STA 35C: Statistical Data Science III

Lecture 16: Linear Model Selection – Regularization Methods

Dogyoon Song

Spring 2025, UC Davis

Agenda

- **Last time:** Model selection via subset selection
  - Best subset selection: identify relevant predictors among many
  - Stepwise selection: A computationally more tractable alternative (greedy alg)
- **Today:** Regularization
  - Overview: what regularization is & why it can help
  - Two main examples in linear regression
    - Ridge regression
    - The lasso (least absolute shrinkage and selection operator)

Recap: Subset selection

Best subset selection:

- Exhaustively search all 2^p subsets
- Pick the best model for each size k, then choose among them (via R^2_{adj} or CV)
- Feasible only for smaller p (high computational cost)

Forward stepwise selection:

- Greedy approach: add one predictor at a time
- Complexity: $\mathcal{O}(p^2)$ instead of 2^p
- May miss the global optimum if local decisions are suboptimal

Backward stepwise selection:

- Greedy approach: remove one predictor at a time
- Similar pros/cons as forward stepwise

Recap: Subset selection – Comparison of search paths

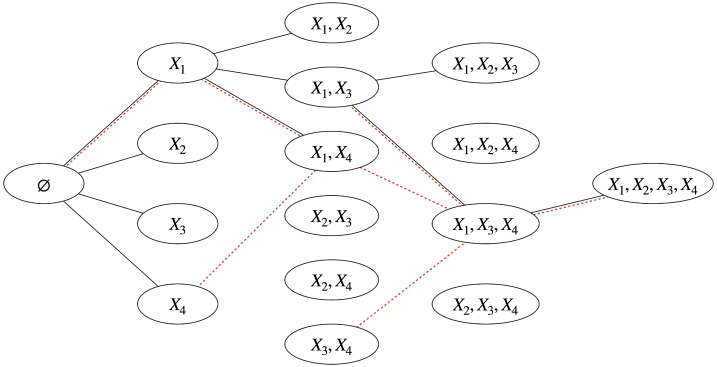


Figure: Illustration of forward stepwise (**black** solid path; successively adding $X_1 \to X_4 \to X_3 \to X_2$) and backward stepwise (**red** dashed path; removing $X_2 \to X_3 \to X_4 \to X_1$). Best subset selection checks all 2^p possibilities; all three can yield different outcomes.

For more R examples, see the discussion section slides on Canvas

Recap: Subset selection – Summary

- **Summary:**
  - **Goal**: identify a relevant subset of predictors
  - **Procedure**: evaluate subsets (all or partial), then pick best via R^2_{adj}, CV, etc.
  - BSS is exhaustive but expensive; stepwise is faster
  - Typically refit the final chosen "best" subset with least squares
- **Advantages**:
  - Direct variable selection: some $\beta_j = 0$ (excluded)
  - Straightforward implementation and intuitive interpretation
- **Disadvantages**:
  - Even stepwise can be costly if p is very large
  - Instability: small changes in the data can alter the chosen "best" subset

$\Rightarrow$ **Regularization** can handle large p, offering stable estimates without discrete exclusion

Regularization: What and why?

Recall least squares: find parameters $\hat{\beta}_0, \ldots \hat{\beta}_p$ that minimize

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{where} \quad \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_{ip} x_p$$

- **Challenges**:
  - If p is large or data are noisy, least squares solutions can be unstable
  - Overfitting (huge variance in $\hat{\beta}$) or no unique solution if $p > n$
- **Idea**: *modify* the objective by adding a penalty on β_j's to stabilize fitting
  - Balance data fidelity *vs.* "simplicity" (by favoring smaller β_j)
  - This approach is called *regularization* (or *shrinkage*)

We will learn two prominent regularization techniques for linear regression:

- **Ridge regression** (ℓ_2 penalty)
- **Lasso** (ℓ_1 penalty)

Ridge regression: 1) Formulation

Ridge regression: Find $\hat{\beta}_0, \ldots \hat{\beta}_p$ that minimize

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

- $\lambda \geq 0$ is a tuning parameter
- Each λ yields a different set of coefficient estimates $\hat{\beta}_\lambda^R$

Remarks:

- No penalty on β_0 (the intercept)
- As $\lambda \to 0$, ridge regression $\to$ standard least squares
- As λ grows, β_j shrinks toward 0
  - Reduces variance of β_j but increases bias

