Variable Selection

PSC 8185: Machine Learning for Social Science

Iris Malone

February 14, 2022

Materials adapted from Sergio Ballacado and Rochelle Terman

Announcements

- Problem Set 3 Released - Due Feb. 28
- Problem Set 2 Extra OH Wed; Due Thurs (12pm ET)
- Reminder: Meet during OH about final project

Recap

Where We've Been:

- Class imbalance produces poor sensitivity rates
- Cross-validation provides estimate of model's (test) error (model assessment)
- Cross-validation identifies optimal tuning parameters (model selection)
- Bootstrap tells us confidence (SE) around estimate

Recap

Where We've Been:

- Class imbalance produces poor sensitivity rates
- Cross-validation provides estimate of model's (test) error (model assessment)
- Cross-validation identifies optimal tuning parameters (model selection)
- Bootstrap tells us confidence (SE) around estimate

New Terminology:

- Undersampling/oversampling
- Kappa Score
- AIC (C_p) and BIC
- Validation Set
- Cross-Validation

Agenda

1. Why Do We Need Variable Selection?
2. Subset Selection
3. Shrinkage (Regularization)
4. Dimensionality Reduction

Why Do We Need Variable Selection?

Recap: Model Assessment

Best ML model maximizes model performance

- “Good” model performance = lowest test MSE
- “Good” model also needs to performs better than No Information Rate (NIR)

Recap: Model Assessment

Best ML model maximizes model performance

- "Good" model performance = lowest test MSE
- "Good" model also needs to performs better than No Information Rate (NIR)

Building a "good" model requires:

- Lots of observations
- Lots of information about observations

Recap: Model Selection

Select best ML model by comparing lots of models:

- Model selection often depends on DGP, n obs., and p variables
- Cross-validation provides way to compare lots of different modeling specifications

Lingering Questions

- How much information does the model need?

Lingering Questions

- How much information does the model need?
- Which predictors are most important?

Information Improves Model Performance

General Rule: The more information you feed a model the better it performs. **Why?**

Information Improves Model Performance

General Rule: The more information you feed a model the better it performs. **Why?**

More information → more learning → better pattern recognition

Implication: Feed as many predictors p to model as possible?

Example

Motivation: Teach model to predict images of cats versus dogs starting with 1 predictor

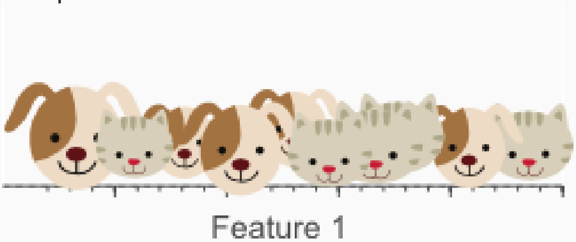


Figure 1: A single feature (5 bins) does not result in a good separation of our training data (5 cats, 5 dogs). More information required.

Source: Vision Dummy (2014)

Example

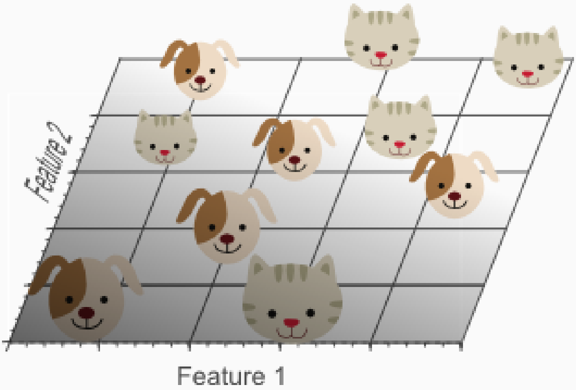


Figure 2: Adding a second feature ($5x5 = 25$ bins) still does not result in a linearly separable classification problem: No single line can separate all cats from all dogs in this example.

