Lecture 18(b): LASSO and Ridge regression

Foundations of Data Science:

Algorithms and Mathematical Foundations

Mihai Cucuringu
mihai.cucuringu@stats.ox.ac.uk

CDT in Mathematics of Random System
University of Oxford

28 September, 2023

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)

Example: 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

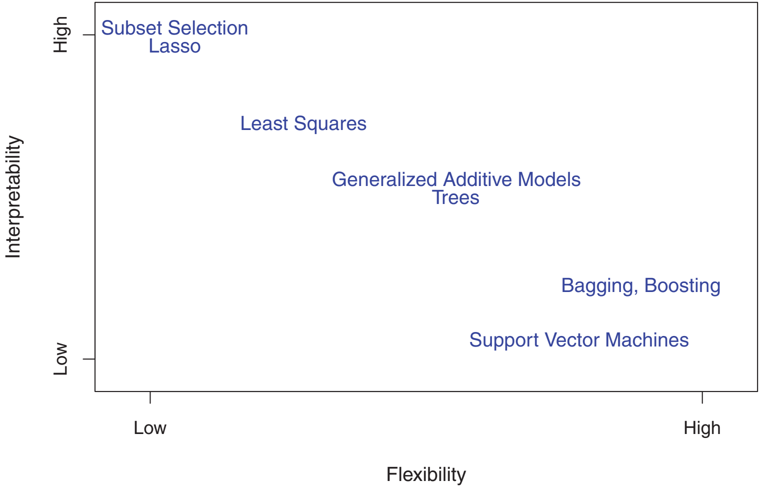


Figure: A representation of the trade-off between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.

R^2

- also called the *coefficient of determination*
- pronounced "R squared",
- gives the proportion of the variance in the dependent variable that is predictable from the independent variable/s

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

where

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

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

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
- adjusted R^2: $1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$
- Mallow's: $C_p = \frac{1}{n}(\text{RSS} + 2p\hat{\sigma}^2)$
- Akaike Information criterion AIC $= \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2p\hat{\sigma}^2)$

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

Prediction Accuracy

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

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

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

- if true relationship is ≈ linear, the OLS will have low bias
- if $n >> p$: OLS also has low variance, and performs 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
- Dimensionality 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 (principal component regression PCR)

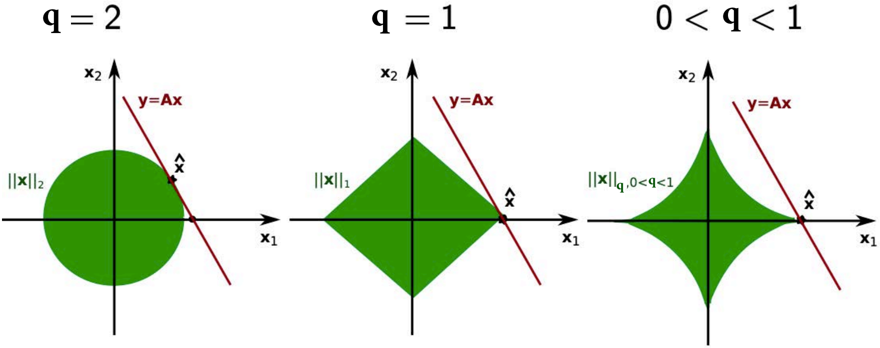
Shrinkage Methods

- 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 ISLR textbook.