Ridge regression: 2) Effects of scaling

The least squares coefficients are *scale equivariant*:

- Multiplying X_j by constant c scales $\hat{\beta}_j$ by $1/c$
- Regardless of scaling, $X_j\hat{\beta}_j$ remains the same, not affecting other coefficients

Ridge regression is *not* scale-equivariant:

- Rescaling one predictor can affect others through the penalty term
- Hence, $X_j\hat{\beta}^R_{j,\lambda}$ depends on both λ *and* predictor scaling

Therefore, it is recommended to *standardize predictors* before ridge via

$$\tilde{x}_{ij} = \frac{x_{ij}}{s_{ij}} \qquad \text{where} \qquad s_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2 \quad \text{and} \quad \bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

Ridge regression: 3) Credit dataset example

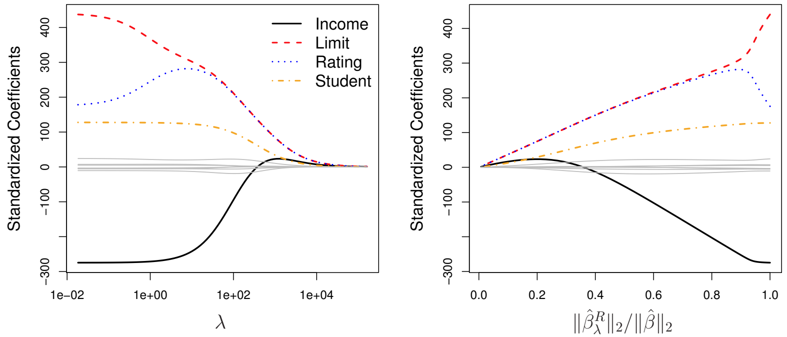


Figure: Standardized ridge coefficients for Credit, plotted vs. λ and $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$ [JWHT21, Figure 6.4]. (Note: $\|v\|_2 = \sqrt{v_1^2 + \cdots + v_p^2}$.)

- As λ increases, all β_j shrink toward 0 (none exactly zero though)
- If variables are correlated, ridge shrinks them together in a “group” manner

Ridge regression: 4) Advantages over least squares

Ridge's advantage rooted in **Bias-variance tradeoff**:

- $\lambda = 0$: no bias but high variance
- Larger λ: more bias but lower variance

When ridge can be most helpful:

- If least squares has high variance (e.g. $p \approx n$ or predictors are collinear)
- If data are noisy and β_j can fluctuate a lot
- Ridge still works even if $p > n$, producing a unique solution

Computational advantages of ridge:

- No need for 2^p model fits (as in best subset); for any λ, ridge requires only a single fit
- Indeed, we can compute solutions for all λ at near the same cost as one OLS fit

Ridge regression: 5) Visualization of bias-variance tradeoff

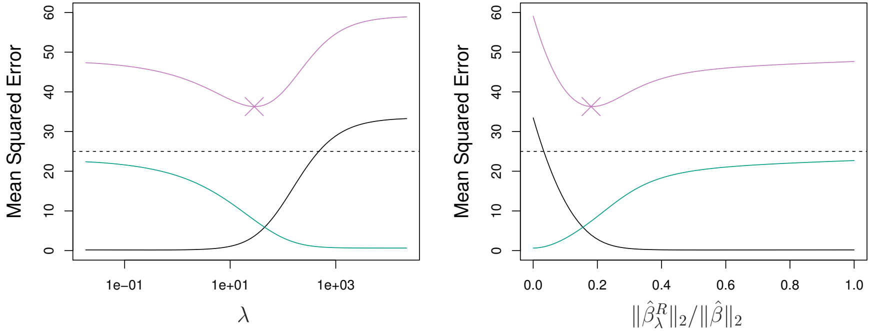


Figure: Bias-variance tradeoff in ridge for a simulated data set ($p = 45$, $n = 50$). Shown are squared bias (**black**), variance (**green**), and test MSE (**purple**) vs. λ [JWHT21, Figure 6.5].