Example

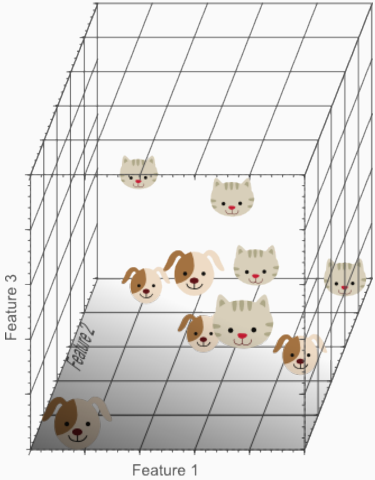


Figure 3: Adding a third feature ($5x5x5 = 125$ bins) results in a linearly separable classification problem in our example. A plane exists that perfectly separates dogs from cats.

Example

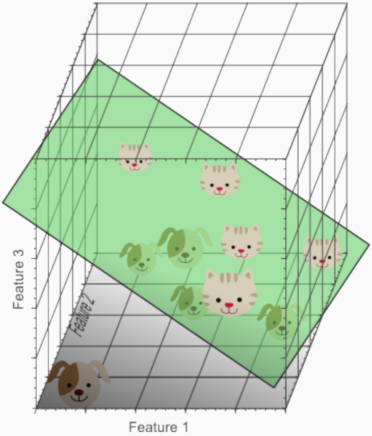


Figure 4: The more features we use, the higher the likelihood that we can successfully separate the classes perfectly.

Implication: Adding more features improves separation? **No!**

Problem: Curse of Dimensionality

Main Idea: Lots of p can lead to **overfitting**

Problem: Curse of Dimensionality

Main Idea: Lots of p can lead to **overfitting**

Reasoning:

- ML methods look for similar observations in various regions of space, e.g. KNN
- As dimensionality (number of variables) grows, there are fewer observations per region → well-separated classes
- High dimensional data → overfitting

Example of Overfitting

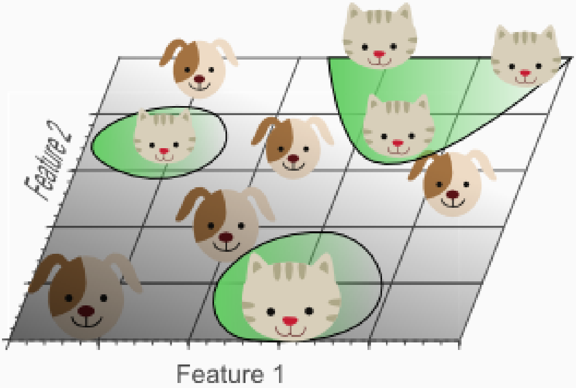


Figure 5: Using too many features results in overfitting due to sparsity of data (10 pets ↔ 125 bins) The classifier starts learning exceptions that are specific to the training data and do not generalize well when new data is encountered.

Risks of Overfitting

1. Poor out-of-sample performance (no external validity)
2. Risk of false positives (spurious correlation)

Solutions to Overfitting

1. Add More Data

Solutions to Overfitting

1. Add More Data but
    - Data requirements grow exponentially as number parameters increase
    - Computationally expensive

Solutions to Overfitting

1. Add More Data but
    - Data requirements grow exponentially as number parameters increase
    - Computationally expensive
2. Add Fewer Variables

Solutions to Overfitting

1. Add More Data but
   - Data requirements grow exponentially as number parameters increase
   - Computationally expensive
2. Add Fewer Variables but
   - May impede model performance
   - Subjective determination of ‘best’ variables
   - Risk overlooking important interactions

Solutions to Overfitting

1. Add More Data but
   - Data requirements grow exponentially as number parameters increase
   - Computationally expensive
2. Add Fewer Variables but
   - May impede model performance
   - Subjective determination of ‘best’ variables
   - Risk overlooking important interactions
3. Variable Selection

3 Variable Selection Techniques

1. Subset Selection
2. Shrinkage (Regularization)
3. Dimensionality Reduction

Variable Selection

Main Idea: Pick optimal number and/or type of variables to maximize model performance

