Regularization

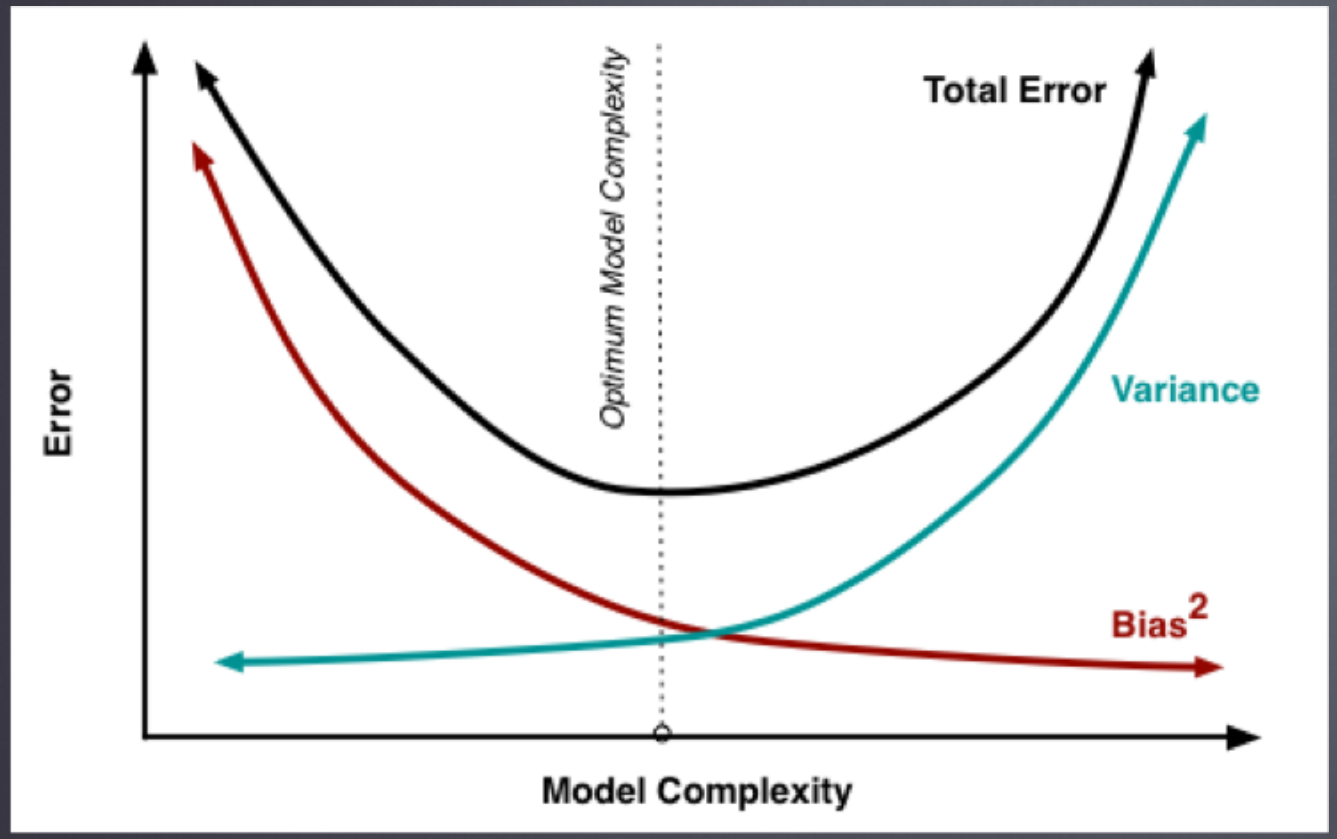
METIS

Learning Goals

- Understand regularization as a practical strategy for reducing overfitting by punishing model complexity
- Correctly describe the relationship between regularization and the model complexity tradeoff
- Build mathematical intuition for how regularization works
  - Slides include appendix that goes further into the math if you're interested!