Regularization penalty

Idea: impose an ℓ_q penalty on the vector of beta coefficients, to promote shrinking them towards zero



Credit: Peter Gerstoft

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

$$\begin{aligned}
\hat{\beta}^{(\text{ridge})} &= \arg\min_{\beta \in \mathbb{R}^p} \overbrace{\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2}^{\text{RSS}} + \lambda \sum_{j=1}^{p} \beta_j^2 \\
&= \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - x_i^T \beta \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\
&= \arg\min_{\beta \in \mathbb{R}^p} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda \underbrace{||\beta||_2^2}_{\text{Penalty}}
\end{aligned}$$

$$\hat{\beta}^{\text{(ridge)}} = \arg\min_{\beta \in \mathbb{R}^p} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda \underbrace{||\beta||_2^2}_{\text{Penalty}}$$

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

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

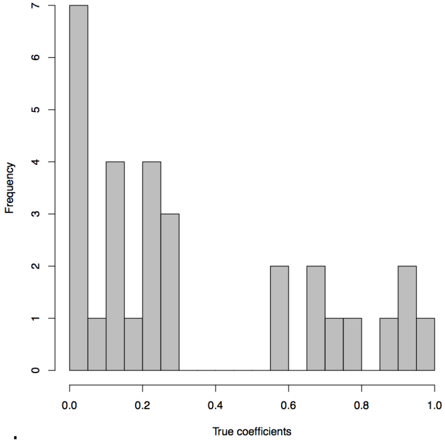
Experimental setup

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 $f(x_i) = x_i^T \beta$
- $\epsilon_i \in \mathbb{R}$
- $\mathbb{E}[\epsilon_i] = 0$
- $\mathrm{Var}[\epsilon_i] = \sigma^2$
- $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$

Experimental setup

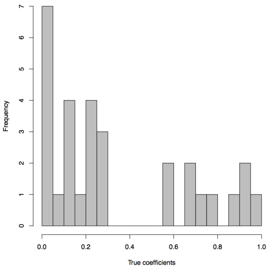
- $n = 50$, $p = 30$, and $\sigma^2 = 1$
- The true model is linear with
  - 10 large coefficients (between 0.5 and 1) and
  - 20 small ones (between 0 and 0.3)
- Histogram of true coefficients

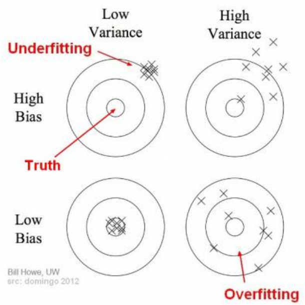


Source: R. Tibshirani

Experimental setup

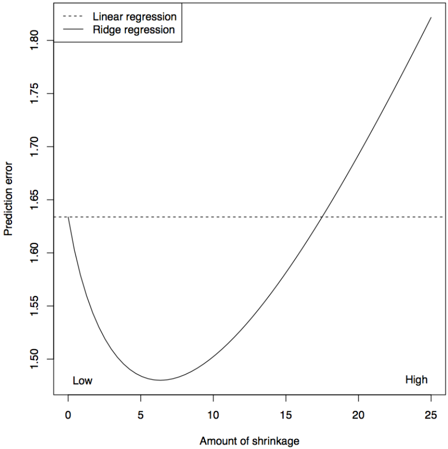
- $n = 50$, $p = 30$, and $\sigma^2 = 1$
- The true model is linear with
  - 10 large coefficients (between 0.5 and 1) and
  - 20 small ones (between 0 and 0.3)
- Histogram of true coefficients





- the linear regression fit yields:
  - Squared bias ≈ 0.006
  - Variance ≈ 0.627
  - Pred. error $\approx 1 + 0.006 + 0.627 \approx 1.633$

Improved prediction via shrinking



	Linear Regression	Ridge Reg. (at its best)
Squared bias	≈ 0.006	≈ 0.077
Variance	≈ 0.627	≈ 0.403
Pred. error	$\approx 1 + 0.006 + 0.627$	$\approx 1 + 0.077 + 0.403$
	≈ 1.633	≈ 1.48

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/package is also available in R.

