling and Standardization

- Standard least squares coefficient estimates are *scale equivariant*: multiplying $X_j$ by a constant $c$ simply leads to a scaling of the least squares coefficient estimate $\widehat{\beta}_j$ by a factor of $1/c$. In other words, regardless of how the $j$-th predictor is scaled, $\widehat{\beta}_j X_j$ will always remain the same.

# Feature Scaling and Standardization

- Standard least squares coefficient estimates are *scale equivariant*: multiplying $X_j$ by a constant $c$ simply leads to a scaling of the least squares coefficient estimate $\widehat{\beta}_j$ by a factor of $1/c$. In other words, regardless of how the $j$-th predictor is scaled, $\widehat{\beta}_j X_j$ will always remain the same.
- In contrast, the ridge regression coefficient estimates can change substantially when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression loss function.
- Unlike the first term, which contains $\widehat{\beta}_j X_j$ parts, the shrinkage penalty contains values of only $\widehat{\beta}_j^2$, thus making is scale-dependent.

# Feature Scaling and Standardization

- Standard least squares coefficient estimates are *scale equivariant*: multiplying $X_j$ by a constant $c$ simply leads to a scaling of the least squares coefficient estimate $\widehat{\beta}_j$ by a factor of $1/c$. In other words, regardless of how the $j$-th predictor is scaled, $\widehat{\beta}_j X_j$ will always remain the same.
- In contrast, the ridge regression coefficient estimates can change substantially when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression loss function.
- Unlike the first term, which contains $\widehat{\beta}_j X_j$ parts, the shrinkage penalty contains values of only $\widehat{\beta}_j^2$, thus making is scale-dependent.
- Therefore, it is best to apply ridge regression after *standardizing* the predictors:

$$\widetilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \overline{x}_j\right)^2}}$$

# Why Does Ridge Regression Improve Over LS: Bias-variance trade-off

- Suppose our test data `Te` consists of a single data point $(x_0, y_0)$. Then

$$MSE = \mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \mathbb{E}\left[\left(f(x_0) - \hat{f}(x_0)\right)^2\right] + \text{Var}(\epsilon_0)$$

$$= \underbrace{\mathbb{E}\left[\left(\hat{f}(x_0) - \mathbb{E}\left[\hat{f}(x_0)\right]\right)^2\right]}_{\text{Var}\left(\hat{f}(x_0)\right)} + \underbrace{\mathbb{E}\left[\left(f(x_0) - \mathbb{E}\left[\hat{f}(x_0)\right]\right)^2\right]}_{\mathbb{E}\left[\text{Bias}^2\left(\hat{f}(x_0)\right)\right]} + \text{Var}(\epsilon_0)$$

- Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set
- Bias refers to the error that is introduced by approximating a real-life problem by a much simpler model

# Why Does Ridge Regression Improve Over LS: Bias-variance trade-off

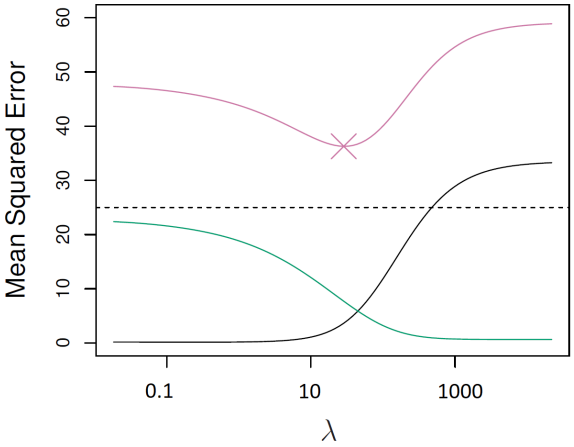
- Suppose our test data `Te` consists of a single data point $(x_0, y_0)$. Then

$$MSE = \mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \mathbb{E}\left[\left(f(x_0) - \hat{f}(x_0)\right)^2\right] + \text{Var}(\epsilon_0)$$

$$= \underbrace{\mathbb{E}\left[\left(\hat{f}(x_0) - \mathbb{E}\left[\hat{f}(x_0)\right]\right)^2\right]}_{\text{Var}\left(\hat{f}(x_0)\right)} + \underbrace{\mathbb{E}\left[\left(f(x_0) - \mathbb{E}\left[\hat{f}(x_0)\right]\right)^2\right]}_{\mathbb{E}\left[\text{Bias}^2\left(\hat{f}(x_0)\right)\right]} + \text{Var}(\epsilon_0)$$

- Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set
- Bias refers to the error that is introduced by approximating a real-life problem by a much simpler model
- Typically as the flexibility of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on MSE amounts to a bias-variance trade-off.

# Why Does Ridge Regression Improve Over LS?




- **Squared bias**, **variance** and **MSE**.
- 45 predictors (p), 50 observations (n)
- Because OLS is free to choose any coefficient values, it tends to pick the ones that provide best fit, meaning less bias and more variance.
- Ridge regression, on the other hand, is penalized for choosing coefficients with high second moments, thus leading to less variance, slightly more bias, but lower MSE overall.