Review: the Bias-Variance Tradeoff

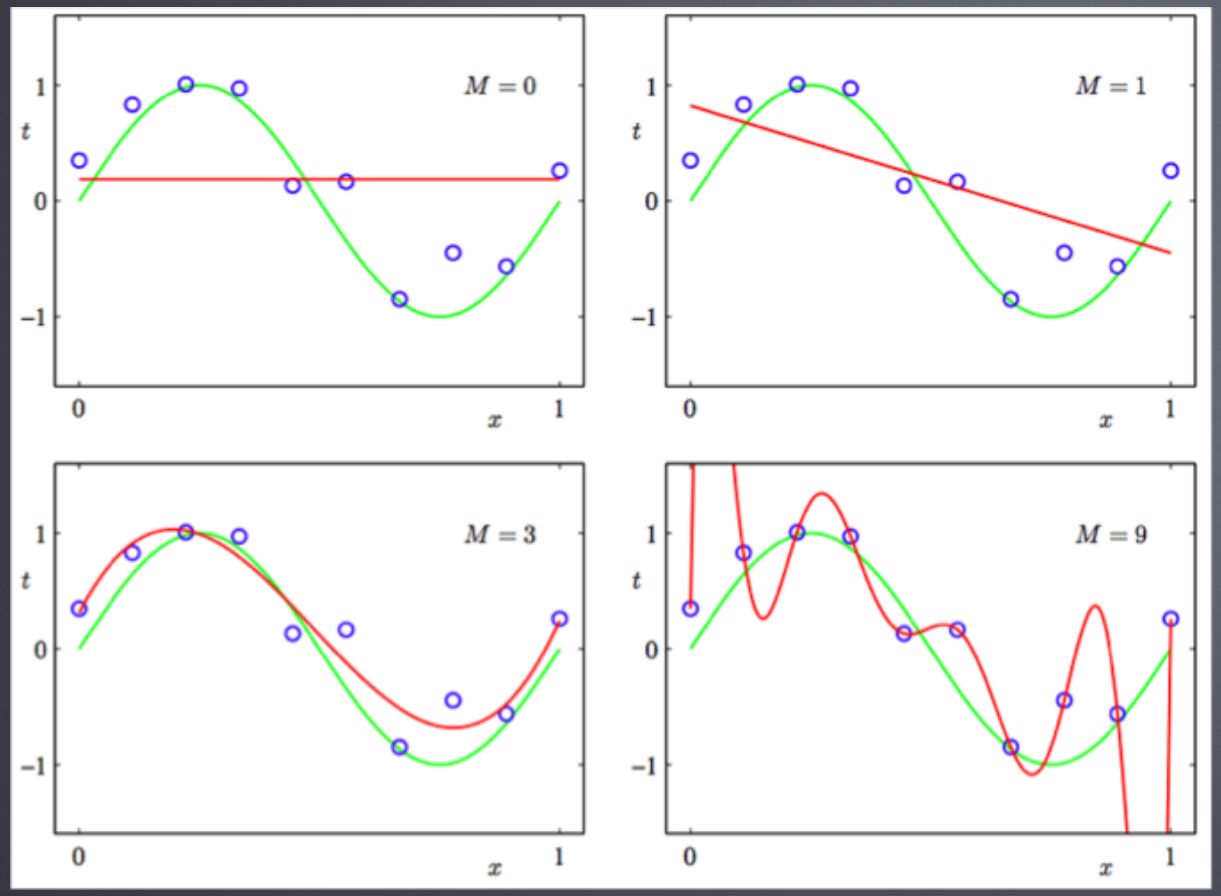
Visualizing the complexity tradeoff



- Finding an optimally predictive model is essentially an exercise in finding the right balance of complexity, i.e. getting the bias-variance tradeoff right
- We search for a model that is elaborate enough to describe the feature-target relationship (not underfit), but not so elaborate that it fits to spurious patterns in the training data (not overfit)

Review: Bias-Variance Tradeoff Example

Complexity tradeoff: polynomial regression



- The higher the degree of a polynomial regression, the more complex the model (lower bias, higher variance)
- Degree 3 is *just right*: the model has sufficient complexity to describe the data without overfitting to noise

Can we tune with more granularity than choosing polynomial degrees?

Yes! By using regularization

What Does Regularization Accomplish?

New cost function

M(w): model error
R(w): complexity cost
Lambda: adjustable weight of complexity cost

$$M(\mathbf{w}) + \lambda R(\mathbf{w})$$

- Regularization adds a term that penalizes model complexity directly into the cost function
- A *regularization strength* parameter lambda controls the tradeoff in priorities: minimizing fit error and minimizing complexity
- Lambda then allows us to continuously adjust the complexity tradeoff: more regularization introduces a simpler model / more bias, while less regularization makes the model more complex and increases variance
- If our model is overfit (variance too high), regularization can often improve generalization error by reducing variance

Reg Cost Function: Ridge Regression

Fit model by minimizing:

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

Warning: scale matters!

