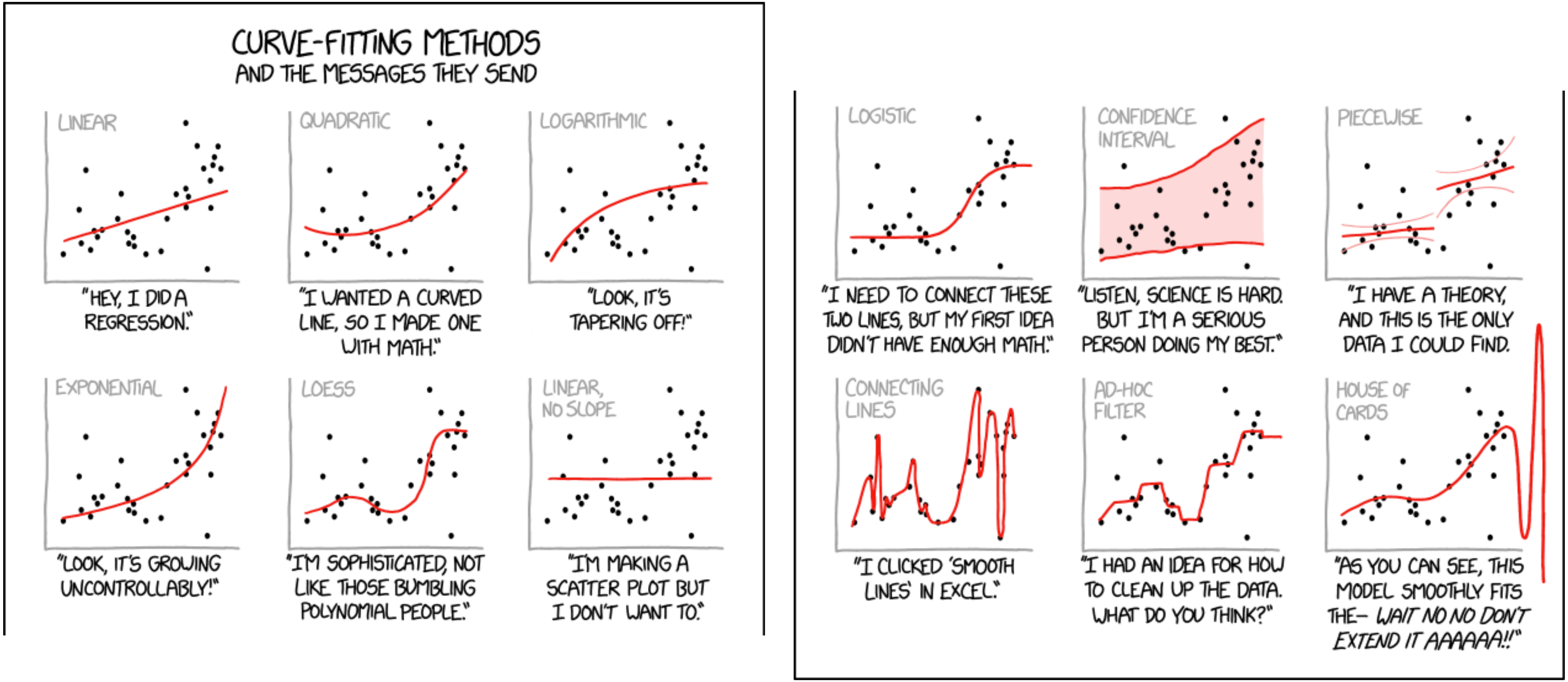


https://www.xkcd.com/2048/

CME 250:
Introduction to Machine Learning

Lecture 3:
Feature Selection,
Regularization & Sparsity

Sherrie Wang
sherwang@stanford.edu

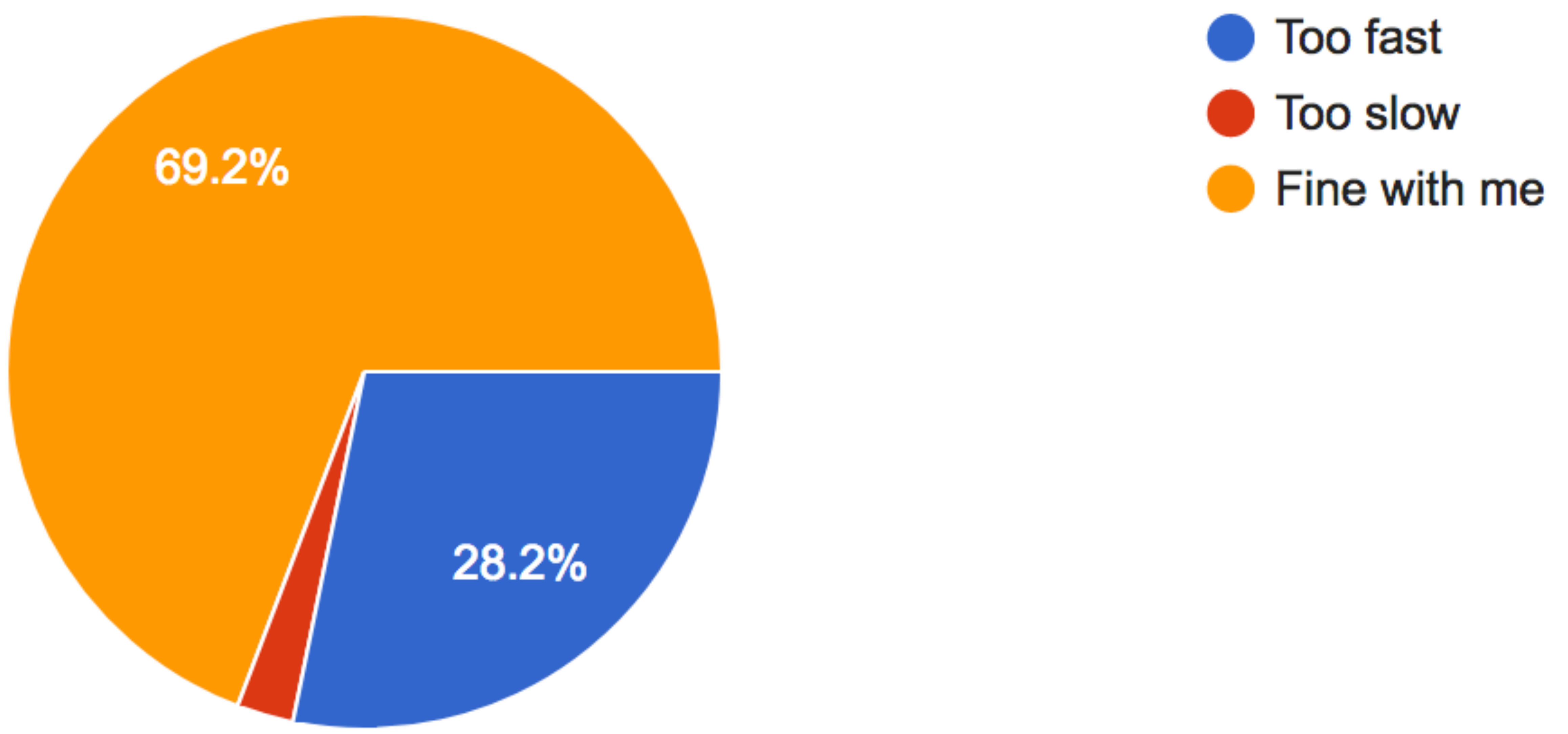
Announcements

Homework 1 is due this Friday, January 25, at 5:00pm.

Homework 2 will be released today, and due next Friday at 5.

Out of town: No lecture next week. And no office hours — sorry!
If you would like to meet about Homework 2 before Monday, email me.