- At $\lambda = 0$, there is no bias but high variance
- Increasing λ significantly reduces variance at the cost of slightly higher bias
- Eventually, added bias overtakes the benefit of reduced variance

Ridge regression: 6) Example with a toy dataset (R script)

- **Problem setup:**
  - $n = 5$, $p = 2$
  - Ridge vs. least squares (at $\lambda = 1$)
- **Goal:** See how ridge regression shrinks coefficients $\hat{\beta}$ compared to standard least squares

```
# Toy data: 5 obs, 2 predictors, no intercept
df <- data.frame(
  x1 = c(1,2,3,4,5),
  x2 = c(2,1,3,1,2),
  y  = c(2,2.5,6,4,6.5)
)
```

```
# OLS fit (no intercept => '-1')
ols_fit <- lm(y ~ x1 + x2 - 1, data = df)
cat("OLS Coeffs:\n", coef(ols_fit), "\n")

install.packages("glmnet") # if not installed
library(glmnet)

# Prepare X,y for glmnet
X <- as.matrix(df[,c("x1","x2")])
y <- df$y

# Ridge fit with lambda=1, no intercept
     penalization
ridge_fit <- glmnet(
  X, y, alpha=0, lambda=1,
  intercept=FALSE, standardize=TRUE
)
cat("Ridge Coeffs (lambda=1):\n", as.matrix(
     coef(ridge_fit)), "\n")
```

Ridge regression: 7) Selecting λ

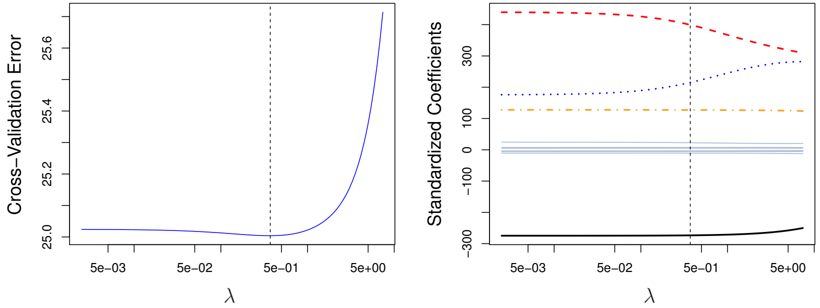


Figure: **Left:** CV errors for ridge regression on the `Credit` dataset with various values of λ. **Right:** Ridge regression coefficient estimates. The vertical dashed lines indicate the λ selected by CV [JWHT21, Figure 6.12].

- Here, the chosen λ is relatively small, meaning minimal shrinkage relative to least square
- The error curve's dip is not very pronounced, suggesting a broad range of λ values yield similar performance

Ridge regression: 8) Summary

- **Ridge regression formulation**:
  - Add a penalty term $\sum_{j=1}^{p} \beta_j^2$ to the least squares objective, scaled by $\lambda \geq 0$
  - Shrinks β_j more strongly as λ grows, stabilizing estimates
- **Bias-variance tradeoff**:
  - Larger $\lambda \Rightarrow$ higher bias but lower variance
  - Especially helpful when OLS has high variance (e.g. large p, or $p > n$)
- **Computation**:
  - Efficient to solve for all λ at roughly the cost of one OLS fit
- **Limitation**:
  - Coefficients seldom reach exactly zero $\implies$ no direct variable selection

The lasso: 1) Formulation

The lasso: Find $\hat{\beta}_0, \ldots, \hat{\beta}_p$ that minimize

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

- $\lambda \geq 0$ is a tuning parameter
- Each choice of λ gives a different set of Lasso estimates $\hat{\beta}^L_\lambda$

Remarks:

- No penalty on β_0 (the intercept)
- As $\lambda \to 0$, lasso $\to$ standard least squares
- As λ grows, many β_j shrink toward *zero*, and some exactly become 0

Key difference from ridge: Lasso can yield exact zero estimates $\Longrightarrow$ **variable selection**!

The lasso: 2) Credit dataset example

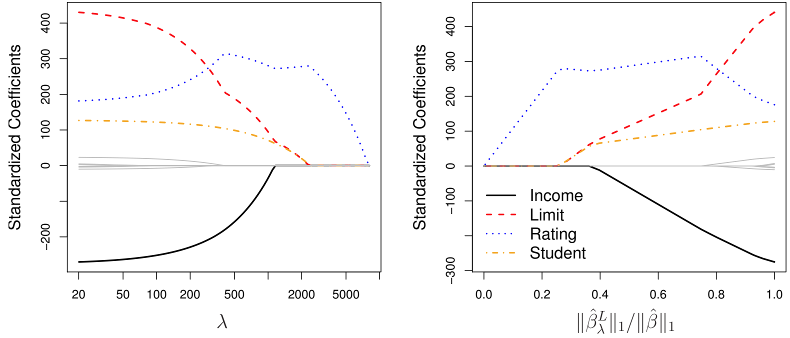


Figure: Standardized lasso coefficients for Credit, plotted vs. λ and $\|\hat\beta^R_\lambda\|_2/\|\hat\beta\|_2$ [JWHT21, Figure 6.6].