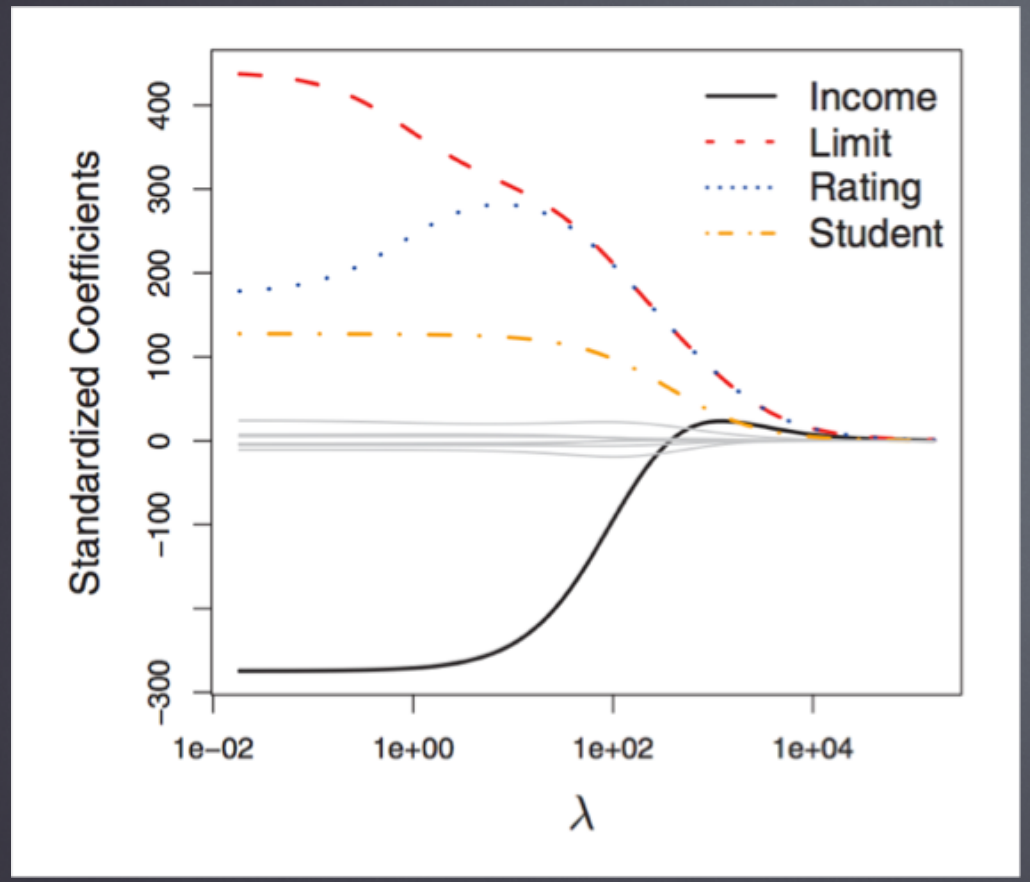
$$x' = \frac{x - \bar{x}}{\sigma}$$

Original — Mean — Standard deviation

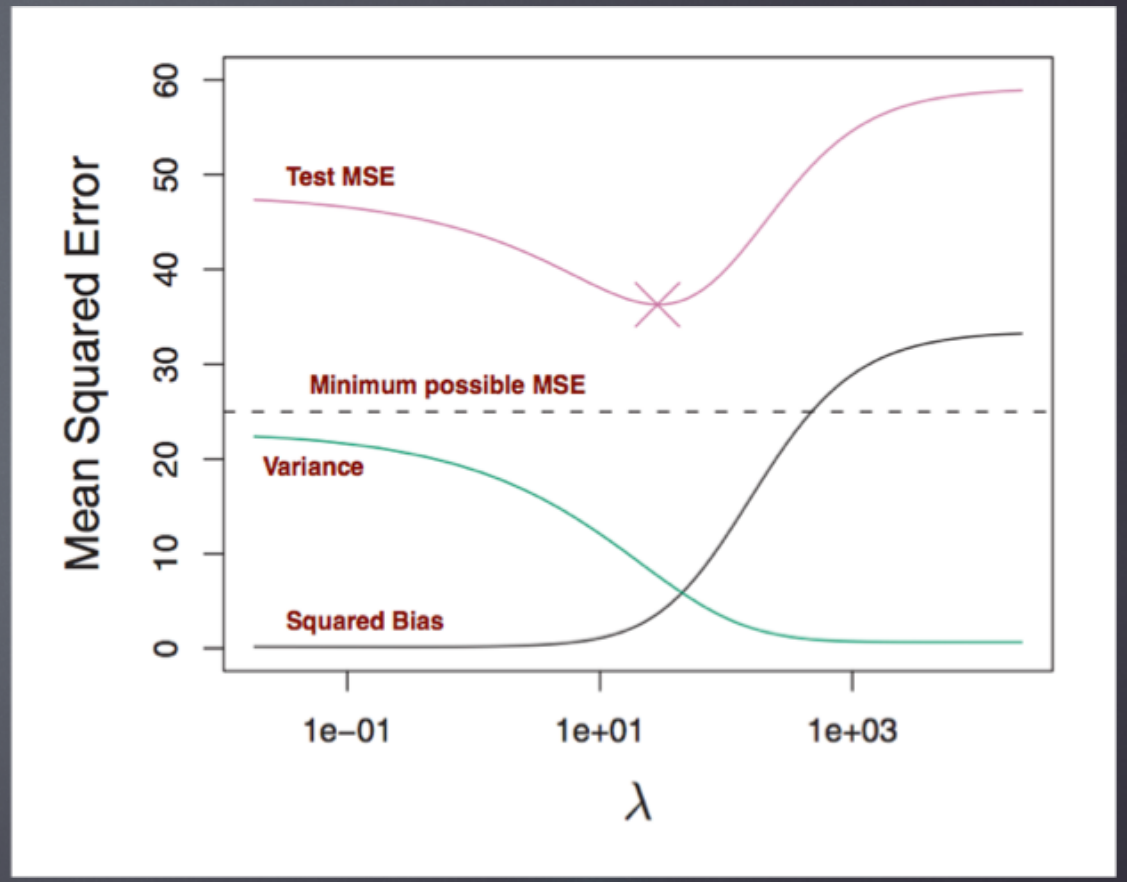
- In *ridge regression*, the complexity penalty is the sum of the squared coefficient values
- The penalty term has the impact of "shrinking" the coefficients toward 0. This constraint imposes bias on the model, but also reduces its variance
- We should always select the correct regularization strength lambda via validation / cross-validation
- It's best practice to *standard scale* the features so that you aren't applying unfair penalties based on the original feature scales

