

# Ridge regression

STAT 471

October 12, 2021

# Where we are

- ✓ **Unit 1:** Intro to modern data mining
- ✓ **Unit 2:** Tuning predictive models

**Unit 3:** Regression-based methods

**Unit 4:** Tree-based methods

**Unit 5:** Deep learning

**Lecture 1:** Logistic regression

**Lecture 2:** Regression in high dimensions

**Lecture 3:** Ridge regression

**[Fall break:** No class]

**Lecture 4:** Lasso regression

**Lecture 5:** Unit review and quiz in class

Homework 1 due the following **Sunday**.

Midterm exam following **Monday (7-9pm)**.

# Idea: Encourage coefficients not to be too large

First, recall linear regression:

$$\hat{\beta}^{\text{least squares}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_{i1} + \cdots + \beta_{p-1} X_{i,p-1}))^2.$$

If there are too many features, the coefficients can get a little wild (high variance!).

To tame those wild coefficients, we add a **penalty** to disincentivize large values:

$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{i,p-1}))^2 + \lambda \sum_{j=1}^{p-1} \beta_j^2$$

for some  $\lambda \geq 0$ .

*penalize coefficients bc they can get wild*

*when will penalty be 0?*

*we want  $\beta$  to be closest to 0*

*penalty*

Ridge regression is defined even if  $p > n$ , as long as  $\lambda > 0$ .