- Lasso can force some coefficients to zero as λ increases
- Achieves *variable selection* directly (predictors with $\hat\beta^L_j = 0$ are excluded)

The lasso: 3) Example with a toy dataset (R script)

- **Setup:**
  - $n = 5$, $p = 2$
  - Compare lasso vs. OLS at $\lambda = 1$
- **Objective:** See how lasso can shrink coefficients to zero

```
# Toy data: 5 obs, 2 predictors, no intercept
df <- data.frame(
  x1 = c(1,2,3,4,5),
  x2 = c(2,1,3,1,2),
  y  = c(2,2.5,6,4,6.5)
)
```

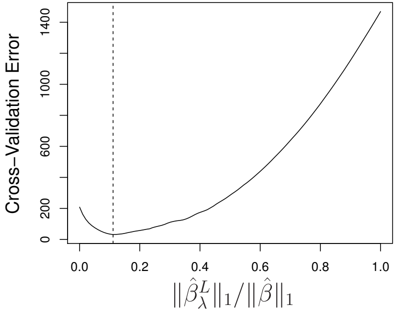
```
# OLS fit (no intercept => '-1')
ols_fit <- lm(y ~ x1 + x2 - 1, data=df)
cat("OLS Coeffs:\n", coef(ols_fit), "\n")

# If needed: install.packages("glmnet")
library(glmnet)

# Prepare X, y
X <- as.matrix(df[, c("x1","x2")])
y <- df$y

# Lasso with alpha=1, lambda=0.1
lasso_fit <- glmnet(X, y,
                    alpha=1,
                    lambda=1,
                    intercept=FALSE,
                    standardize=TRUE)
cat("Lasso Coeffs (lambda=0.1):\n",
    as.matrix(coef(lasso_fit)),"\n")
```

The lasso: 4) Selecting λ



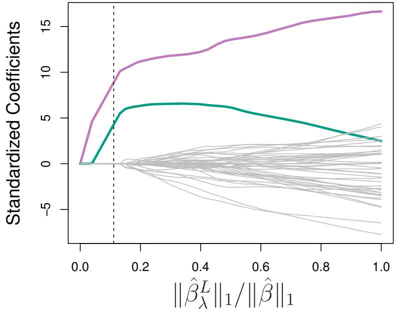


Figure: **Left:** Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data. **Right:** The corresponding lasso coefficient estimates, with the two *signal* variables in color and the *noise* variables in gray. The vertical dashed line indicates the fit that minimizes the cross-validation error [JWHT21, Figure 6.13].

- The lasso cleanly separates two *signal* variables from *noise* variables
- In contrast, standard least squares (far right, with $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1 = 1$) only identifies the purple variable without discarding the noise predictors

The lasso: 5) Summary

- **Lasso formulation**:
    - Penalty term $\sum_{j=1}^{p} |\beta_j|$ scaled by λ
    - As λ grows, some β_j become exactly zero $\implies$ variable selection
- **Lasso advantages**:
    - Encourages a *sparse* model for easier interpretability
    - Slightly more complex than ridge, but fairly efficient to solve
- **Ridge vs. lasso**:
    - Lasso can set some coefficients to *exact zeros*, while ridge never does
    - Ridge tends to be more stable especially when predictors are highly correlated
    - Both typically tune λ via cross-validation

Wrap-up & Takeaways

Subset selection:

- Great for small p, but can be expensive or unstable for large p
- Final model is refit by least squares on the chosen subset

Regularization:

- **Ridge**: ℓ_2 penalty shrinks all coefficients toward zero, good if many have modest nonzero effects
- **The lasso**: ℓ_1 penalty can set some coefficients exactly to zero (variable selection)
- **Tuning** λ typically via cross-validation for both

Next lecture:

- Geometric intuition for ridge vs. lasso
- Transition to multiple hypothesis testing

References

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
An Introduction to Statistical Learning: with Applications in R, volume 112 of *Springer Texts in Statistics*.
Springer, New York, NY, 2nd edition, 2021.