Ridge Regression in Action

Shrinkage effect as regularization strength increases



Complexity tradeoff: variance reduction may outpace increase in bias, leading to a better model fit!



Alternative: LASSO Regression

Fit model by minimizing:

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

Math aside: penalties are closely related to L1/L2 norms, which measure vector length

Lasso - L1

$$\|\beta\|_1 = \sum|\beta_j|$$

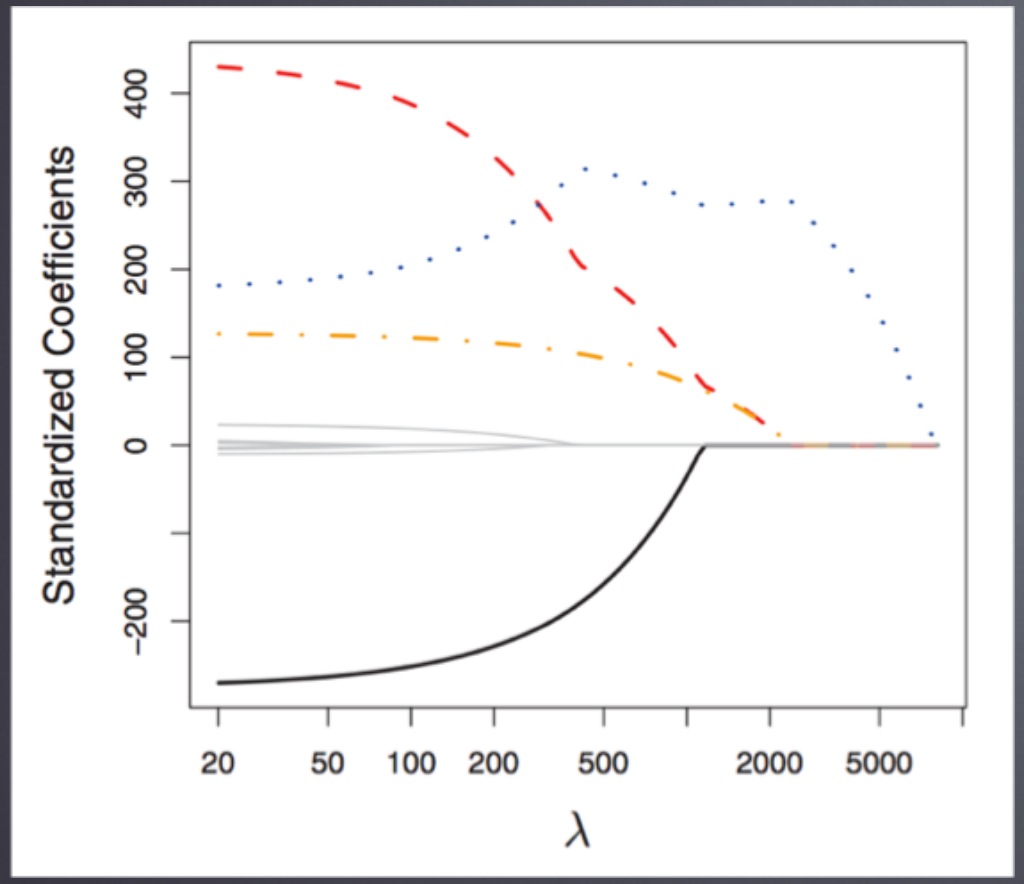
Ridge - L2

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p}\beta_j{}^2}$$

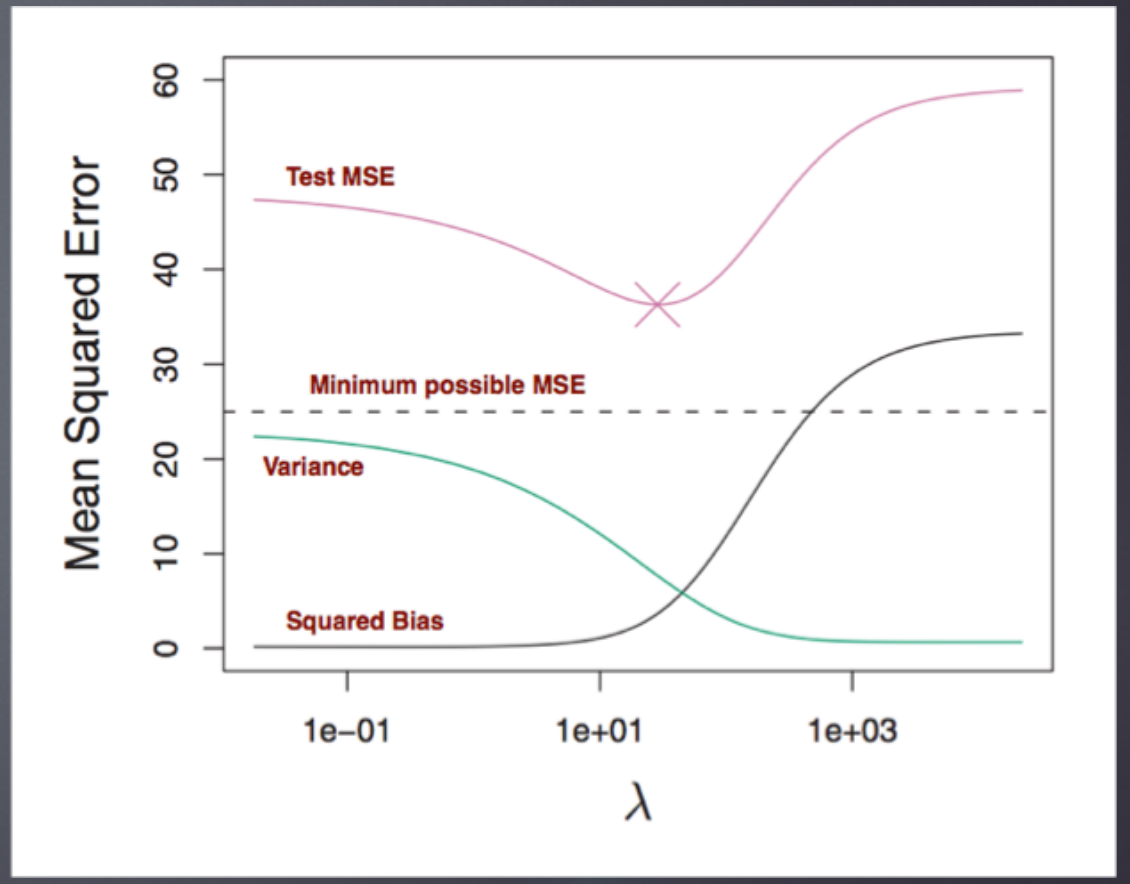
- In *LASSO regression*, the complexity penalty is the sum of the absolute value of the coefficients
- The *SS* stands for “shrinkage” and “selection”, and the *A* stands for “absolute” (Least Absolute Shrinkage and Selection Operator)
- Similar effect to ridge in terms of complexity tradeoff – increasing lambda raises bias but lowers variance
- Unlike ridge, LASSO performs *feature selection*, in that as lambda increases coefficients start to be zeroed out

LASSO Regression in Action

Shrinkage and selection effect as regularization strength increases: some features drop to 0



Complexity tradeoff: variance reduction may outpace increase in bias, leading to a better model fit!



Ridge vs. LASSO?

Hybrid approach: elastic net

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

- Luckily, validation/cross-validation gives us an empirical method for selecting between different models. Everything depends on the data, we should always validate!
- LASSO's feature selection property yields an interpretability bonus, but may underperform if the target truly depends on many of the features
- We can also try a hybrid approach, *elastic net*, which introduces a new parameter *alpha* that balances a tradeoff between L1 and L2 penalties