Lecture Pace



Agenda

Slides are online at
cme250.stanford.edu

- Curse of Dimensionality
- Subset Selection Methods
- Shrinkage Methods
  - Ridge Regression
  - The Lasso

Roadmap

We've covered:

- Linear regression
- Logistic regression
- K-nearest neighbors
- Dataset splitting
- Bias-variance tradeoff

Coming up:

- Non-linear supervised algorithms
- Cross validation
- Missing data
- Unsupervised learning

But first…

In transitioning from linear to non-linear methods, we will consider alternative ways to fitting linear regression besides least squares

Why?

- Better prediction accuracy
- More interpretability

Curse of Dimensionality

Curse of Dimensionality

Problems arise when analyzing data in **high-dimensional spaces**

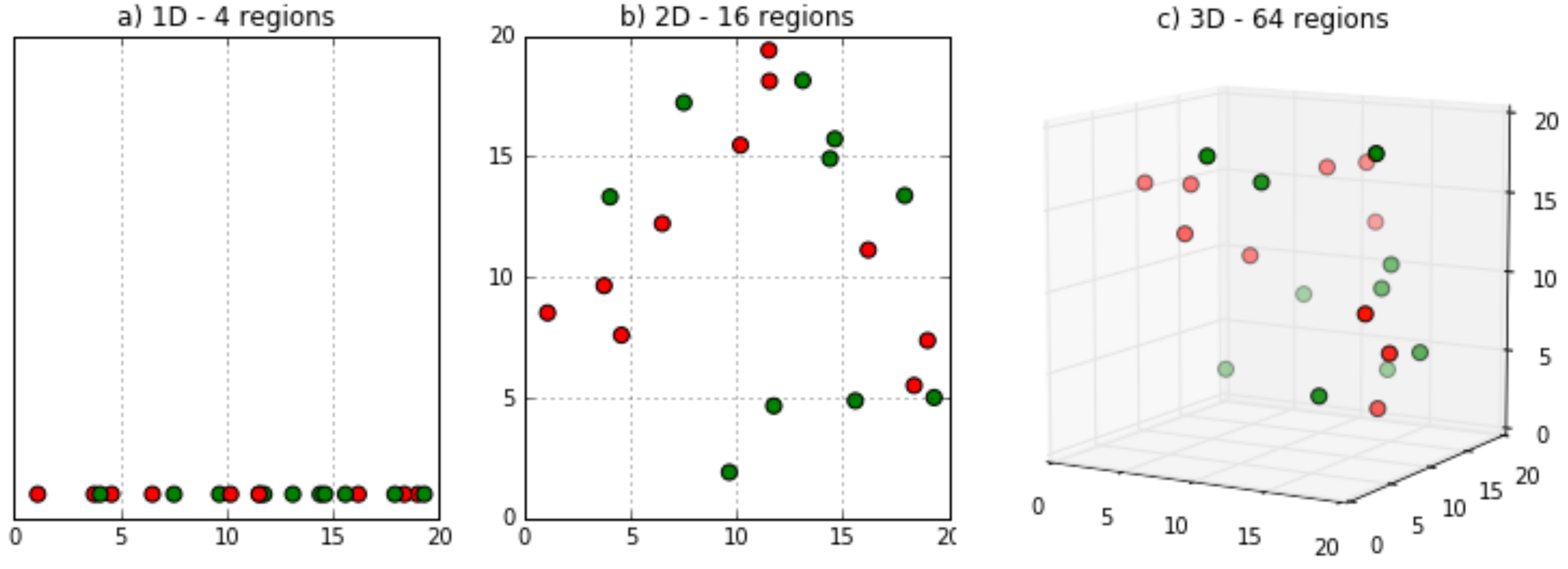
What is a "high" dimension?

In a machine learning context, any time $p > n$, we are in a high-dimensional setting. Problems with high dimensionality can be experienced even when $n > p$, if p is large. The amount of data you need depends on the complexity of your task and algorithm.

$p = \text{dimension of input}, n = \text{number of samples}$

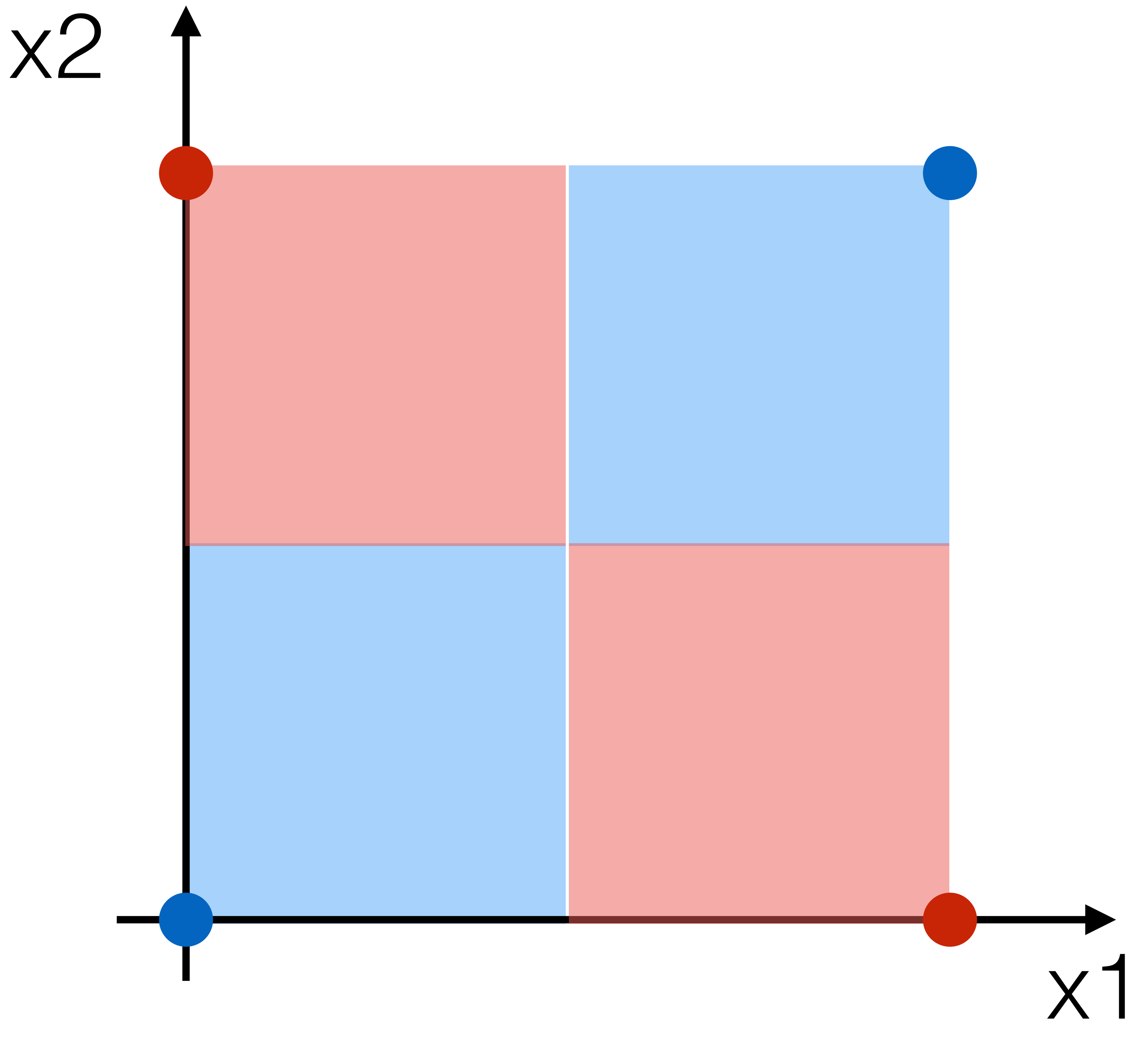
Curse of Dimensionality

The underlying issue is that as dimensionality increases, the volume of the space grows so fast that the available data becomes sparse. It becomes difficult to tell how regions of feature space correspond to the response