1. Subset Selection
2. Shrinkage (Regularization)
3. Dimensionality Reduction

Variable Selection

Main Idea: Pick optimal number and/or type of variables to maximize model performance

1. Subset Selection
   - Compare models of varying complexity
   - Pick optimal number of features based on best RSS
   - Fit model using reduced set of variables
2. Shrinkage (Regularization)
3. Dimensionality Reduction

Variable Selection

Main Idea: Pick optimal number and/or type of variables to maximize model performance

1. Subset Selection
2. Shrinkage (Regularization)
    - Keep all features, but shrink value of parameters close to zero (ridge)
    - Keep all features, but shrink value of (some) parameters to zero (lasso)
3. Dimensionality Reduction

Variable Selection

Main Idea: Pick optimal number and/or type of variables to maximize model performance

1. Subset Selection
2. Shrinkage (Regularization)
3. Dimensionality Reduction
    - Identify related features in similar regions of space
    - Collapse related features into single linear combination or projection M
    - Fit model using reduced set of projections M

Subset Selection

Best Subset Selection

Main Idea: Compare models with varying number of $k \in [0, p]$ predictors and pick model with best performance

Estimation Goal: Identify optimal k number of predictors

Procedure:

- Identify $\binom{p}{k} = \frac{p!}{k!(p-k)!}$ possible models
- Run model for every possible k
- Choose model with lowest RSS

Example Subset Selection: Credit Card Debt

Motivation:

- Want to predict level of credit card debt based on age, gender, student status, race, etc.
- Need to know key characteristics to target future credit approval

Baseline Approach:

- Fit different model combinations with varying k predictors
- Estimate RSS and R^2 for each model
- Choose model with lowest RSS

Example: Credit Dataset

Problem: Training RSS and R^2 always increase as we increase k

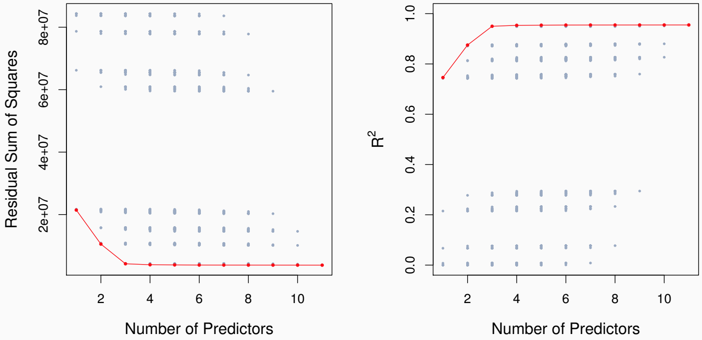


Figure 6: Training RSS and R^2 for increasingly complex models (Fig. 6.1)

Model Optimization

- Need to minimize the test (validation) error, not the training error
- Assess lowest test error using assessment tool kit:
  - AIC
  - BIC
  - Adjusted R^2
  - Cross-Validation

Model Optimization

In practice, we use AIC (C_p), BIC, and adjusted R^2 frequently used over cross-validation.

Why?

- Less expensive to compute
- Better asymptotics with large n
- Extends nicely to non-linear models (e.g. logistic regression)

Common k Selection Techniques: AIC, BIC, Adjusted R^2

Example: Credit Data

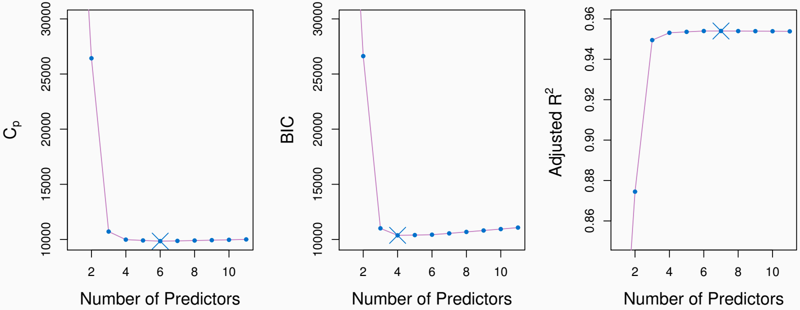


Figure 7: Non-monotonic model performance with test data (Fig 6.2)