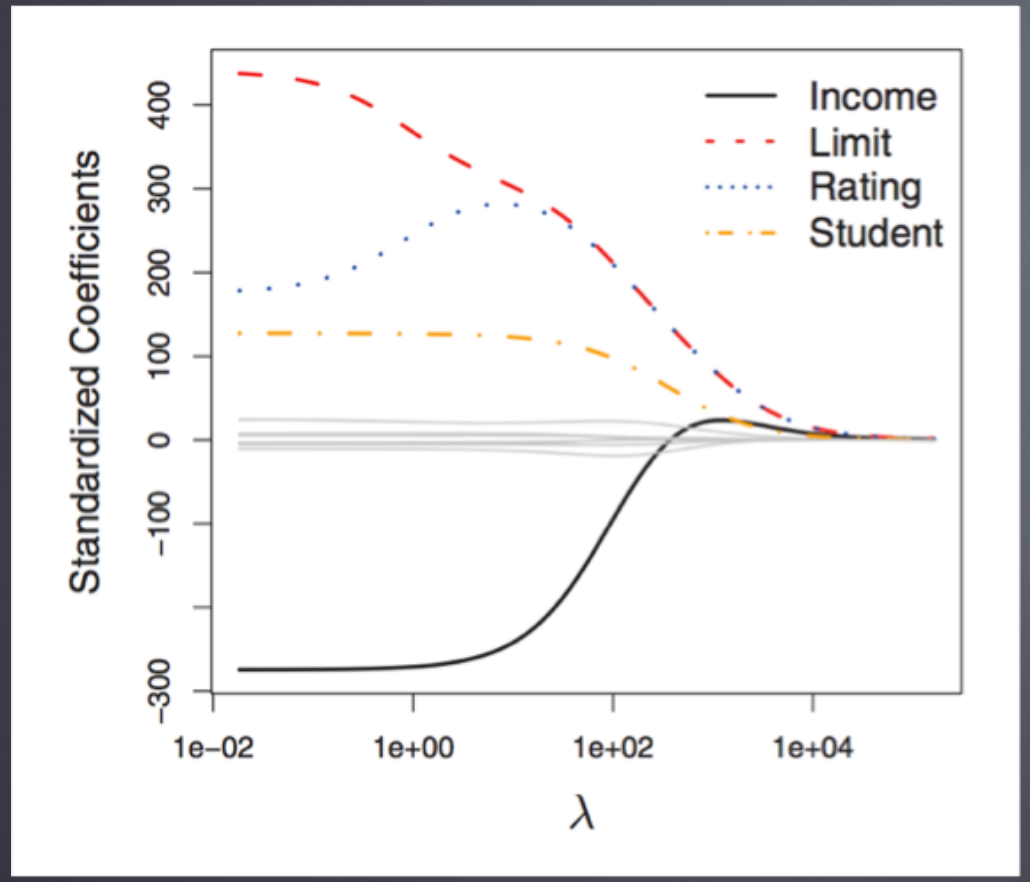
Regularization: Digging Deeper Into the Math (Optional / Appendix)

WHY DOES IT REALLY WORK?

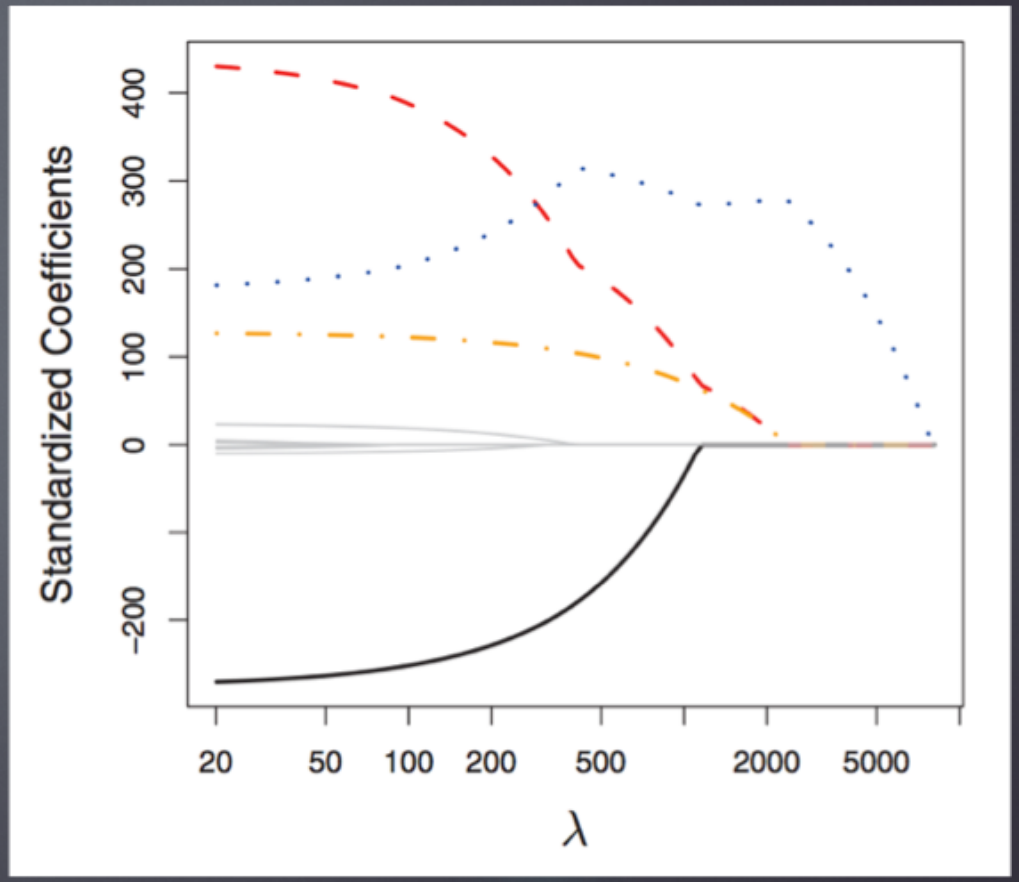
The Analytic View

Increasing L2/L1 penalties force coefficients to be smaller, restricting their plausible range. A smaller range for coefficients must be simpler/lower variance than a model with an infinite possible coefficient range.

