Data, Environment and Society: Lecture 15: Model Selection and Regularization, continued

Instructor: Duncan Callaway
GSI: Salma Elmallah

October 22, 2019

Plagiarism

"Plagiarism is defined as use of intellectual material produced by another person without acknowledging its source, for example:

- **Wholesale copying of passages from works of others into your homework, essay, term paper, or dissertation without acknowledgment.**
- Use of the views, opinions, or insights of another without acknowledgment.
- Paraphrasing of another persons characteristic or original phraseology, metaphor, or other literary device without acknowledgment."

(source: Berkeley Division of Student Affairs)

Recap lecture objectives from last time

- Refine our understanding of model identification as an optimization problem

$$\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot R(\beta)$$

Important: We drop the λ term for prediction, i.e. predictions are just

$$\hat{y}_i = X_i\hat{\beta}$$

where X_i and $\hat{\beta}$ are vectors

- Understand what "regularization" is and why we do it
  - A tool for adapting optimization problems to be "well behaved"
  - In statistical learning, a tool to tradeoff bias and variance

But note, R causes you to solve a different problem than the original $\rightarrow$ parameter bias

Recap lecture objectives from last time, ctd

- Continue thinking about how to adjust errors to compare models with different p
  - k-fold cross validation, AIC, BIC, adjusted R^2...
- Learn the tradeoffs between subset selection, ridge and lasso
  - Speed (fastest to slowest): Ridge, Lasso, Subset
  - Subset selection and Lasso do feature selection. Ridge does not.
  - You can naturally tune prediction bias-variance with Ridge and Lasso

Today's objectives

1. Quick review of the basic mechanics of Subset selection, Ridge and Lasso.
2. Build deeper intuition on how they work and how they differ.
3. Learn how the bias-variance tradeoff gets tuned with regularization term parameters.
4. Understand the tradeoffs between these methods in more detail
5. Understand the importance of standardizing your variables.
6. Epilogue: the elastic net, a machine learning mashup.

Objective 1

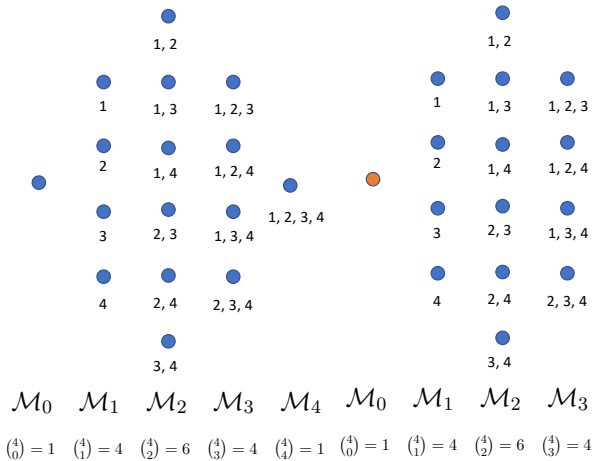
Some quick review.

Stepwise selection: Forward Selection

Forward selection: Start with $\mathcal{M}_0$. Then to choose the best model from each higher "level":

First, within each level, add one predictor at a time to the best model from the lower level. Use R^2 or other to find the best model from this set of $\mathcal{M}_{k-1}$ "$+1$"

Second, choose from your list $\{\mathcal{M}_0, \mathcal{M}_1, ...\mathcal{M}_p\}$ via cross validation or adjusted error metrics.



A bit more recap

$$\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot R(\beta)$$

The regularizing term, $R(\beta)$ can take a lot of forms. We looked at

$$R(\beta) = \sum_{i=1}^{K} I(\beta_i) \qquad = \|\beta\|_0 \quad \text{(subset selection)}$$

$$R(\beta) = \sum_{i=1}^{K} |\beta_i| \qquad = \|\beta\|_1 \quad \text{(lasso)}$$

$$R(\beta) = \sum_{i=1}^{K} \beta_i^2 \qquad = \|\beta\|_2^2 \quad \text{(ridge)}$$

Note, last time I referred to $\sum_{i=1}^{K} \beta^2$ as the 2-norm. It's not! ($\|\beta\|_2 = \sqrt{\sum_{k=1}^{K} \beta^2}$ is.)

Side note: "Regularization"

Regularization Refers to the process of adding a term to the objective function of a problem that

- Makes the problem "well behaved" (easier to solve)
- In statistical learning, allows us to tune model flexibility
- Solves a different problem from the one you originally wanted.