Curse of Dimensionality

Sample	x1	x2	y
1	0	0	0
2	1	1	0
3	0	1	1
4	1	0	1



Curse of Dimensionality

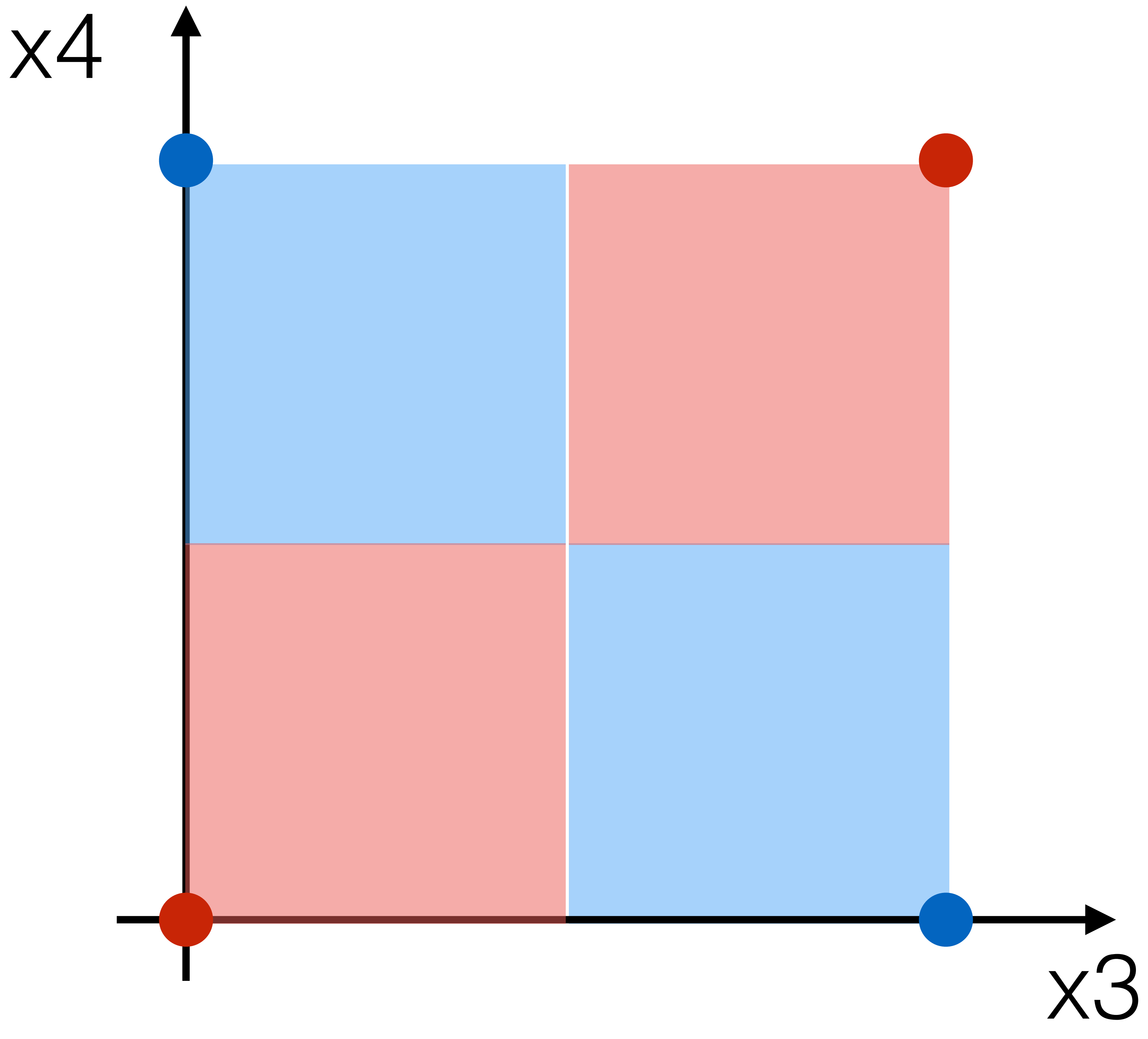
Sample	x1	x2	x3	x4	y
1	0	0	1	0	0
2	1	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	1

Signal dimensions   Noise dimensions

Curse of Dimensionality

Your model might learn…

Sample	x3	x4	y
1	1	0	0
2	0	1	0
3	1	1	1
4	0	0	1



…decision boundaries that are noise

Hughes Phenomenon

With a fixed number of training samples, the predictive power of a classifier or regressor first increases as number of features used increases, but then decreases