Ridge



LASSO



The Geometric View

Below are mathematically equivalent formulations of the optimization objectives of ridge/LASSO

Ridge

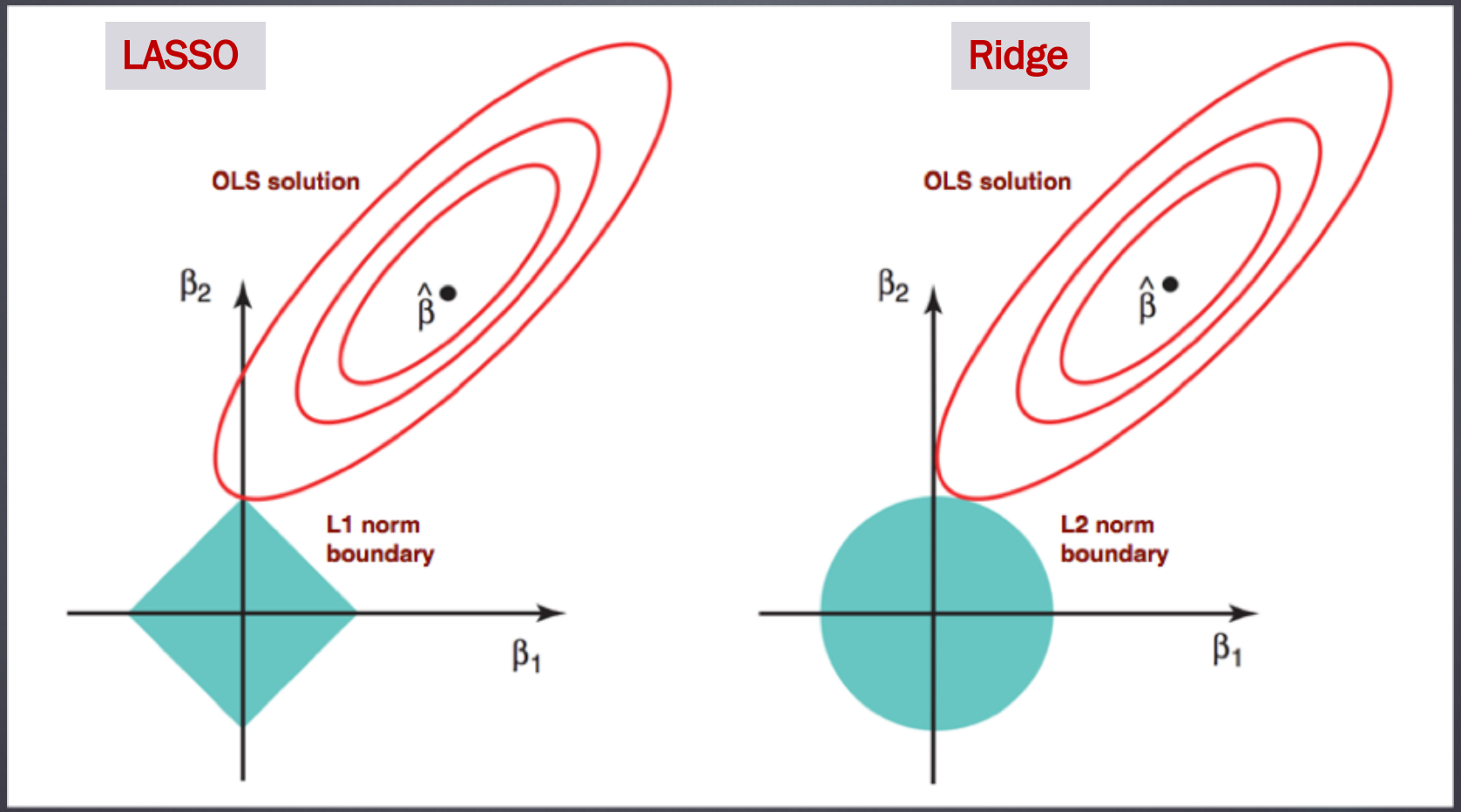
$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

LASSO

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

The Geometric View Cont.