In our case, the sum of squared coefficients in Ridge makes the problem very simple to solve, but we get coefficients that are biased.

Objective 2

Building deeper intuition on how these methods work and how they differ.

Identifying parameters

$$\hat{\beta}_{\text{ols}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\left(\mathbf{X}^T\mathbf{Y}\right)$$
$$\hat{\beta}_{\text{ridge}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_k\right)^{-1}\left(\mathbf{X}^T\mathbf{Y}\right)$$
$$\hat{\beta}_{\text{lasso}} = \text{Something you need to solve numerically}$$
$$\text{(with something like gradient descent)}$$

Here

- λ is a tuning parameter – it is not unique.
- $\mathbf{I}_k$ is the $k \times k$ identity matrix

Important!

- Ridge and Lasso produce different β estimates for different choices of λ.
- The λ term is not involved in prediction. Predictions are just $\hat{y}_i = X_i\hat{\beta}$

Regularized problems can be converted to constrained problems

$$\text{Subset selection:} \quad \min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot \sum_{k=1}^{K} I(\beta_k) \quad \Leftrightarrow \quad \begin{cases} \min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 \\ \text{subject to} \\ \sum_{k=1}^{K} I(\beta_k) \leq s \end{cases}$$

Important:

- λ and s are parameters that need to be tuned.
- Increasing λ has the same effect as decreasing s. (Forces selection of fewer features.)
- λ and s are not independent.
- I'm not deriving this result, but I want the intuition to be clear – ask questions if needed!

Regularized problems can be converted to constrained problems

Subset selection: $\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot \sum_{k=1}^{K} I(\beta_k) \quad \Leftrightarrow \begin{cases} \min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 \\ \text{subject to} \\ \sum_{k=1}^{K} I(\beta_k) \leq s \end{cases}$

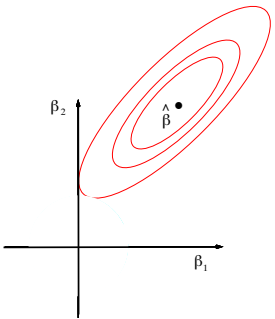
Lasso: $\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot \sum_{k=1}^{K} |\beta_k| \quad \Leftrightarrow \begin{cases} \min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 \\ \text{subject to} \\ \sum_{k=1}^{K} |\beta_k| \leq s \end{cases}$

Ridge: $\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot \sum_{k=1}^{K} \beta_k^2 \quad \Leftrightarrow \begin{cases} \min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 \\ \text{subject to} \\ \sum_{k=1}^{K} \beta_k^2 \leq s \end{cases}$

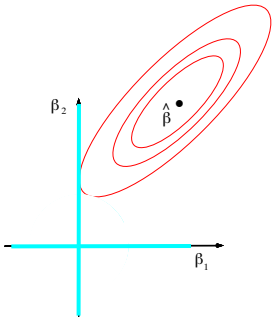
Setting intuition about constrained problems

Imagine a simple two-feature problem.

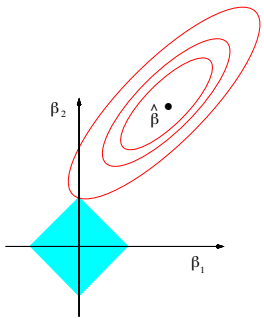
No constraints. Red lines are "constant RSS" contours



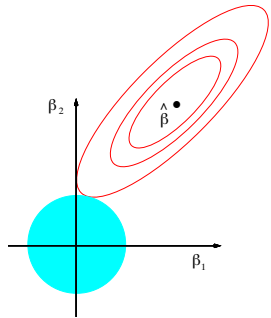
Subset selection. If $\sum_{k=1}^{K} I(\beta_k) \leq 1$, solutions in blue area.



Lasso. Solutions must be in blue area.



Ridge. Solutions must be in blue area.



Figures adapted from Elements of Statistical Learning

How do the parameter estimates compare?

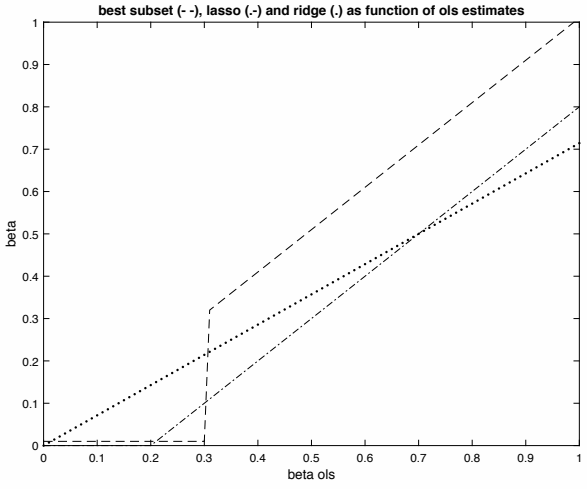


Figure taken from Imbens 2015 NBER lecture

Model: $y_i = \beta x_i + \epsilon_i$

Lines are the result of increasing the "true" β, then comparing estimates from different loss functions.