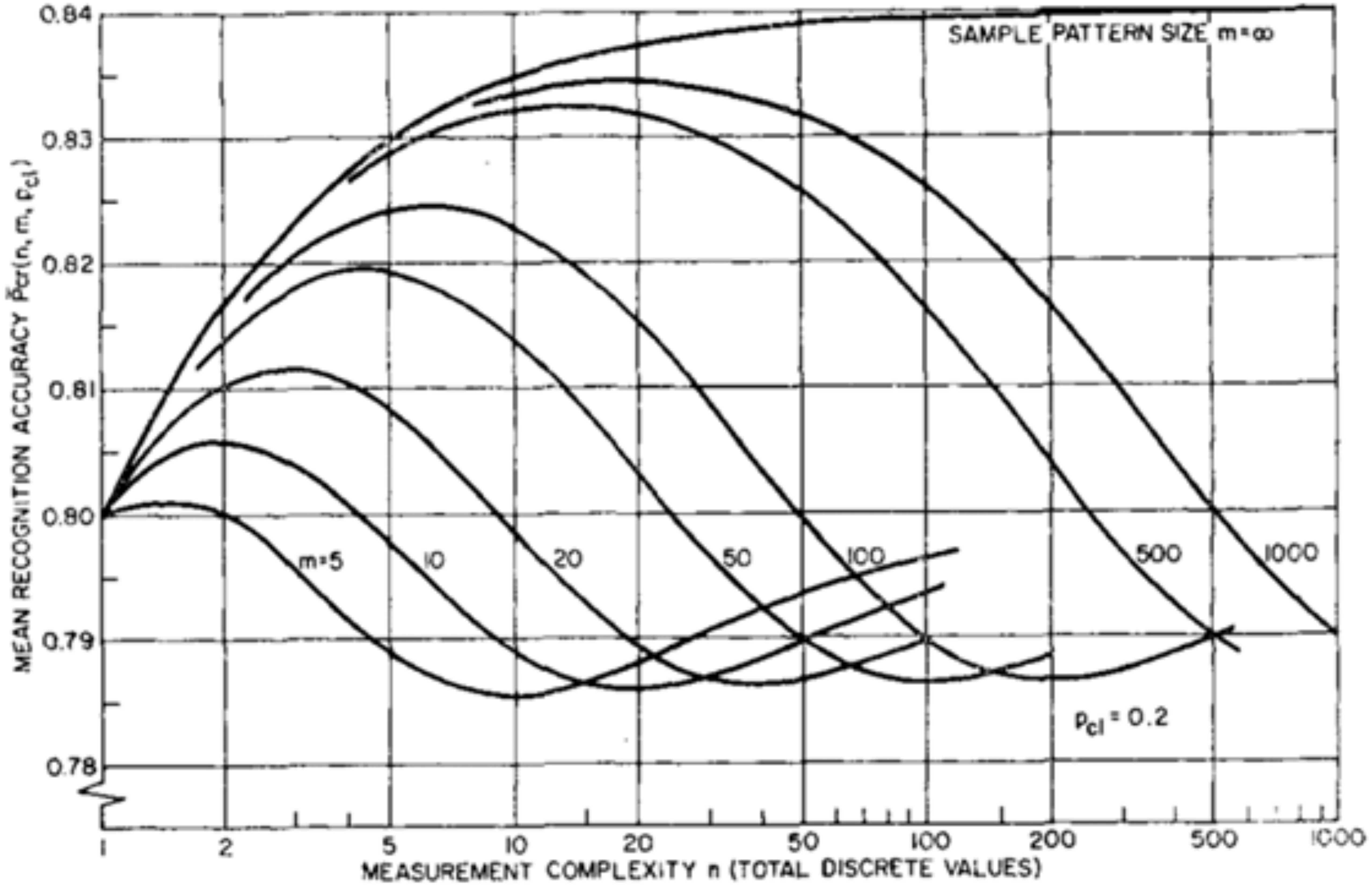
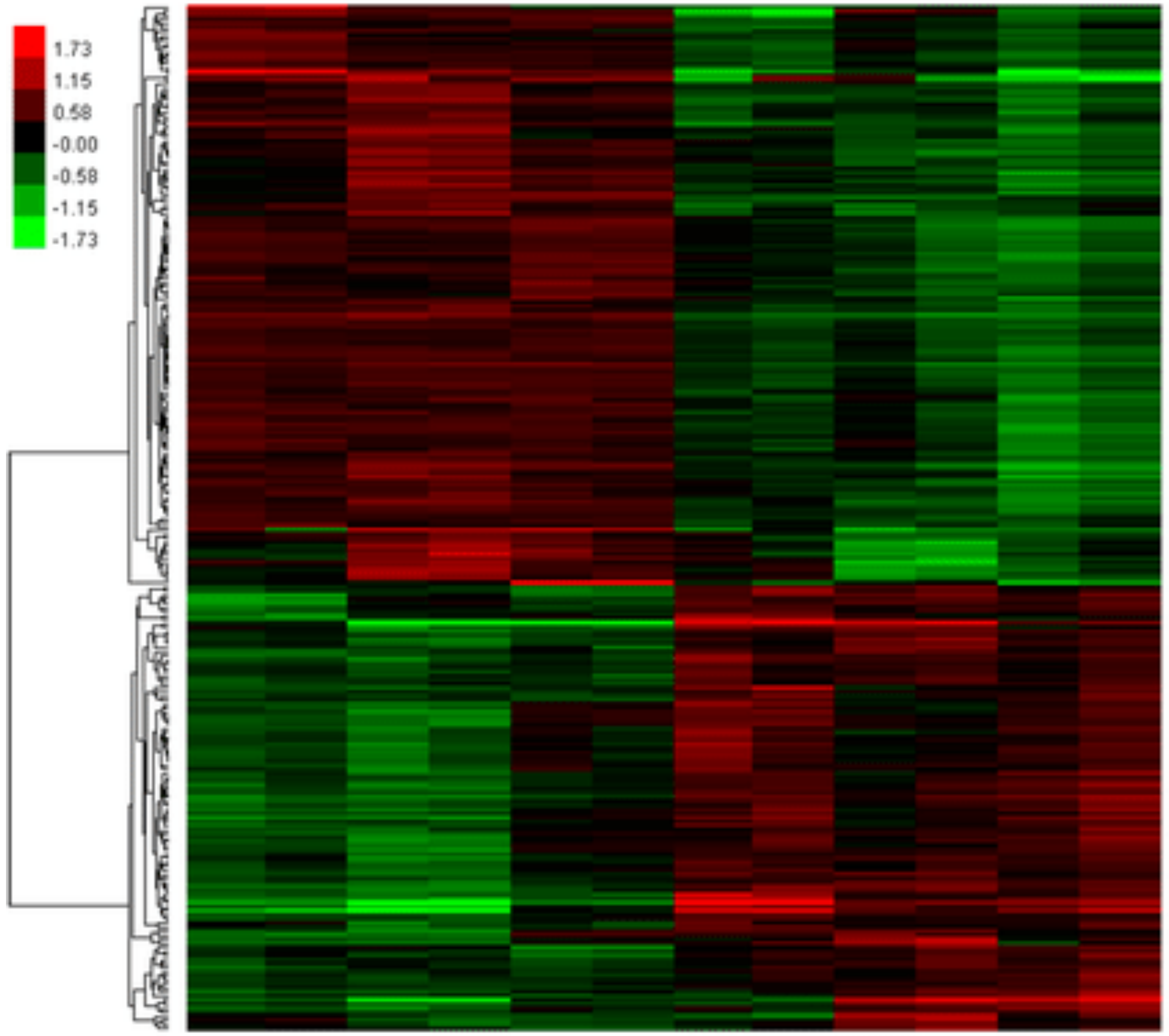


Fig. 4. Finite data set accuracy ($p_{c1} = \frac{1}{5}$).

Gene Expression Data



Thousands of genes, often not that many samples

Image Data

Every pixel is
a feature

Back to Linear Regression

If $n >> p$, then the least squares estimates tend to have low variance and will generalize well to unseen data.

If n not much bigger than p, least squares (even with a linear model!) will have high variance and can overfit.

If $p > n$, there is no longer a unique least squares estimate.

Back to Linear Regression

In some cases, some or many input variables are not even associated with the response. They are totally irrelevant to our task.

We want to remove these variables.

Three Types of Solutions

Subset Selection

Identify a subset of the p features that we think are related to the response

Shrinkage

Fit a model on all p features, but shrink their coefficients toward zero to reduce model variance

Dimension Reduction

Project p features into a lower dimensional space, then build a model in this lower space

Three Types of Solutions

Subset Selection

Identify a subset of the p features that we think are related to the response

Shrinkage

Fit a model on all p features, but shrink their coefficients toward zero to reduce model variance

Dimension Reduction

Project p features into a lower dimensional space, then build a model in this lower space

Future lecture (unsupervised)

Subset Selection

Best Subset Selection

TL;DR: Fit a separate least squares regression for each possible combination of the p predictors.