Under this geometric formulation, the cost function minimum is found at the intersection of the penalty boundary and a contour of the traditional OLS cost function surface. The geometry reveals the selection effect of LASSO (intersection at a corner/axis zeroes out coefficients)



The Probabilistic View

Bayes!: Regularization imposes certain priors on the regression coefficients

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta|X) = f(Y|X,\beta)p(\beta)$$

$$p(\beta) = \prod_{j=1}^{p} g(\beta_j)$$

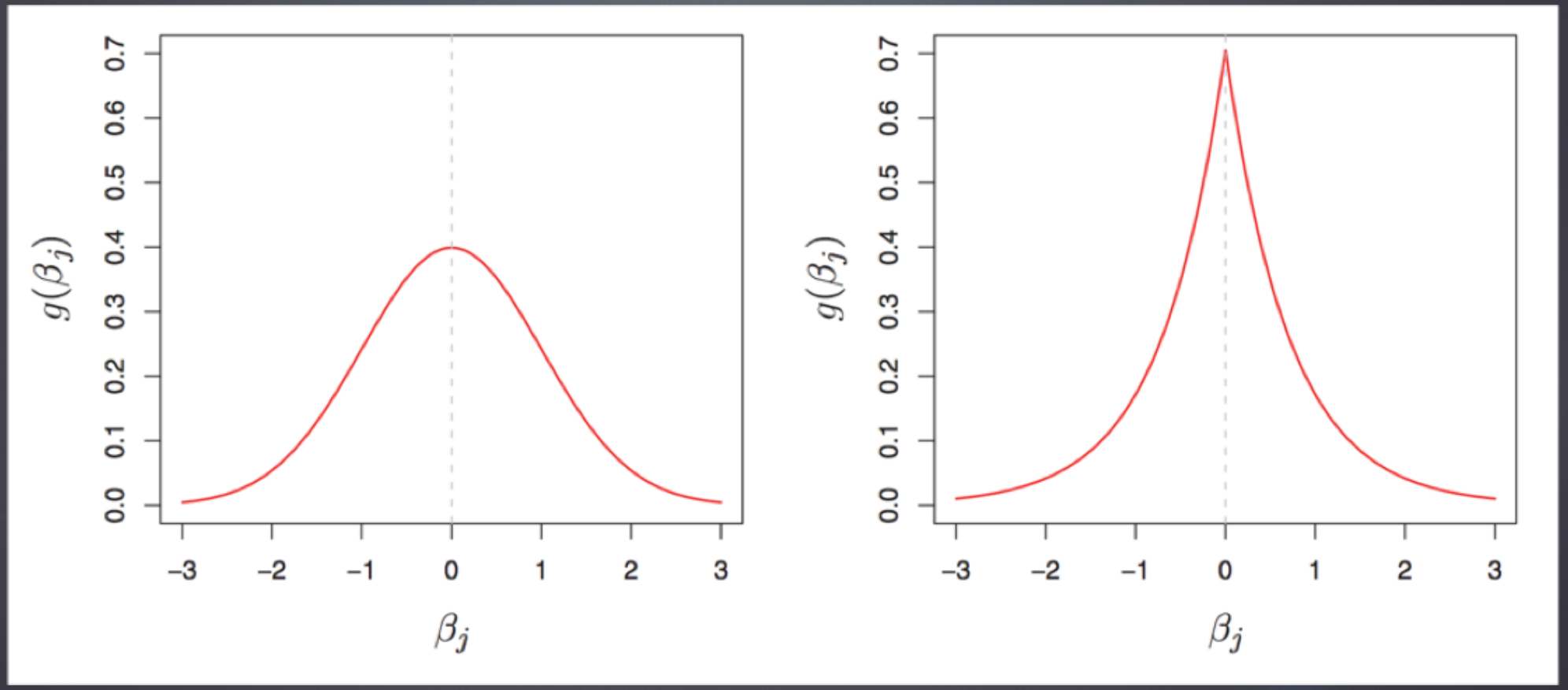
- Letting f be the likelihood (probability of target given parameter vector beta) and p(beta) be the prior distribution of beta, we can calculate the posterior of beta
- p(beta) is derived from independent draws of a prior coefficient density function g that we choose when regularizing
- L2 (ridge) regularization imposes a normal prior on the coefficients, while L1 (lasso) regularization imposes a Laplacian prior on the coefficients

The Probabilistic View Cont.

Visualizing these prior distributions again reveals the difference in behavior between ridge and LASSO: the Laplacian distribution has peaked density at 0, explaining its tendency to zero out some coefficients

Ridge: gaussian prior

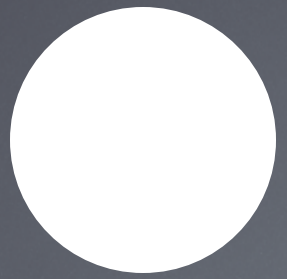
LASSO: Laplacian prior



Lesson Recap

Complexity Tradeoff

- Optimizing predictive models is all about finding the right bias/variance tradeoff
- We need models that are sufficiently complex to capture patterns in data, but not so complex that they overfit to noise



Regularization

- Reduce complexity by penalizing it in cost function
- Increases bias, but reduces variance – may be worth the trade
- Options: L2, L1, Can validate the choice and strength

How it Works

- Analytically: penalty constrains the coefficient range
- Geometrically: L1/L2 imposes bounded regions
- Probabilistically: imposes prior on coefficients

Thank You!

Image Citations

- Slides 3: Scott Fortmann-Roe
- Slide 4: Justin Domke
- Slides 6-11; 13-17: Introduction to Statistical Learning with Applications in R