# Lasso regression

- Unlike subset selection, which generally selects models that involve just a subset of all variables, ridge regression will include all $p$ predictors in the final model. This makes ridge regression completely infeasible when $p > n$, as if often the case, for example, with Internet-related data.

# Lasso regression

- Unlike subset selection, which generally selects models that involve just a subset of all variables, ridge regression will include all $p$ predictors in the final model. This makes ridge regression completely infeasible when $p > n$, as if often the case, for example, with Internet-related data.
- The LASSO (Least Absolute Shrinkage and Selection Operator) is an alternative that overcomes this disadvantage. It achieves that by using a different type of shrinkage penalty:

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

- In statistical lingo, this type of penalty is known as $\ell_1$-penalty, because it uses $\ell_1$ norm of coefficient vector $\beta$ given by $||\beta||_1 = \sum|\beta_j|$. Ridge regression, on the other hand, uses $\ell_2$ norm as a penalty, given by $||\beta||_2 = \sum\beta_j^2$

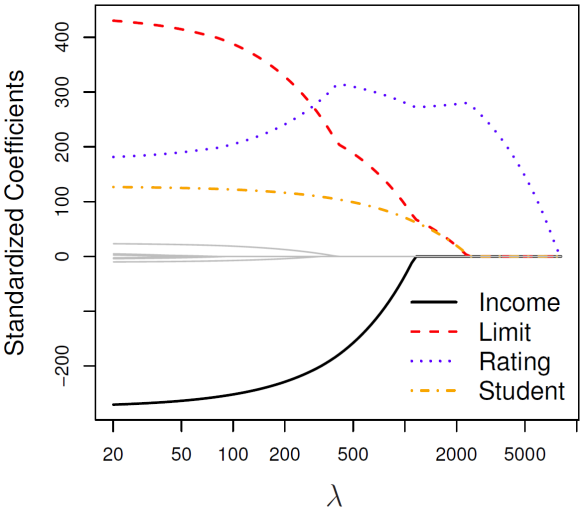
# Lasso Variable Selection

- As with ridge regression, lasso shrinks all coefficient estimates towards zero.
- However, unlike $\ell_2$ penalty in ridge regression, lasso penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter $\lambda$ is sufficiently large.

# Lasso Variable Selection

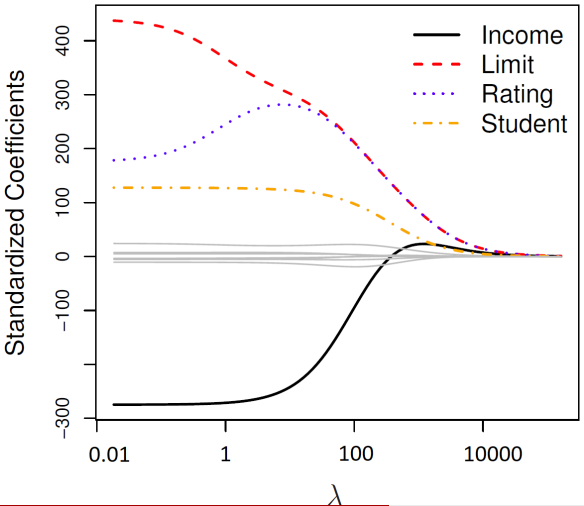
- As with ridge regression, lasso shrinks all coefficient estimates towards zero.
- However, unlike $\ell_2$ penalty in ridge regression, lasso penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter $\lambda$ is sufficiently large.
- Hence, much like best subset selection, lasso performs variable selection, starting with the full set of $p$ variables. We say that lasso yields *sparse* models — that is, models that involve only a subset of the variables.
- As in ridge regression, selecting a good value of $\lambda$ for lasso is critical; cross-validation is again the method of choice.

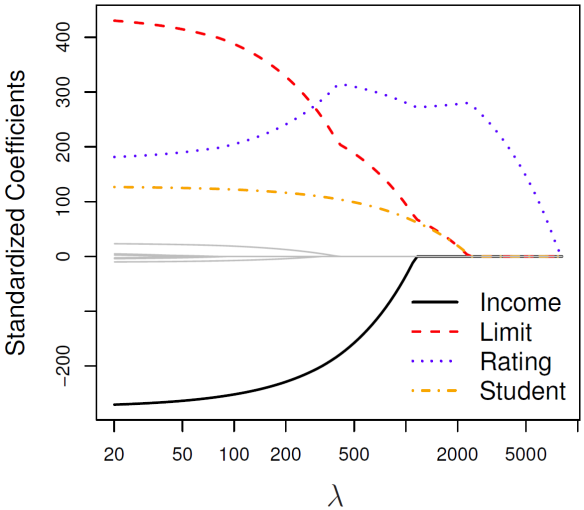
# Credit Card Data Example




- Similar to ridge regression, setting $\lambda$ to sufficiently high value will shrink all coefficients to zero.
- Unlike ridge regression, lasso coefficients will get shrunk exactly to zero in a single jump, without smooth continuous decline.
- Additionally, while ridge regression shrinks all coefficients close to zero around the same values of $\lambda$, lasso sets some coefficients to zero much earlier than others.