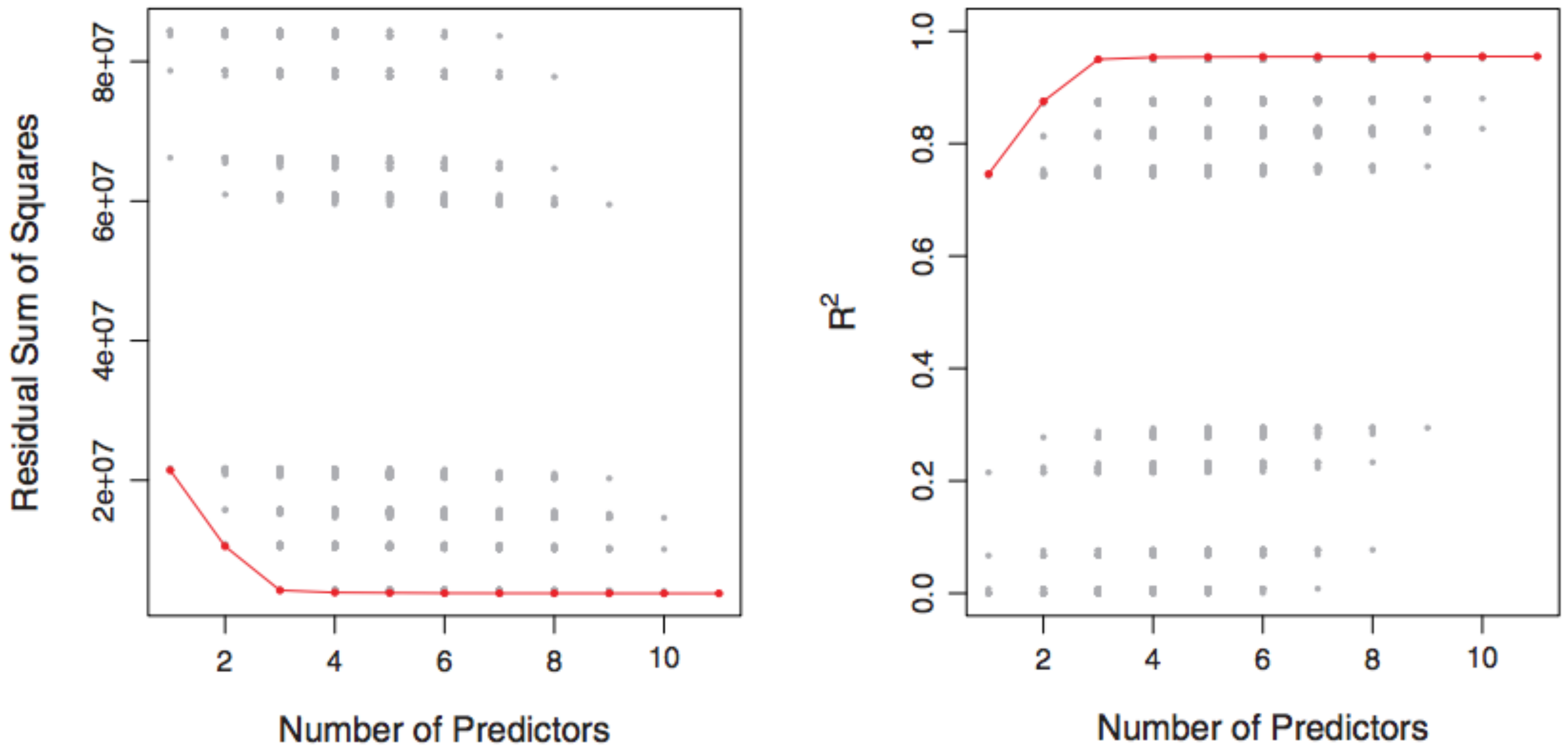
Best Subset Selection

TL;DR: Fit a separate least squares regression for each possible combination of the p predictors.

Algorithm:

1. Start with a model M_0 that has no predictors.
2. For $k = 1, 2, \ldots, p$:
   (a) Fit all models that contain exactly k predictors.
   (b) Pick the best model M_k (based on smallest R^2 or other metric).
3. Pick the best model from $M_0, \ldots, M_p$ using validation set R^2 or other metric.

Best Subset Selection



This will go down eventually as dimension gets too high

FIGURE 6.1, ISL (8th printing 2017)

Best Subset Selection

Pro: Exhaustive; finds best possible subset of features

Con: As p grows, the number of models to fit grows exponentially

2^p models involve subsets of p predictors

E.g. if $p = 10$, you have to consider approximately 1,000 models

E.g. if $p = 20$, you have to consider over 1,000,000 models!

Stepwise Selection

TL;DR: Greedily add or remove variables from the p predictors that give the greatest additional improvement to model fit.

Forward Stepwise Selection

TL;DR: Greedily add ~~or remove~~ variables from the p predictors that give the greatest additional improvement to model fit.

Algorithm:

1. Start with a model M_0 that has no predictors.
2. For $k = 0, 1, \ldots, p - 1$:
    (a) Consider all $p - k$ models that add one predictor to M_k.
    (b) Pick the best model M_{k+1} (based on smallest R^2 or other metric).
3. Pick the best model from M_0, …, M_p using validation set R^2 or other metric.

Backward Stepwise Selection

TL;DR: Greedily ~~add or~~ remove variables from the p predictors that give the greatest additional improvement to model fit.

Algorithm:

1. Start with a model M_p that has all predictors.
2. For $k = p, p - 1, \ldots, 1$:
   (a) Consider all k models that remove one predictor from M_k.
   (b) Pick the best model M_{k-1} (based on smallest R^2 or other metric).
3. Pick the best model from M_0, …, M_p using validation set R^2 or other metric.

Stepwise Selection

Pro: Computationally feasible

Con: Not guaranteed to get best subset of predictors for final model

Shrinkage Methods

Shrinkage Methods

TL;DR: Fit a model containing all p predictors using a technique that *constrains* or *regularizes* the coefficient estimates.

Equivalently, the method shrinks coefficient estimates toward zero.

Shrinkage Methods

TL;DR: Fit a model containing all p predictors using a technique that *constrains* or *regularizes* the coefficient estimates.

Equivalently, the method shrinks coefficient estimates toward zero.

Why does this work? Shrinking coefficients reduces the model variance.

Multiple Linear Regression

Recall that in linear regression we are trying to minimize the squared error:

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2$$

Ridge Regression

Find β values that minimize a "penalized" error:

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

In the case of one feature:

$$\hat{\beta} = \arg\min_{\beta}(y - (\beta_0 + \beta_1 x))^2 + \lambda(\beta_0^2 + \beta_1^2)$$

Ridge Regression

Find β values that minimize a "penalized" error:

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \boxed{\lambda||\vec{\beta}||_2^2}$$

Penalizes large β values by penalizing sum of squares

Ridge Regression

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

λ is a hyperparameter that needs to be tuned.

When $\lambda = 0$, the penalty term has no effect, and ridge regression produces least squares estimates.

As $\lambda \to \infty$, the impact of the shrinkage penalty grows. Ridge regression estimates $\to 0$.

Finding a good value of λ is crucial to good ridge regression performance.