Bias and variance of ridge regression

$$\hat{\beta}^{(\text{ridge})} = \arg\min_{\beta \in \mathbb{R}^p} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda \underbrace{||\beta||_2^2}_{\text{Penalty}}$$

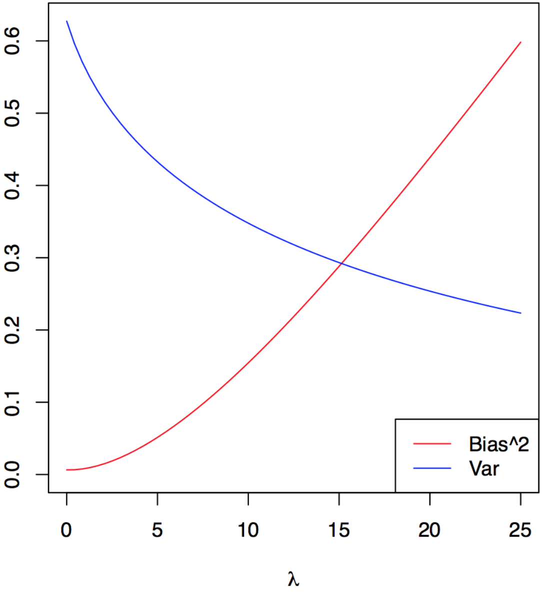
Bias and variance:

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

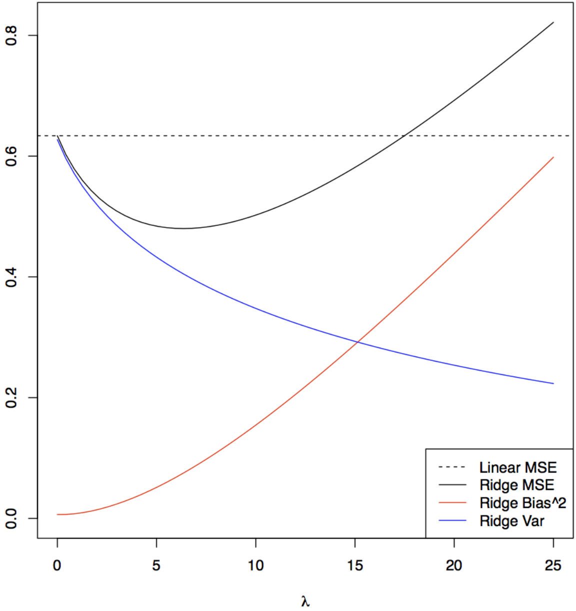
The general trend is:

- The bias increases as λ increases
- The variance decreases as λ increases

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 λ
- 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) λ 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 $X \in \mathbb{R}^{n \times p}$

$$\begin{aligned}
\hat{\beta}^{(\text{lasso})} &= \underset{\beta \in \mathbb{R}^p}{\arg\min} \overbrace{\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2}^{\text{RSS}} + \lambda \overbrace{\sum_{j=1}^{p} |\beta_j|}^{\text{Penalty}} \\
&= \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \left(y_i - x_i^T \beta \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\
&= \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda \underbrace{||\beta||_1}_{\text{Penalty}}
\end{aligned}$$

$$\underset{\beta \in \mathbb{R}^p}{\arg\min} \quad \underbrace{||y - X\beta||_2^2}_{\text{Loss}} \quad + \quad \lambda \underbrace{||\beta||_1}_{\text{Penalty}}$$

- The tuning parameter λ controls the strength of the penalty, and (like ridge regression), we get
  - $\hat{\beta}^{(lasso)}$ = the usual OLS estimator, whenever $\lambda = 0$
  - $\hat{\beta}^{(lasso)}$ = 0, whenever $\lambda = \infty$