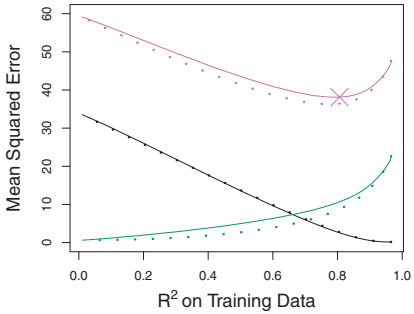
We often call regularization approaches "shrinkage" methods because, in general, they smoosh parameters closer to zero.

Objective 3

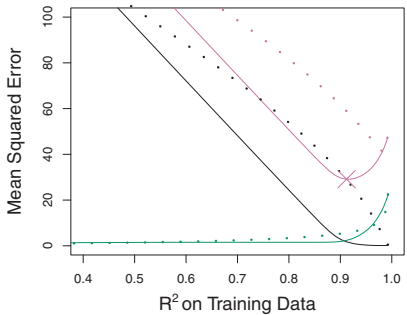
Learn how the bias-variance tradeoff gets tuned with regularization term parameters.

Lasso or ridge?

Simulated data set from $n = 50$ and $p = 45$. Figure shows test MSE on y-axis. Each point on the curve corresponds to a different λ.



The actual response is a function of **all 45 predictors**.

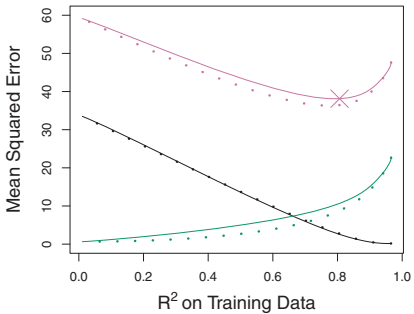


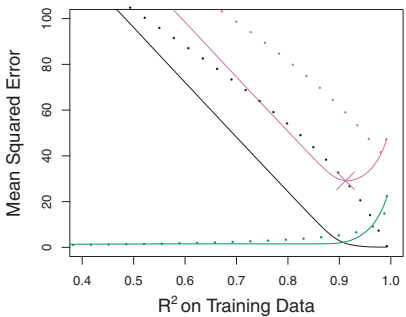
The actual response is a function of **only 2 of the 45 predictors.**

Red show total MSE. (Black and green are variance and bias.) Dashed line – best Ridge. Solid line – best lasso.

Lasso or ridge – a general trend

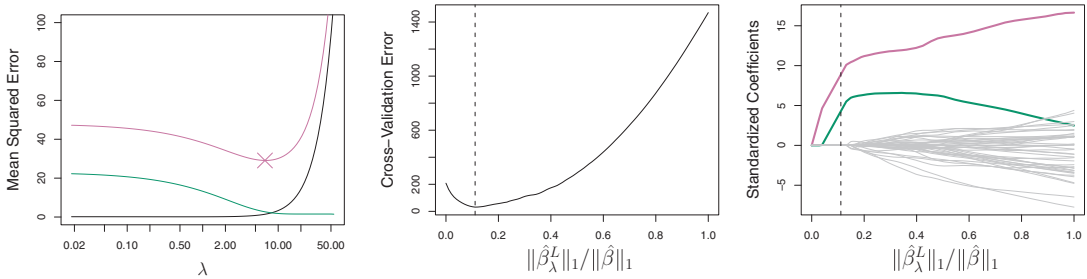
Ridge works best when the actual number of features is high. Lasso works best when the actual number of features is low.





Choosing λ

As we've seen, λ can take a range of values. Here we see the tradeoff for the 2-of-45 predictors example lasso example from the last slide.



Simple λ selection strategy: use k-fold cross validation to test performance on a grid of λ values.

Regularize for parameter estimation only! y-axis errors above are just $\sum_i (y_i - X_i\hat{\beta})^2$

Objective 4

Understand the tradeoffs between these methods in more detail

Ridge regression advantages over OLS

First. Suppose some of your features are linear combinations of the others. That means you can write $x_{i,j} = Ax_{i,-j}$ for at least 1 value of j.