Ridge Regression

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

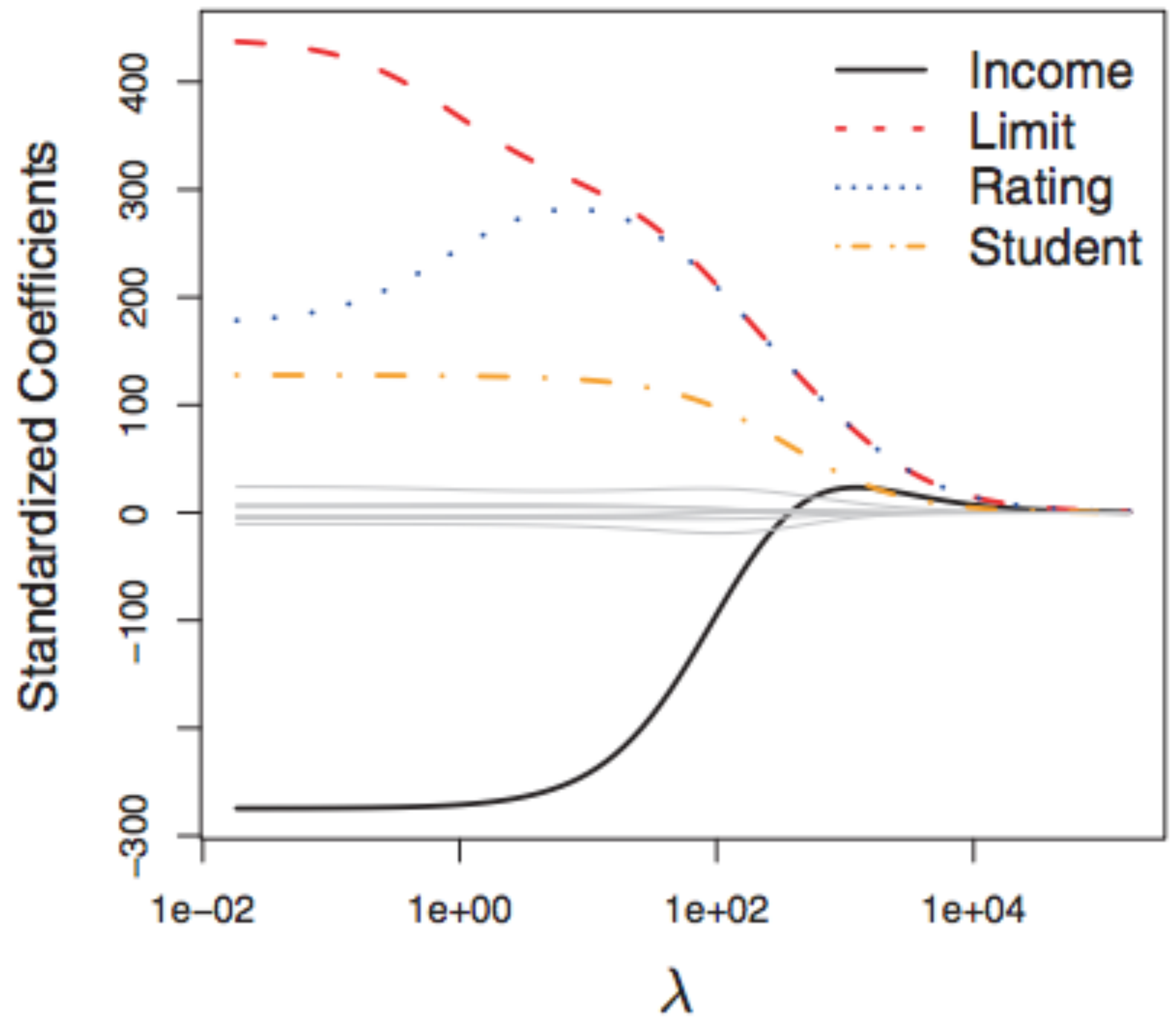


FIGURE 6.4, ISL (8th printing 2017)

Ridge Regression

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

Why is this method an improvement over least squares?

Recall the bias-variance tradeoff.

As $\lambda \to \infty$, the model becomes less flexible. Variance decreases. Bias increases.

Ridge Regression

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

Which curve is bias?
Which is variance?
Which is overall error on the test set?

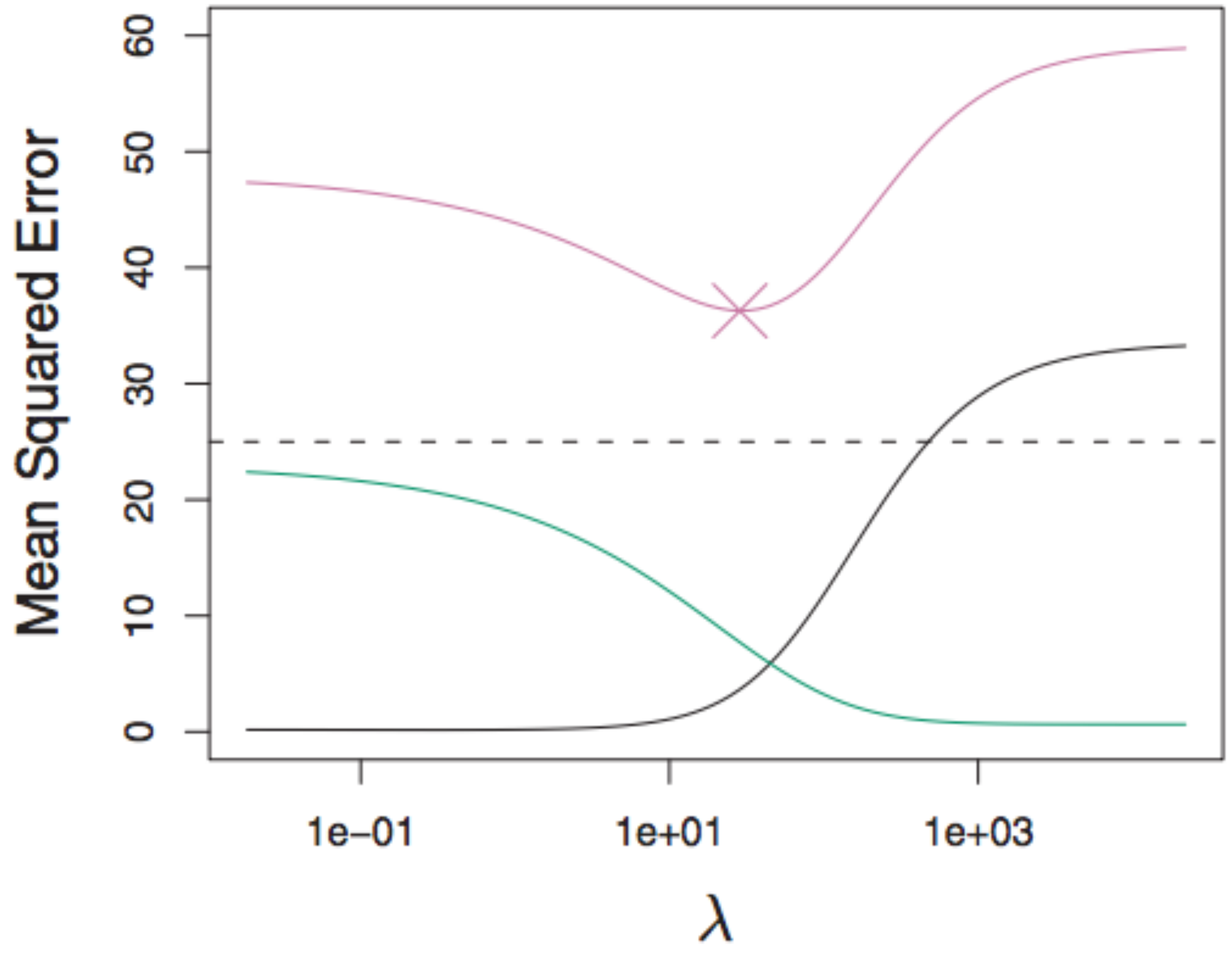


FIGURE 6.5, ISL (8th printing 2017)

As long as we are reducing more variance than adding bias, overall error will decrease.

Ridge Regression

Pro: Computationally feasible, reduces variance in linear regression when $p > n$ or n is not much larger than p, allows for a unique solution when $p > n$

Con: Includes all p predictors in final model, does not perform feature selection (which boosts interpretability) by setting any β to zero

In Search of Sparsity

$$y = \beta_0 + \beta_1 x_1 + \cancelto{0}{\beta_2} x_2 + \beta_3 x_3 + \cancelto{0}{\beta_4} x_4 + \cancelto{0}{\beta_5} x_5 + \beta_6 x_6$$

Offers interpretability when p is large

The Lasso

Find β values that minimize a "penalized" error:

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_1$$

In the case of one feature:

$$\hat{\beta} = \arg\min_{\beta}(y - (\beta_0 + \beta_1 x))^2 + \lambda(|\beta_0| + |\beta_1|)$$