# RIDGE vs LASSO



# RIDGE vs LASSO

- Why is it that in lasso regression we get some of the coefficients shrunk exactly to zero, but not in ridge regression?

# RIDGE vs LASSO

- Why is it that in lasso regression we get some of the coefficients shrunk exactly to zero, but not in ridge regression?
- One can show that lasso and ridge regression coefficient estimates solve the following problems:

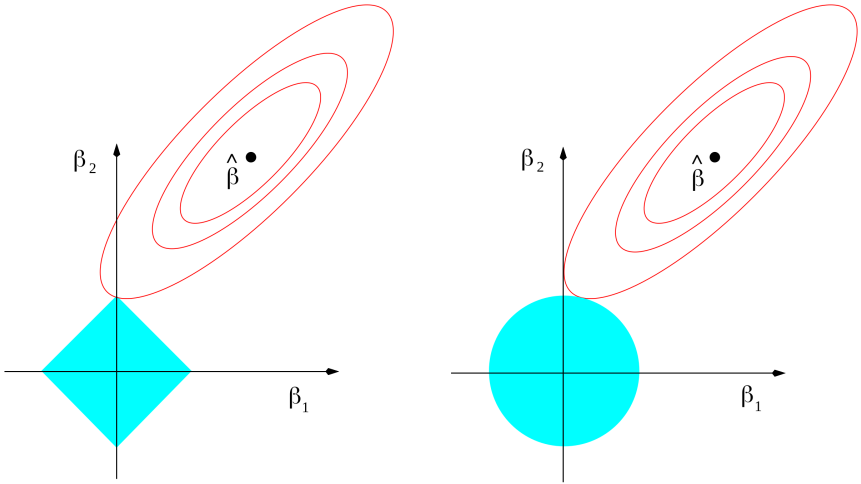
$$\min_{\beta} \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

and

$$\min_{\beta} \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

- These two problems have a useful geometric representation that shows exactly why lasso induces sparsity among coefficients.

# RIDGE vs LASSO




Left: lasso Right: Ridge

# Selecting the Tuning Parameter $\lambda$

- Both with ridge and with lasso we need to select the value for the tuning parameter $\lambda$ or equivalently, the value of the constraint $s$ in a way that will not lead to overfitting or other mistakes. Cross-validation provides a simple way to tackle this problem.

# Selecting the Tuning Parameter $\lambda$

- Both with ridge and with lasso we need to select the value for the tuning parameter $\lambda$ or equivalently, the value of the constraint $s$ in a way that will not lead to overfitting or other mistakes. Cross-validation provides a simple way to tackle this problem.
- We choose a grid of $\lambda$ values and fit a separate model for every value from that grid using K-fold cross-validation.
- We then compute the cross-validation error for each value of $\lambda$ and select the one for which that error is smallest.

# Selecting the Tuning Parameter $\lambda$

- Both with ridge and with lasso we need to select the value for the tuning parameter $\lambda$ or equivalently, the value of the constraint $s$ in a way that will not lead to overfitting or other mistakes. Cross-validation provides a simple way to tackle this problem.
- We choose a grid of $\lambda$ values and fit a separate model for every value from that grid using K-fold cross-validation.
- We then compute the cross-validation error for each value of $\lambda$ and select the one for which that error is smallest.
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

# Pros and Cons of Lasso

- In terms of overall fit neither ridge regression nor lasso will universally dominate the other.

# Pros and Cons of Lasso

- In terms of overall fit neither ridge regression nor lasso will universally dominate the other.
- In general, one might expect lasso to perform better when the response is a function of only a relatively small number of predictors. However, that is never known *a priori* with real-life data.

# Pros and Cons of Lasso

- In terms of overall fit neither ridge regression nor lasso will universally dominate the other.
- In general, one might expect lasso to perform better when the response is a function of only a relatively small number of predictors. However, that is never known *a priori* with real-life data.
- Ridge regression can perform better if $p < n$ and there is no a priori reason for some of the variables to not be included in the model.

# Pros and Cons of Lasso

- In terms of overall fit neither ridge regression nor lasso will universally dominate the other.
- In general, one might expect lasso to perform better when the response is a function of only a relatively small number of predictors. However, that is never known *a priori* with real-life data.
- Ridge regression can perform better if $p < n$ and there is no a priori reason for some of the variables to not be included in the model.
- Lasso can perform variable selection and model estimation with $p > n$, but has a known issue of ignoring groups of correlated variables (e.g. performance metrics of NBA players) and almost randomly selecting only one variable out of the group.

# Lasso and Economics

- Despite its know flaws, over the past decade lasso has become very popular with both academic researchers and applied economists.
- From the theoretical perspective, multiple extensions and variations of lasso has been suggested, and today advanced versions of it can deal both with correlated regressors (group lasso, elastic net) and biased estimates (adaptive lasso, post-lasso).
- The main driving force behind is the ability to tackle datasets that previously were completely unusable due to number of variables $p$ being close or even larger than sample size $n$.
- Additionally, lasso allows economists to utilize *sparse* structural models, e.g. consumer preferences across hundreds of product attributes with most of them having zero importance.