Then $\mathbf{X}^T\mathbf{X}$ is not "full rank" and you can't invert it. I.e., $(\mathbf{X}^T\mathbf{X})^{-1}$ doesn't exist.

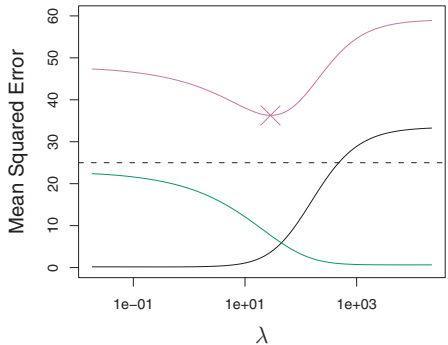
$$\text{e.g., } \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \quad \text{vs.} \quad \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} = \begin{bmatrix} 1+\lambda & 2 & 3 \\ 2 & 4+\lambda & 6 \\ 3 & 6 & 9+\lambda \end{bmatrix}$$

But you *can* invert $\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_k\right)$. (Since you've added a little shift to the diagonals of the matrix, which restores linear independence)

Second. Computation! It's faster than subset selection (but solves a different problem if your objective is parameter interpretation).

Ridge regression advantages over OLS, ctd

Third. Bias-variance tradeoff! Figure from ISLR



green: variance

black: bias

red: total error

...but how can we choose λ? The short answer is k-fold cross validation.

Model selection tradeoffs

	Subset selection	Ridge	Lasso
Computing time	high	very low	low
Drives parameters to zero?	yes	no	yes
Parameters biased relative to "true" model?	no	yes	yes
Handles correlated features well?	yes	yes	no
Interpretability?	size of parameters	not really	presence of parameters

Lasso and ridge:

+ less prediction variance than OLS, especially with many predictors
− more prediction bias than OLS

With highly correlated predictors, Lasso is unstable: indifferent between

- $\hat{\beta}_1 = 0$ and $\hat{\beta}_2 = \beta_1 + \beta_2$
- $\hat{\beta}_1 = \beta_1 + \beta_2$ and $\hat{\beta}_2 = 0$

Ok, so which should I use?

You won't know ahead of time if your model *actually* needs a lot of features or not.

...and it's very easy to re-run a specification with Ridge, then Lasso.

...so you might as well try them both.

Subset selection can be computationally more prohibitive, so you may not be able to run it, depending on your computing environment.

Objective 5. First, a thought experiment.

Consider a simple example: Suppose you want to build a predictive model for ER visits in the East Bay

- Dependent variable, y: Daily ER visits at all hospitals in Oakland.
- Features:
  - X_1, Particulate matter concentration from EPA's downtown Oakland sensor, in g/m3
  - X_2, Average daily temperture at the Oakland Airport.

You use the model:

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}$$

Suppose you fit the model with OLS as well as ridge regression (RR), with $\lambda = 0.1$.

Thought experiment, continued

- Dependent variable, y: Daily ER visits at all hospitals in Oakland.
- Features:
  - x_1, Particulate matter concentration from EPA's downtown Oakland sensor, in g/m3
  - x_2, Average daily temperture at the Oakland Airport.

You realize you want to convert x_1 to μg/m3.

Which model will yield different predictions with the transformed data? The one fit with ridge or the one with OLS? Answer: **Ridge!**

In OLS, β_1 grows by 10^6 to balance out the re-scaled of x_1.

With ridge, λ will squash β_1, meaning it won't grow as much as it did with OLS. The predictions will change.

$\Rightarrow$ Objective 5: Understand that you need to standardize variables

For ridge **and** lasso, it's important to "standardize" your variables:

$$x' = \frac{x}{\sigma_x} \tag{1}$$

...and then fit the model to the normalized values.

Any guesses why?

Reason: This way variables with large range don't dominate the solution.

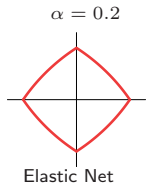
Hot topic: Elastic nets...

...are cool!

Elastic nets

- Drive parameters to zero like lasso
- Deals with correlated predictors well, like ridge (by shinking them together)
- Give you another !&@!%* parameter to tune
- Still aren't always best – good to try several shrinkage methods, not just this.

$$\lambda \sum_{k=1}^{K} \left(\alpha\beta_j^2 + (1-\alpha)|\beta_j|\right)$$



from Elements of Statistical Learning