The Lasso

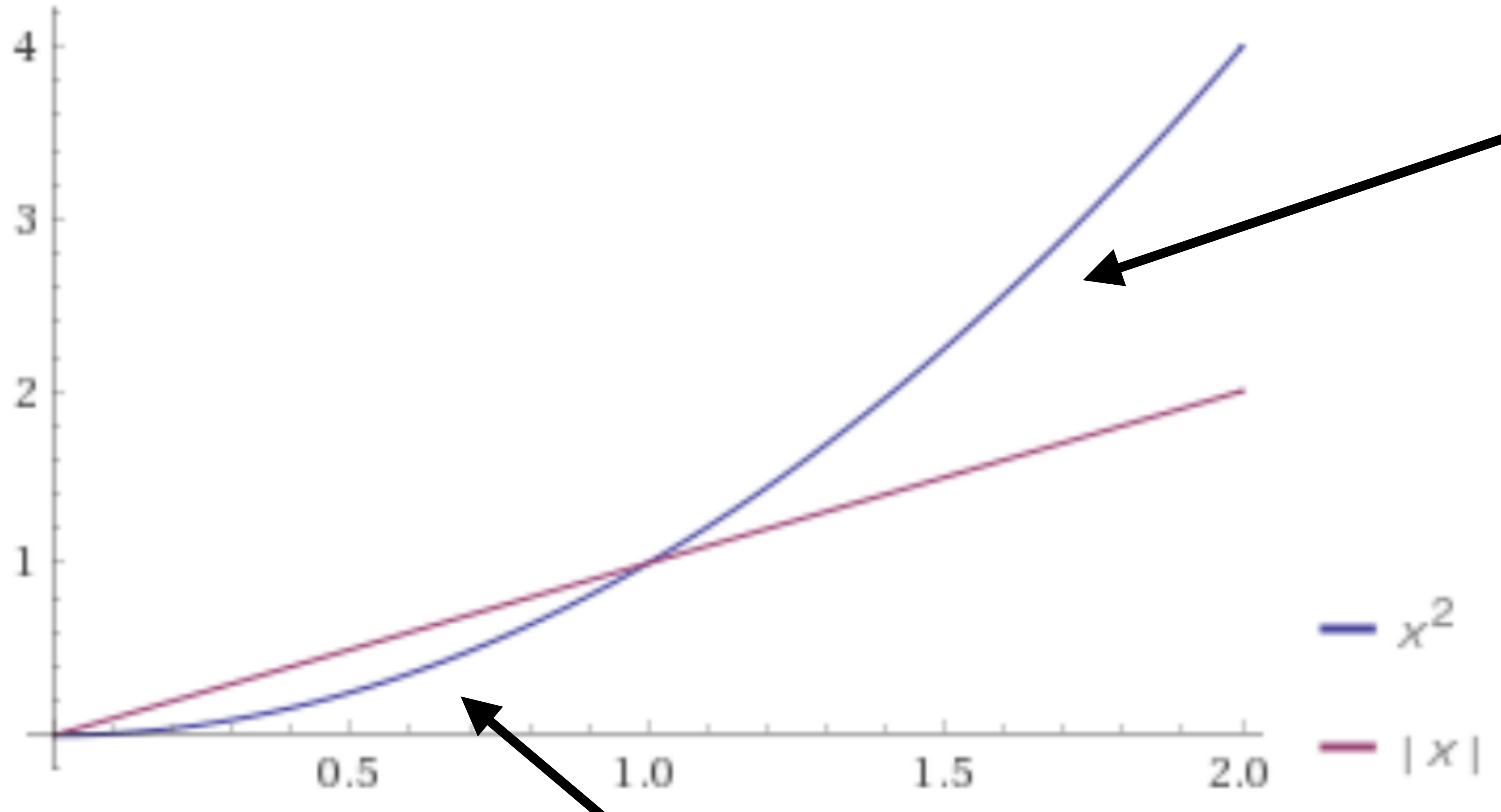
"**L**east **a**bsolute **s**hrinkage and **s**election **o**perator"

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \boxed{\lambda||\vec{\beta}||_1}$$

Penalizes sum of absolute values

Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society. Series B (Methodological) (1996): 267-288.