Alternative Selection Technique: Cross-Validation of k

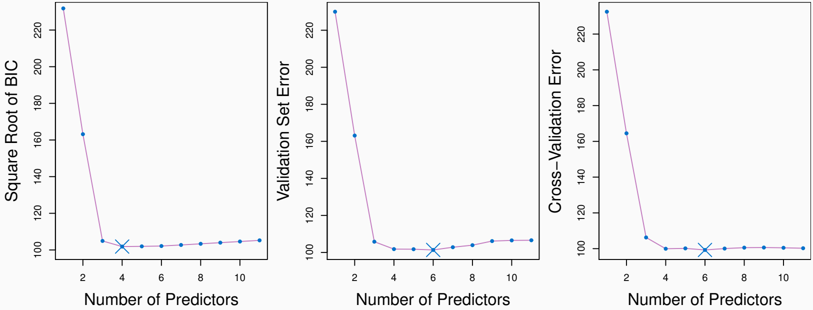


Figure 8: Non-monotonic model performance with test data (Fig 6.3)

Limits to Best Subset Selection

Problems:

- Very expensive computationally → have to fit 2^p models
- Selected model can still have high variance
- If there are too many model combination possibilities of $\binom{k}{p}$ we once again increase our chance of overfitting.
- Bias-variance tradeoff problems

Bias-Variance Trade-Off Problems

Given subset selection limits, we could restrict our search space (e.g. $k \in [0, \bar{p}]$) for the best model

Bias-Variance Trade-Off Problems

Given subset selection limits, we could restrict our search space (e.g. $k \in [0, \bar{p}]$) for the best model

- Bounding reduces the **variance** of the selected model ☺
- But also increases the model **bias** ☹

Alternatives to Best Subset Selection

- Forward Selection
- Backward Selection
- Hybrid Selection
- Mixed stepwise selection
- Forward stagewise selection

Forward Selection

- **Main Idea:** Start with a model containing no predictors and iteratively increase its complexity by adding one variable at a time.
- Identify most important variable based on how well its addition improves model fit (**mean increase in accuracy**)
- Allows $p > n$

Backward Selection

- **Main Idea:** Start with a model containing <u>all</u> predictors and iteratively decrease its complexity by removing one variable at a time
- Identify most important variable based on how well its removal worsens model fit (**mean <u>decrease</u> in accuracy**)
- Requires $p < n$

Example: Forward Selection vs Backward Selection

- Motivation: Estimate relationship between Y and $[X_1, X_2, X_3]$
- Assume $X_1, X_2 \sim N(0, \sigma)$ independent
- Procedure:
  - Regress Y onto X_1, X_2, X_3
  - Perform Different Subset Selection
- Forward Selection Starting Estimate:

$$\hat{Y} = \beta_3 X_3$$

- Backward Selection Starting Estimate:

$$\hat{Y} = \beta_3 X_3 + \beta_2 X_2 + \beta_1 X_1$$

Example: Forward Selection vs Backward Selection

- True DGP:

$$X_3 = X_1 + 3X_2$$
$$Y = X_1 + 2X_2 + \epsilon$$
$$Y = X_3 + X_2 + \epsilon$$

- Different Selection Techniques → Different Variable Importance
- Identify Most Relevant Predictors:
  - Forward:
    - $X_3 \to X_3, X_2 \to X_3, X_2, X_1$
    - Optimal = X_3, X_2
  - Backward:
    - $X_3, X_2, X_1 \to X_1, X_2 \to X_2$
    - Optimal = X_1, X_2

Advantages and Disadvantages to Subset Selection

Advantages

Disadvantages

Advantages and Disadvantages to Subset Selection

Advantages

- Popular in 1980s/1990s
- Straightforward algorithm
- Performs variable selection

Disadvantages

Advantages and Disadvantages to Subset Selection

Advantages

- Popular in 1980s/1990s
- Straightforward algorithm
- Performs variable selection

Disadvantages

- Not guaranteed to yield best model
- Variable input sequence yields different results
- Can miss interactions between variables
- Risk high variance models

Shrinkage (Regularization)

Shrinkage Methods

Main Idea: Estimate a model with all predictors p and shrink irrelevant coefficients $\hat{\beta}$ to 0

Why Shrink?