# The effect of the penalty parameter $\lambda$

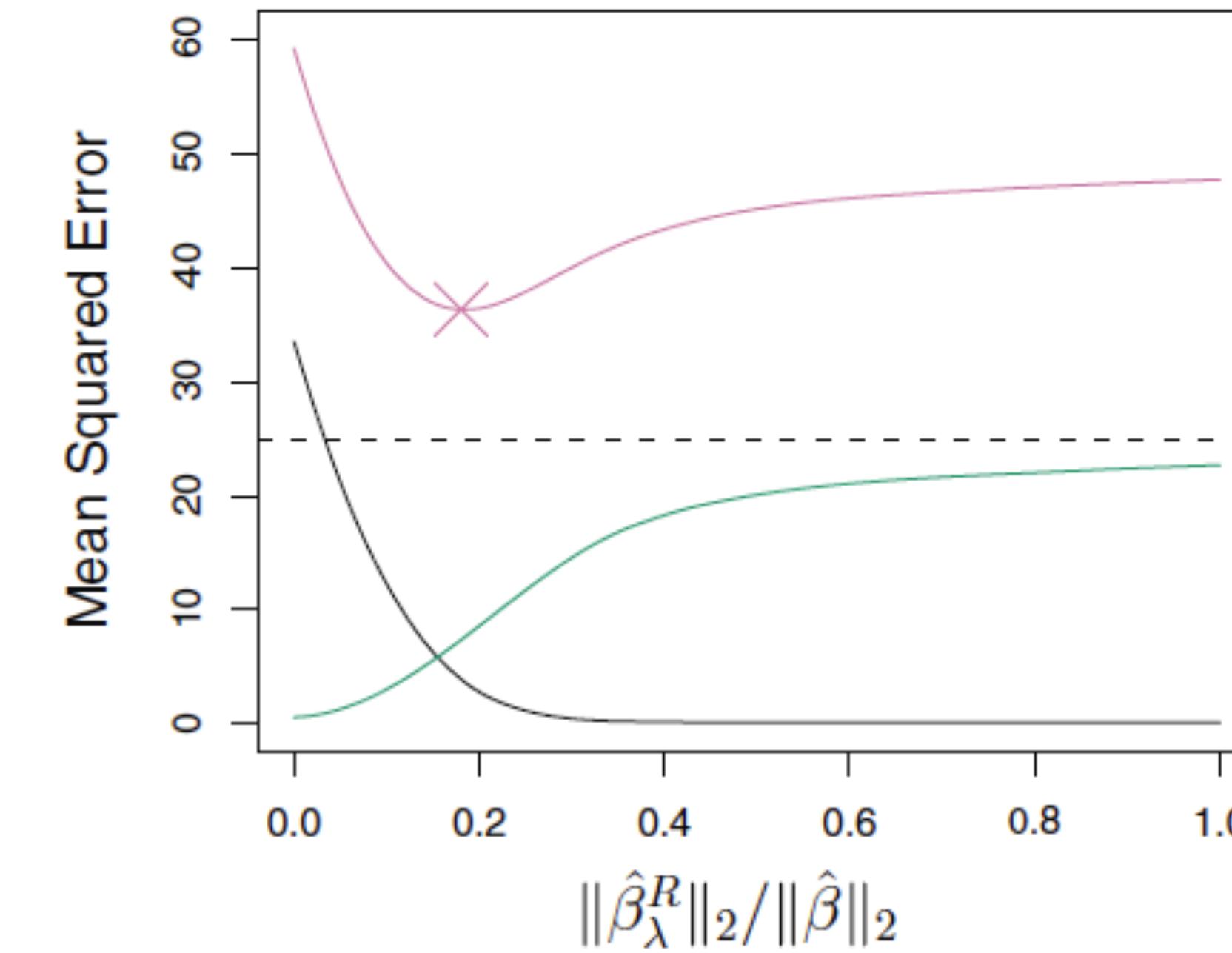
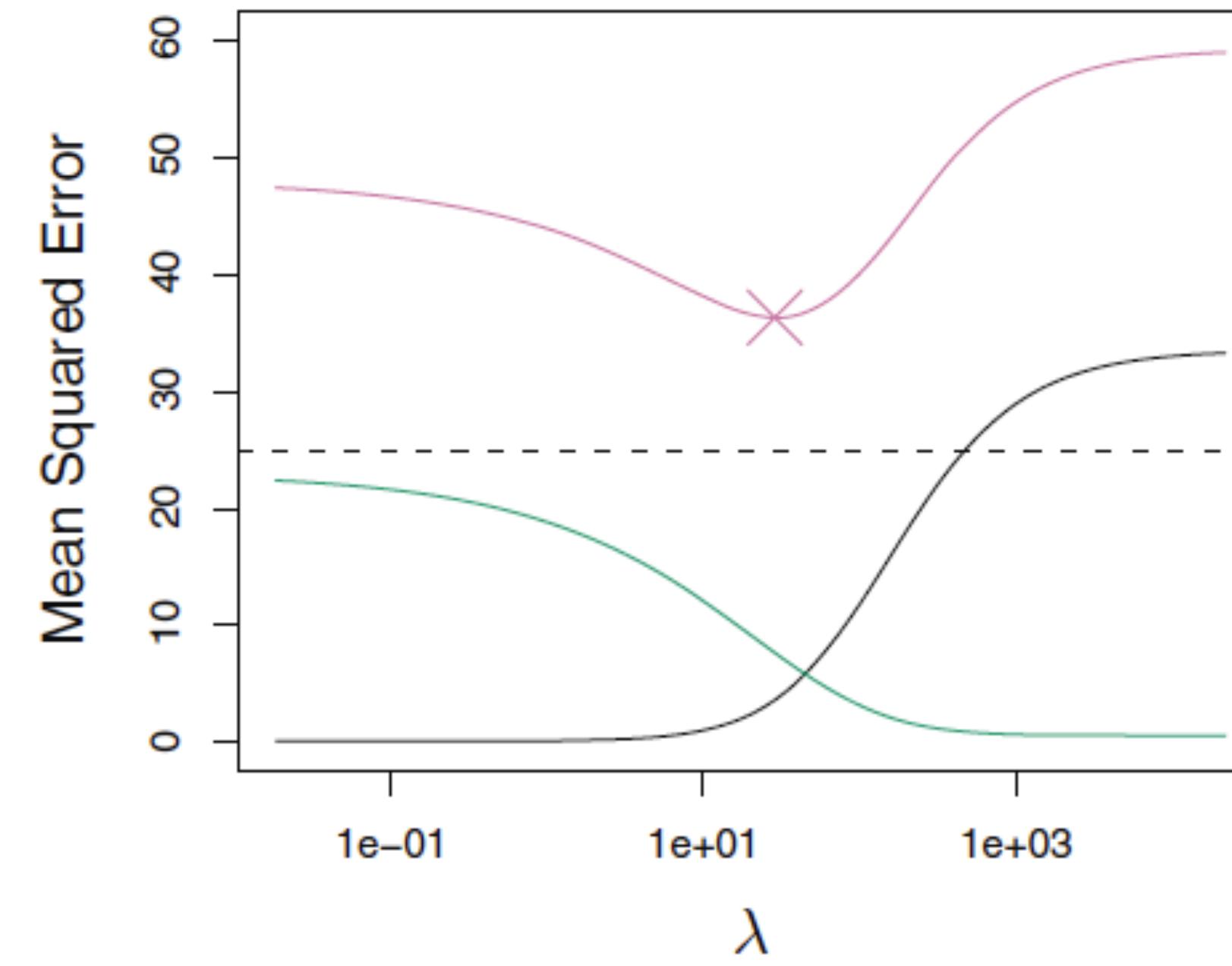
$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip-1}))^2 + \lambda \sum_{j=1}^{p-1} \beta_j^2$$