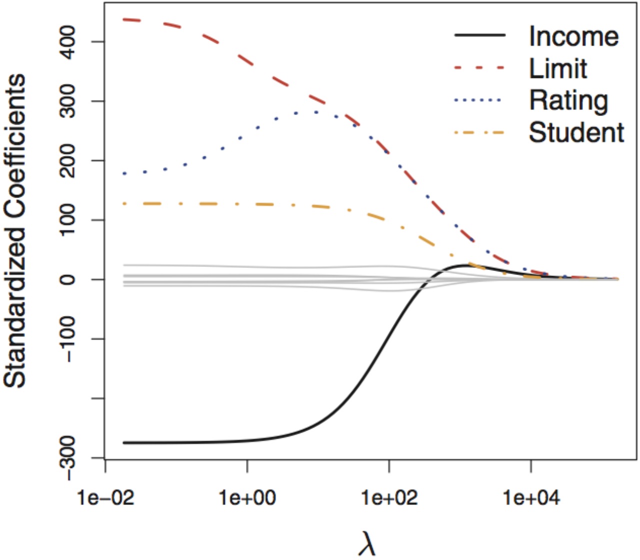
For $\lambda \in (0, \infty)$, we are balancing the trade-offs:

- 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**

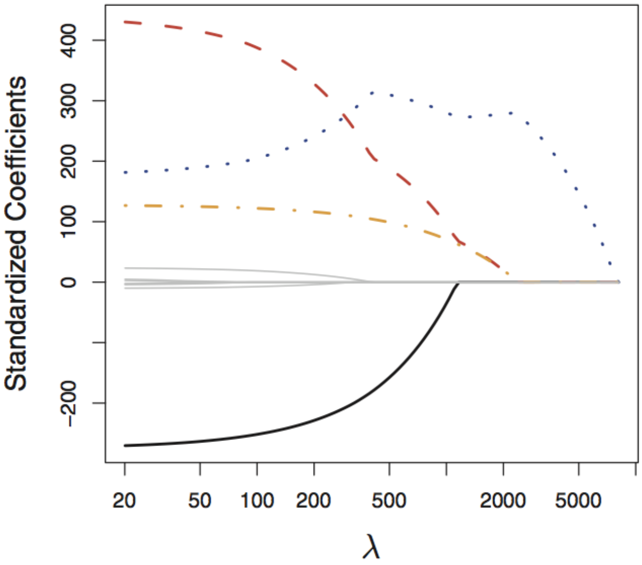
LASSO (vs. Ridge):

- LASSO performs variable selection in the linear model
- has no closed-form solution (various optimization techniques are employed)
- as λ 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")
  ```

Elastic net - the best of both worlds

Elastic Net combines the penalties of Ridge and LASSO.

$$\hat{\beta}^{(\text{elastic net})} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{||y - X\beta||_2^2}_{\text{Loss}} + \lambda_1 \underbrace{||\beta||_1}_{\text{Penalty}} + \lambda_2 \underbrace{||\beta||_2}_{\text{Penalty}}$$

Addresses several shortcomings of LASSO:

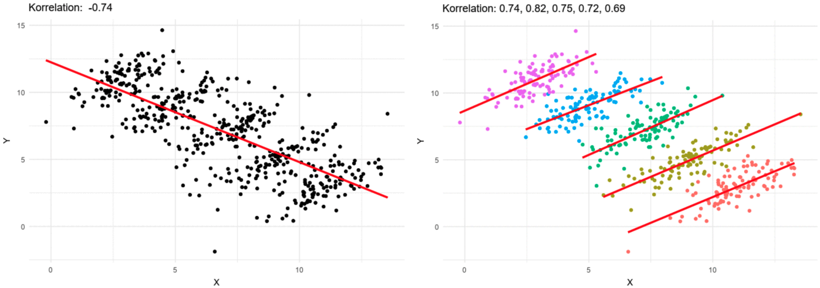
- for $n < p$ (more covariates/features than samples) LASSO can select only n covariates (even if more are truly associated with the response)
- it tends to select only one covariate from any set of highly correlated covariates
- for $n > p$, if the covariates are strongly correlated, Ridge tends to perform better

Elastic Net:

- highly correlated covariates will tend to have similar regression coefficients (desirable *grouping effect*)

Simpson's paradox - beware!

Phenomenon in statistics when certain trends that appear when a dataset is separated into groups are reversed when the data are aggregated.



- can be resolved when confounding variables and causal relations are appropriately addressed in the statistical modeling
- misleading results that the misuse of statistics can generate

Source: Wiki