- Collinearity or $p > n$ creates high variance (unstable) models
- Shrinking introduces bias (if true $\beta > 0$), but can decrease variance of estimates. When latter effect is larger, this decreases overall test error

Two Types of Shrinkage Methods

- **Ridge Regression:** Keep all features, but shrink value of parameters close to zero
- **LASSO (lasso):** Keep all features, but shrink value of (some) parameters to zero

Ridge Regression

Main Idea: Estimate linear regression, but add a **shrinkage penalty** that reduces parameter size to minimize error

Ridge Regression

Main Idea: Estimate linear regression, but add a **shrinkage penalty** that reduces parameter size to minimize error

Recall OLS Loss Function:

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

Ridge Regression

Main Idea: Estimate linear regression, but add a **shrinkage penalty** that reduces parameter size to minimize error

Recall OLS Loss Function:

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

Ridge Regression Loss Function:

$$\text{RSS} + \lambda \sum_{j=1}^{p}\beta_j^2$$

Ridge Regression Loss Function

Loss Function:

$$\text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge Regression Loss Function

Loss Function:

$$\text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Shrinkage Penalty (ℓ_2 norm aka $\lambda \sum_{j=1}^{p} \beta_j^2$)

- Regulates size of loss function by shrinking parameter size
- Small penalty (less shrinkage) when ...
  - $\beta_1, \ldots, \beta_p$ already close to zero (true $\beta \approx 0$)
  - λ close to zero

Finding Optimal λ

The parameter λ is a **tuning parameter**:

- $\lambda = 0$ means no penalty $\rightarrow$ OLS estimates $\hat{\beta}_j$
- $\lambda = \infty$ means high penalty $\rightarrow$ $\hat{\beta}_j \approx 0$

Finding Optimal λ

The parameter λ modulates the importance of fit (variance) vs coefficient shrinkage (bias). Need to minimize bias-variance tradeoff.

How to Choose?

- Estimate $\hat{\beta}$ for many values of λ
- Choose optimal λ by cross-validation

Example Ridge Regression

Ridge regression of default in the Credit dataset

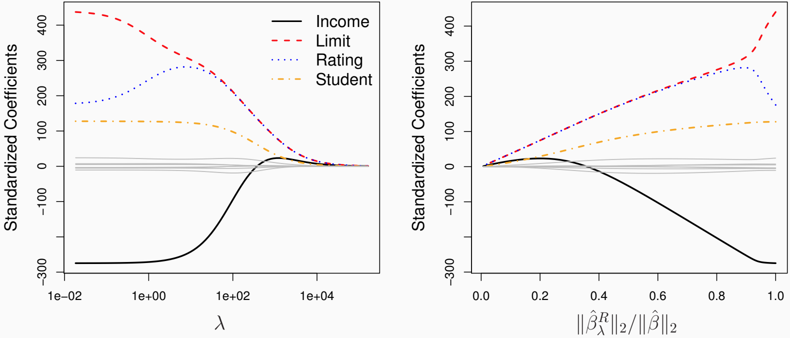


Figure 9: Coefficients as function of λ and distance of β from zero (ℓ_2 norm) (Fig 6.4)