- The larger  $\lambda$  is, the more of a penalty there is.  
*if  $\lambda$  goes to  $\infty$ , force all of  $\beta$  to be  $\infty$  small*
- For  $\lambda = 0$ , we get back ordinary least squares.
- For  $\lambda = \infty$ , we get  $\beta_1 = \cdots = \beta_{p-1} = 0$ , leaving only the intercept (which is not penalized).  
*how many df when  $\lambda = \infty$ , df=1*

We should think of  $\lambda$  as controlling the flexibility of the ridge regression fit, like the degrees of freedom in a spline fit. However, larger  $\lambda$  means fewer degrees of freedom.

# The bias-variance tradeoff for ridge regression

@ left, have  
no penalization  
& high variance  
hope as  $\uparrow \lambda$ ,  $\downarrow$   
variance and hope  
bias does not  
 $\uparrow$  too much

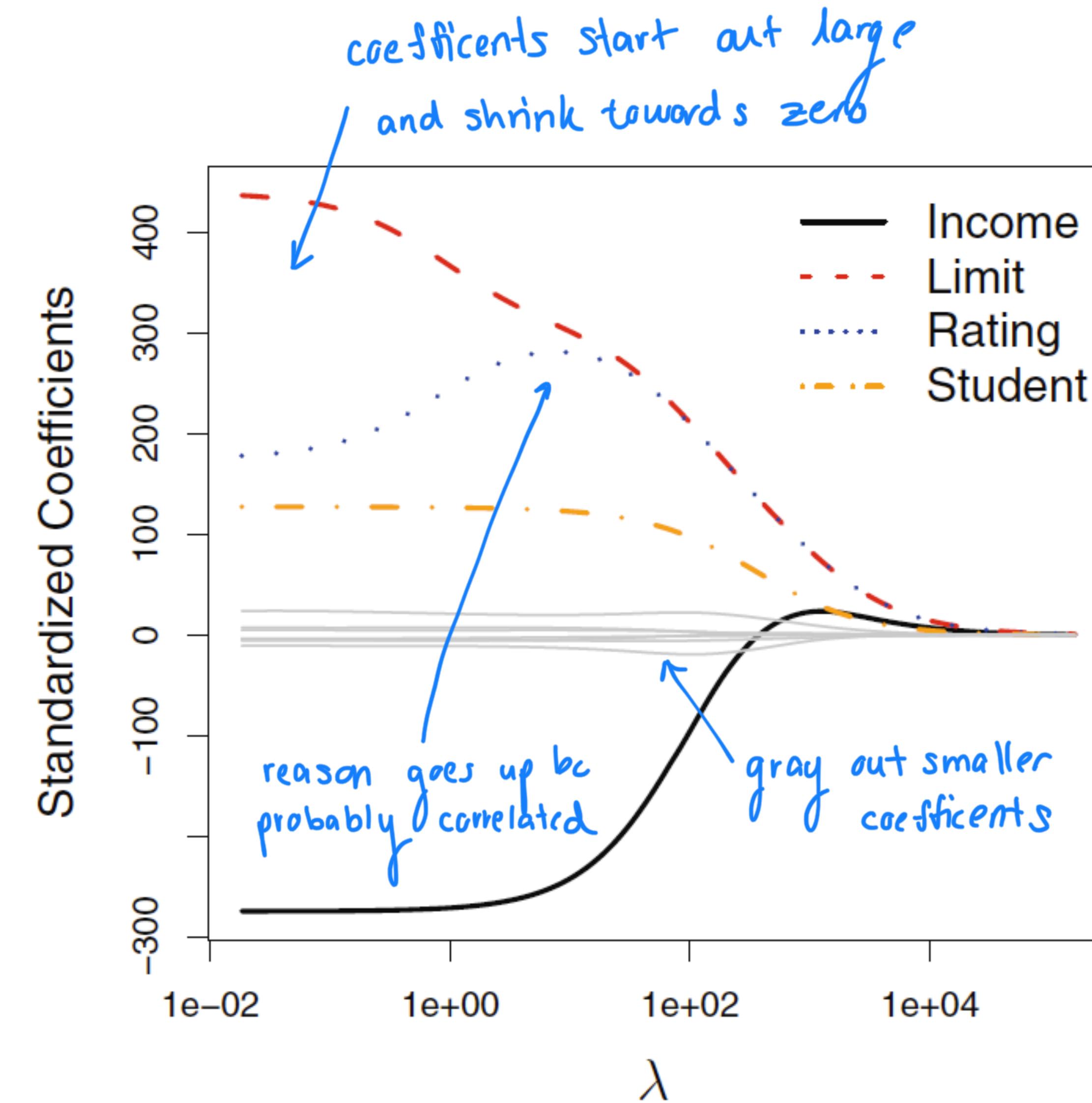


**FIGURE 6.5.** Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of  $\lambda$  and  $\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$ . The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

In practice,  $\lambda$  is chosen by cross-validation.

# A pictorial representation of ridge regression

need to fit a model  
for each  $\lambda$



# Ridge regression in a simple case

Suppose that  $n = p - 1$  and  $X_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$ , i.e.  $Y_j = \beta_j + \epsilon_j$ .

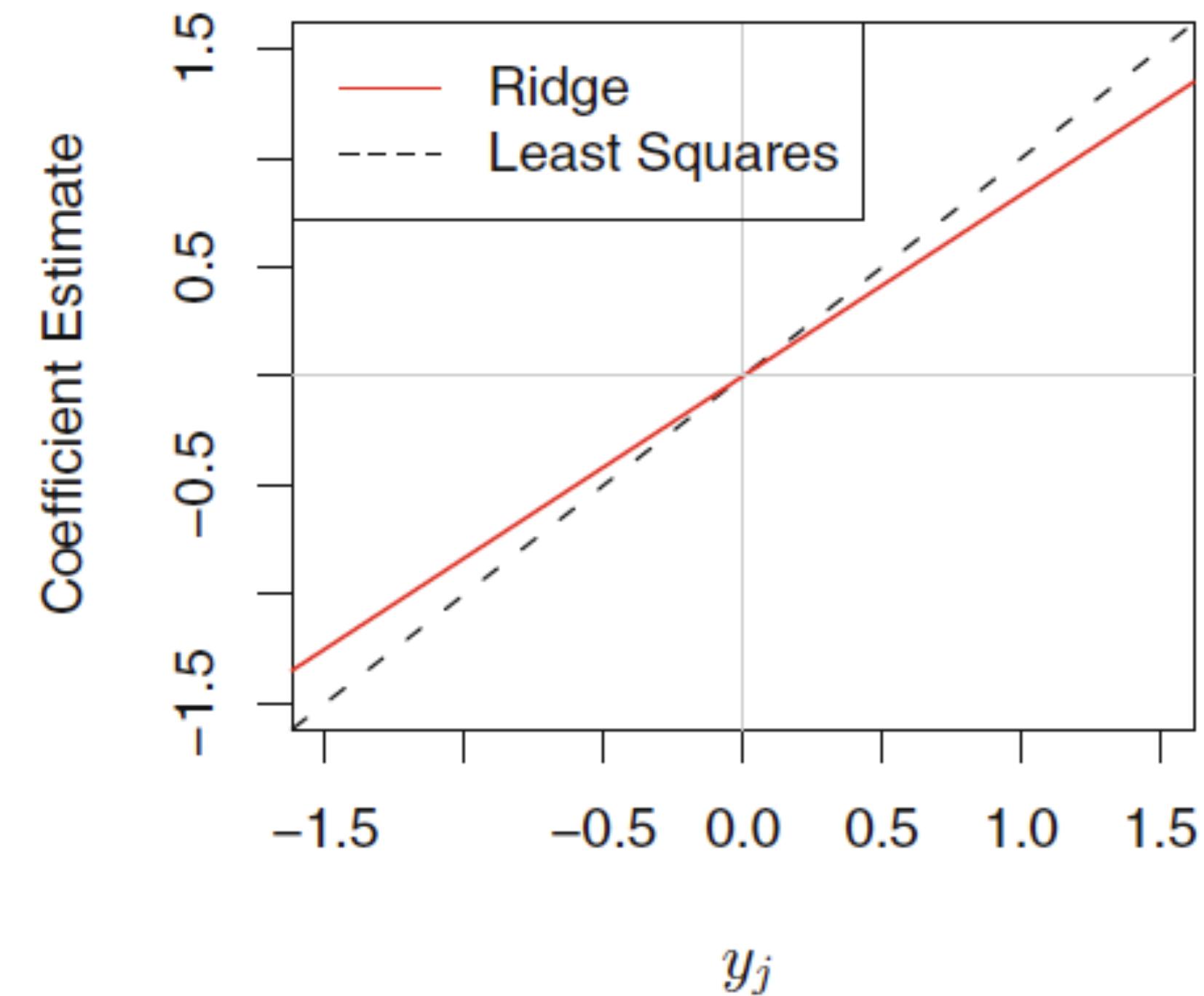
took least squares regression & just shrunk a little