As λ increases the ℓ_2 norm of $\hat{\beta}_\lambda$ decreases

Bias-Variance Tradeoff of λ

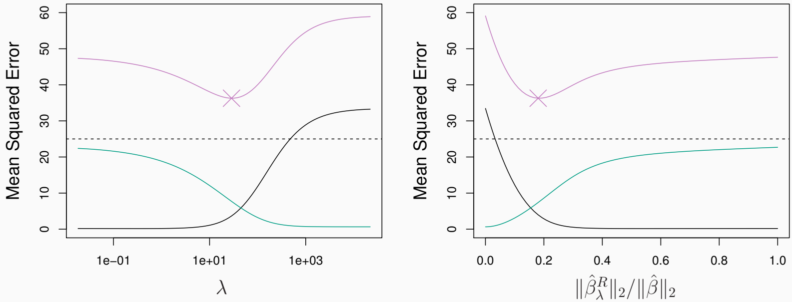


Figure 10: Shrinking can introduce bias if true $\beta > 0$, but can also decrease variance of estimates. When latter effect dominates, overall MSE decreases. (Fig 6.5)

Advantages and Disadvantages to Ridge Regression

Advantages

Disadvantages

Advantages and Disadvantages to Ridge Regression

Advantages

- Performs well when $p > n$ or lots of collinearity
- Faster than best subset selection

Disadvantages

Advantages and Disadvantages to Ridge Regression

Advantages

- Performs well when $p > n$ or lots of collinearity
- Faster than best subset selection

Disadvantages

- No variable selection $\rightarrow$ includes all predictors
- All predictors $\rightarrow$ reduces model interpretability
- Performs poorly if true f non-linear

Lasso Regression

LASSO: “least absolute shrinkage and selection operator”

Main Idea: Like ridge regression, but with shrinkage penalty that reduces some parameter sizes to zero

Lasso Loss Function:

$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j^2|$$

We call $\lambda \sum_{j=1}^{p} |\beta_j^2|$ the ℓ_1 norm.

Tl;DR: Lasso shrinks *some* coefficients to zero and keeps others intact.

Example of Lasso Regression

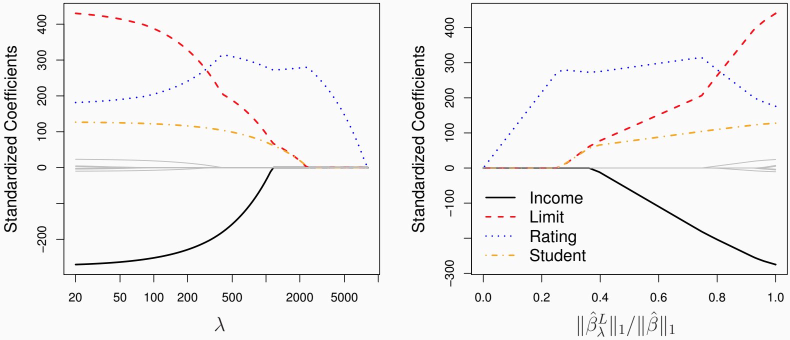


Figure 11: Coefficients as function of λ and distance of β from zero (ℓ_2 norm) (Fig 6.6)

As λ increases the ℓ_1 norm of $\hat{\beta}_\lambda$ decreases

Comparison of Loss Functions

- OLS

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

- Ridge (L2 Regularization)

$$\text{RSS} + \lambda \sum_{j=1}^{p}\beta_j^2$$

- Lasso (L1 Regularization)

$$\text{RSS} + \lambda \sum_{j=1}^{p}|\beta_j^2|$$

Why does Lasso Shrink to Zero?

Constraint space (where $\beta = 0$) is larger for lasso

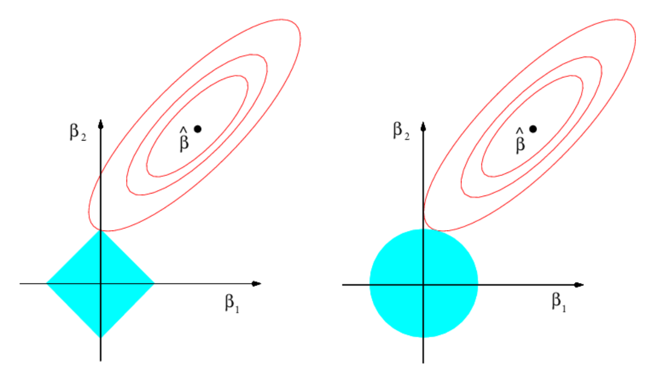


Figure 12: Compare lasso (square) to ridge (circle) constraint regions. Red ellipses contours of RSS. Assume budget s for size of constraints. If s large enough, then blue space contains red ellipses. If s small, then coef given by first point at which ellipse contacts constraint region.

Why does Lasso Shrink to Zero?

Constraint space (where $\beta = 0$) is larger for lasso

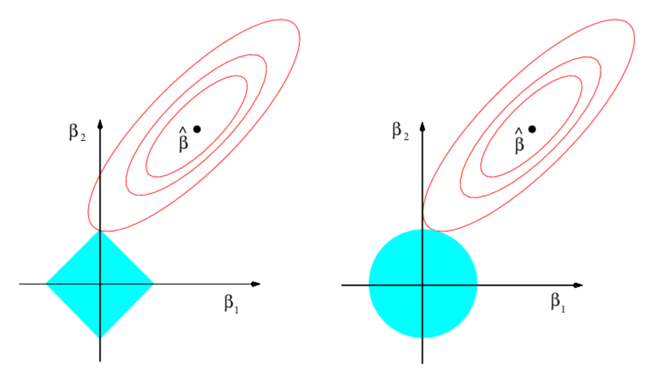


Figure 13: There are “corners” in the lasso constraint. If the sum of squares “hits” one of these corners, then the coefficient corresponding to the axis is shrunk to zero. As p increases, the multidimensional diamond has an increasing number of corners, and so it is highly likely that some coefficients will be set equal to zero.

When to choose Lasso vs Ridge?

Case 1: If most coefficients are non-zero → prefer ridge

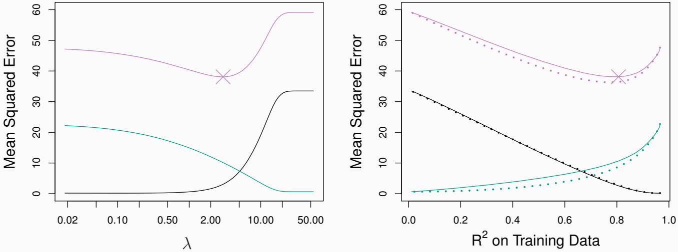


Figure 14: Plot of simulated data comparing squared bias (black), variance (green), and test MSE (purple) for lasso (solid) vs ridge (dashed) (Fig 6.8)

Key Takeaway: Bias is about same for both methods. Variance and MSE smaller for ridge.

When to choose Lasso vs Ridge?

Case 2: If only a few coefficients are non-zero → prefer lasso

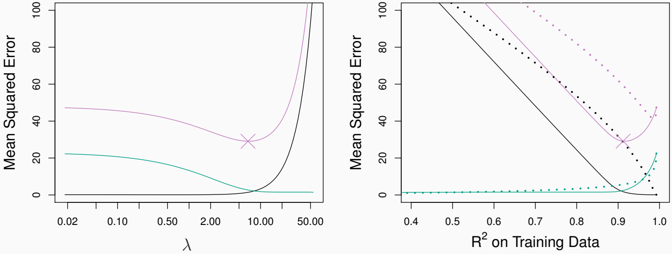


Figure 15: Plot of squared bias (black), variance (green), and test MSE (purple) for lasso (solid) vs ridge (dashed) (Fig 6.9)

Key Takeaway: Bias, variance, and MSE lower for the lasso.

Advantages and Disadvantages to Lasso

Advantages

- Performs inference and prediction
- Performs variable selection → excludes irrelevant variables by shrinking β
- More parsimonious models
- Easier to interpret

Disadvantages

- Performs worse than ridge if most variables unrelated to outcome
- Performs poorly if true f non-linear