Consider fitting ridge regression without intercept:

$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \sum_{j=1}^{p-1} (Y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p-1} \beta_j^2.$$

In this simple case,  $\hat{\beta}_j^{\text{OLS}} = Y_j$  and  $\hat{\beta}_j^{\text{ridge}} = Y_j/(1 + \lambda)$  (OLS stands for ordinary least squares).

So  $\hat{\beta}^{\text{ridge}} = \frac{1}{1 + \lambda} \hat{\beta}^{\text{OLS}}$ , i.e. the ridge estimate is obtained by *shrinking* the OLS estimate by a factor of  $1 + \lambda$ .  
*only applies to this specific setup*



# The importance of feature scaling

\* all features are assumed to be on same scale \*

want to first standardize

Suppose  $X_1$  is height. Does it matter if it's measured in inches or feet?

For least squares, does not matter. If  $X_1 \rightarrow 12X_1$ , then  $\hat{\beta}_1 \rightarrow \frac{1}{12}\hat{\beta}_1$ .

$$\hat{\beta}^{\text{least squares}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_{i1} + \dots + \beta_{p-1} X_{i,p-1}))^2$$

For ridge regression, it does matter! Implicitly, all features assumed on the same scale.

$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_{i1} + \dots + \beta_{p-1} X_{i,p-1}))^2 + \lambda \sum_{j=1}^{p-1} \beta_j^2$$

# Feature standardization

To put features on the same scale, center each feature and divide by its std. dev.:

$$x_{ij}^{\text{std}} = \frac{x_{ij} - \bar{x}_j}{\hat{\sigma}_j}; \quad \hat{\sigma}_j^2 = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2.$$

Feature standardization is recommended before applying ridge regression.

measuring everything in sd of \_\_\_\_\_

# Treatment of correlated features

Linear regression coefficients for correlated features tend to be unstable.

Ridge regression is more stable, “splitting the credit” among correlated features.

For example, consider the linear regression

$$y = \beta_1 X_1 + \beta_2 X_1 + \epsilon,$$

when correlated,  
accidentally added same  
coefficient twice

where we've accidentally added the same feature twice.

- Linear regression is undefined because  $(\beta_1, \beta_2)$  and  $(\beta_1 - c, \beta_2 + c)$  give the same RSS for each  $c$ .
- Ridge regression will obtain  $\hat{\beta}$  from  $y = \beta X_1 + \epsilon$ , and set  $\hat{\beta}_1 = \hat{\beta}_2 = \frac{1}{2} \hat{\beta}$ .  
*ridge regression smart enough to realize duplicates*

# Logistic regression with ridge penalty

Logistic regression can be penalized, just like linear regression!

Recall  $\mathcal{L}(\beta)$ , the logistic regression likelihood. We can view  $-\log \mathcal{L}(\beta)$  as analogous to the linear regression RSS. Continuing the analogy, we can define

$$\hat{\beta}^{\text{ridge}} = \arg \min_{\beta} \left\{ -\log \mathcal{L}(\beta) + \lambda \sum_{j=1}^{p-1} \beta_j^2 \right\}.$$

*Find Beta  
that gives smallest  
# of this regression*

*why minimize the negative  
want to maximize  
likelihood, so want to  
minimize negative likelihood*

**Subtle point:** While  $\hat{\beta}^{\text{ridge}}$  is trained based on a (penalized) log-likelihood, during cross-validation we should choose  $\lambda$  based on whatever measure of test error we care about (e.g. misclassification error).

*choose lambda based on classification pt. you care about in the end*

# Can we get p-values from a ridge regression?

Unfortunately, we can't. Ridge regression is exclusively a prediction method.

no, bc statistical significance of p-value is muddled by  
ridge regression

does R standardize variables  $\rightarrow$  yes, but easier to do yourself  
`model.matrix()` usual matrix, but need to remove intercept  
 $\alpha = 0 \rightarrow$  ridge;  $\lambda$  are in the background