Dimensionality Reduction

High-Dimensional Data

- Most of the methods we've discussed work best when n is much larger than p

High-Dimensional Data

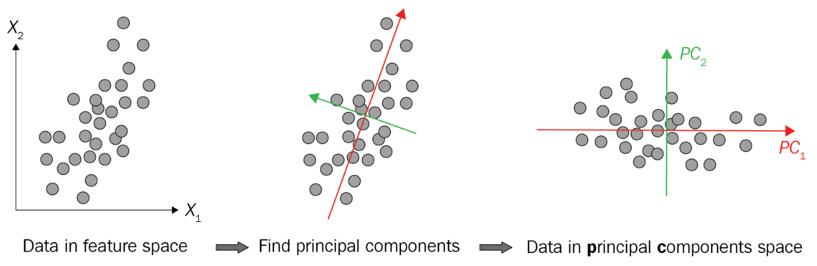
- Most of the methods we've discussed work best when n is much larger than p
- However $p >> n$ is now common due to experimental advances and cheaper computers

High-Dimensional Data

- Most of the methods we've discussed work best when n is much larger than p
- However $p >> n$ is now common due to experimental advances and cheaper computers
- Examples:
    - Medicine: Predict heart disease using clinical observations (blood pressure, salt consumption, age) plus 500,000 single nucleotide polymorphisms
    - Marketing: Predict online shopping patterns using search terms (number words in dictionary)

Dimensionality Reduction

Main Idea: Define a small set of M predictors which summarize the information in all p predictors.



Principal Component Regression (PCR)

Main Idea: Estimate linear regression using M predictors

Procedure:

- Identify similarities between groups of predictors $X_1, X_2, \ldots, X_p$
- Transform groups of predictors into M linear combination known as **principal component** z

$$z_m = \sum_{i=1}^{p} \phi_{jm} X_j$$

- Re-estimate linear regression using smaller ($M < p$) components to get coefficients Θ

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i$$

Limits to Principal Component Regression

- Unsupervised method: incorporates no information about response
- Performs poorly if data not standardized/normalized
  - PCR is variance-maximizing algorithm
  - Will weight high variance predictors over low variance predictors
  - Unscaled data can skew results (suggest only 1 variable important)

Example of Principal Component Regression

Determine optimal M using cross-validation:

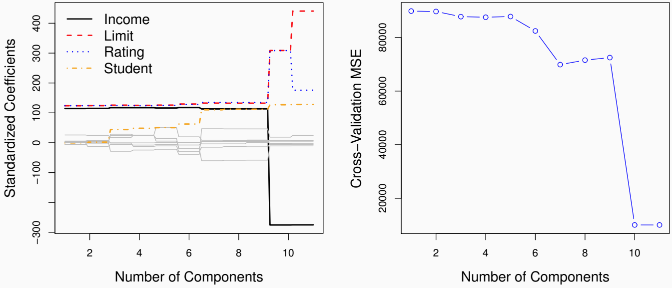


Figure 16: PCR coefficient estimates for different M; ten-fold CV of test MSE using different PCR M

Coefficients shrink as we decrease number of linear combinations M

Partial Least Squares (PLS)

Main Idea: Try to find the linear combination of predictors M that is most highly correlated with the response

Procedure:

- Identify new set of features $Z_1, \ldots, Z_m$ that are linear combinations of predictors
- Assess how well different features predict y
- Weight variables most strongly related to the response
- Use weighted features Z to re-estimate linear regression

Advantages and Disadvantages to PLS

Advantages

Disadvantages

Advantages and Disadvantages to PLS

Advantages

- Accounts for response variable (supervised learning problem)
- Performs comparable to ridge regression/PCR
- Sometimes less bias than PCR

Disadvantages

Advantages and Disadvantages to PLS

Advantages

- Accounts for response variable (supervised learning problem)
- Performs comparable to ridge regression/PCR
- Sometimes less bias than PCR

Disadvantages

- Performs poorly if data non-standardized
- Sometimes worse variance than PCR
- Tendency to overfit

Conclusion

- Variable selection can improve model performance
- Problem of overfitting:
  - Curse of dimensionality
  - Resolve by reducing model variance (decreasing p)
- Perform variable selection using subset selection, shrinkage, or dimensionality reduction techniques
- Use cross-validation to determine tuning parameters