Ridge vs. Lasso Penalty



Quadratic is harsh here

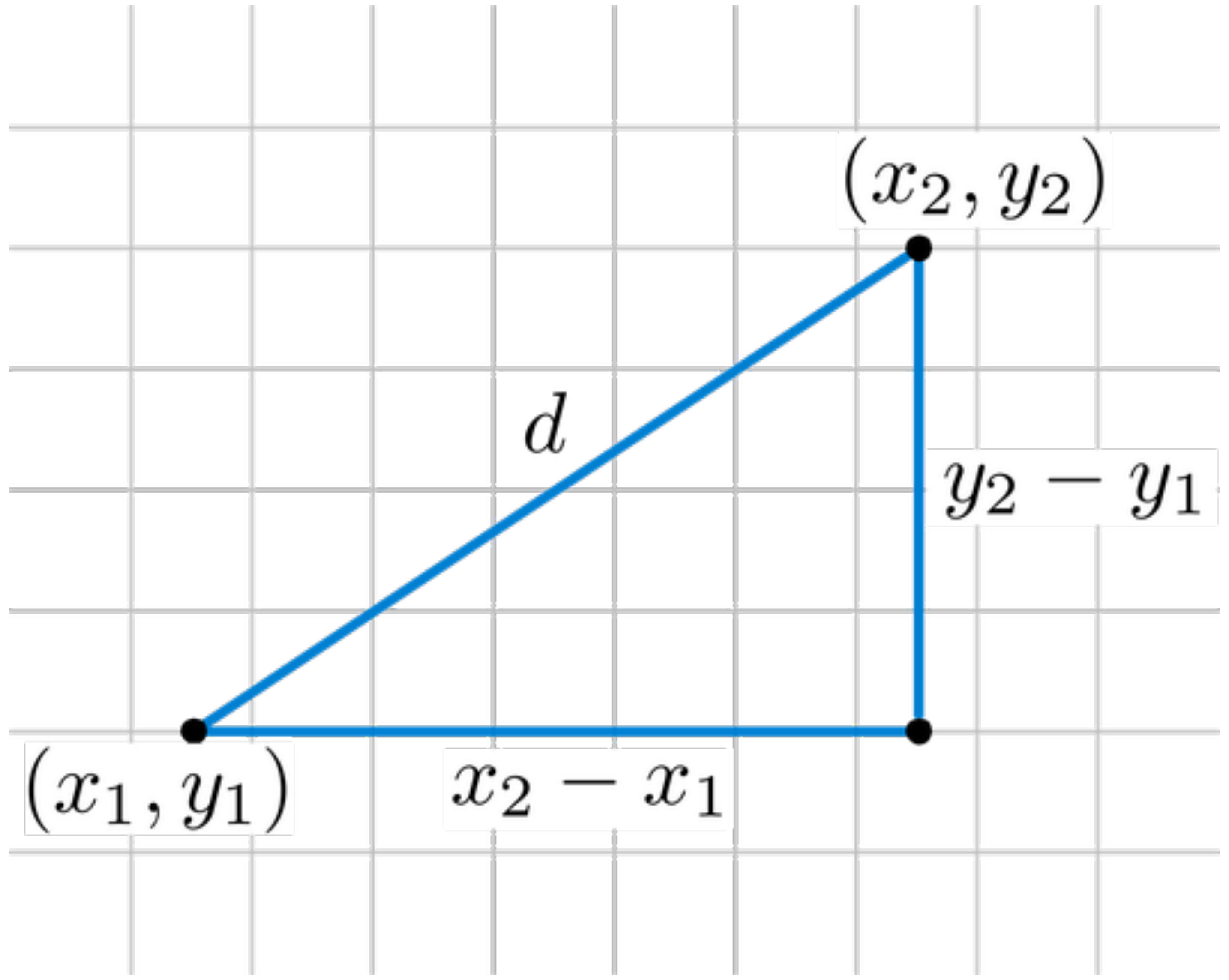
Quadratic is kind here

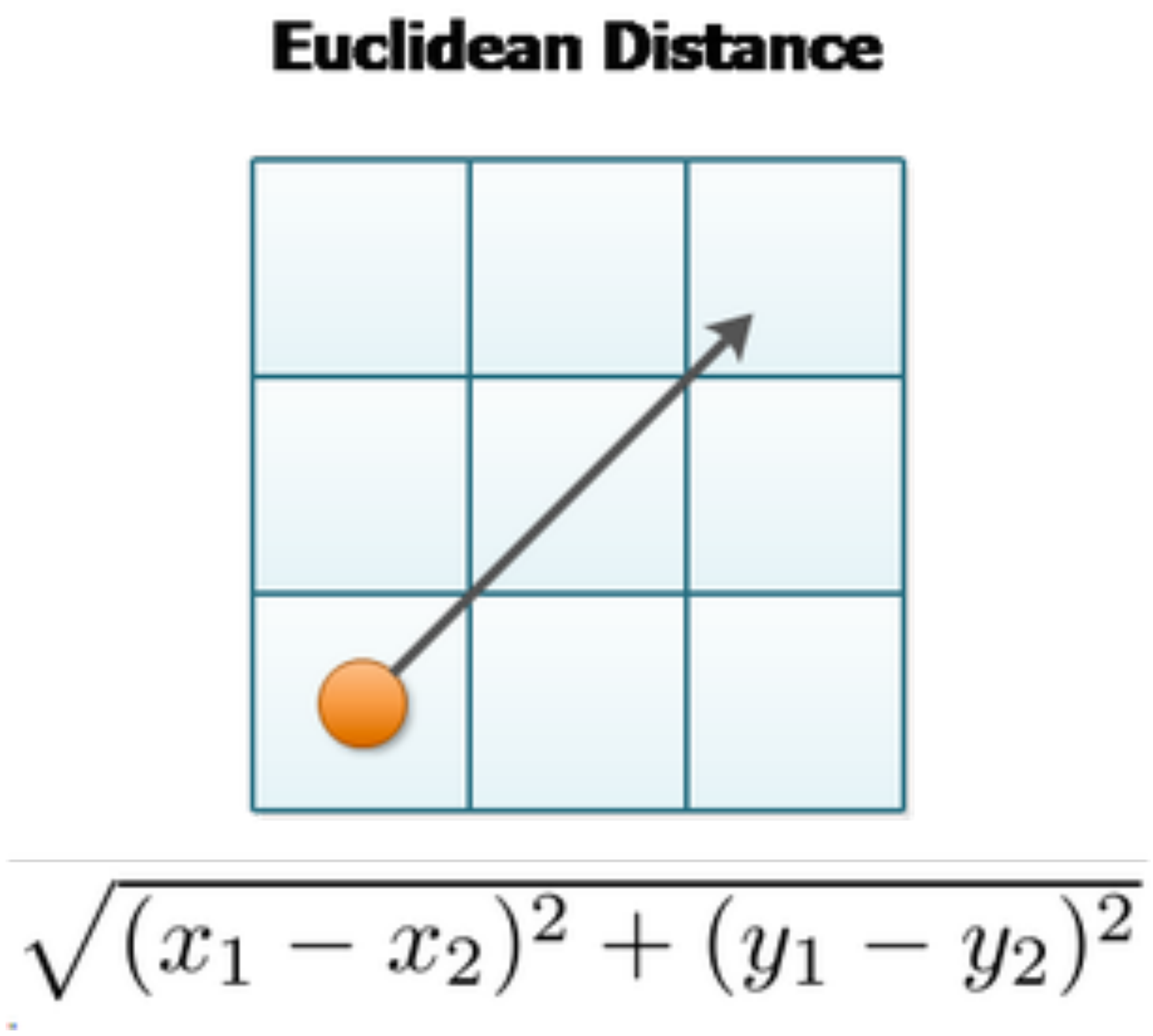
A Detour About Distances

To explain why lasso results in sparse coefficient estimates, we first need to understand different measures of distance.

Different Measures of Distance

We are familiar with Euclidean, or L2, distance.



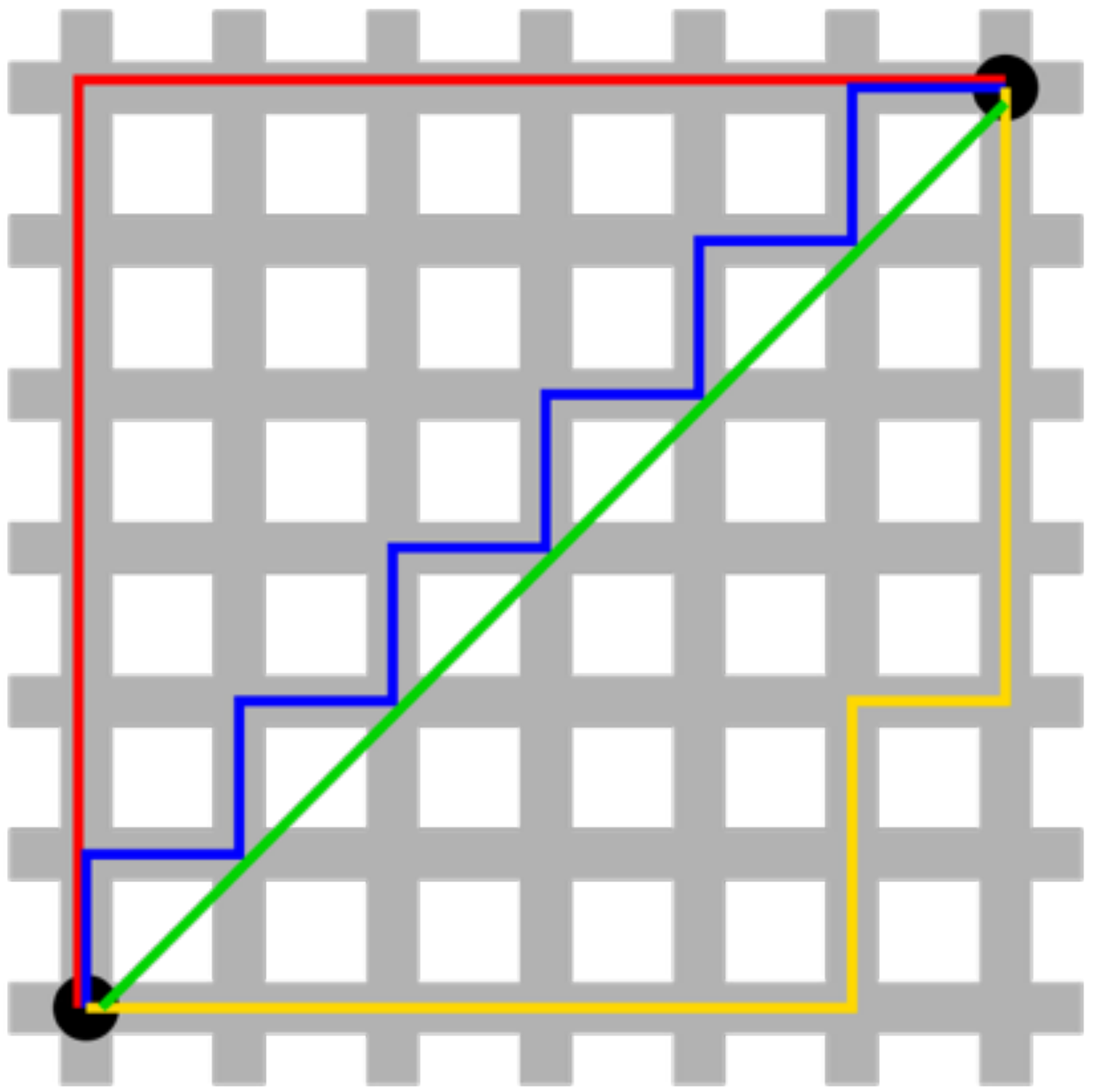


A vector's Euclidean distance from the origin

$$||\vec{x}||_2$$

Different Measures of Distance

Another distance metric is Manhattan, or L1, distance.



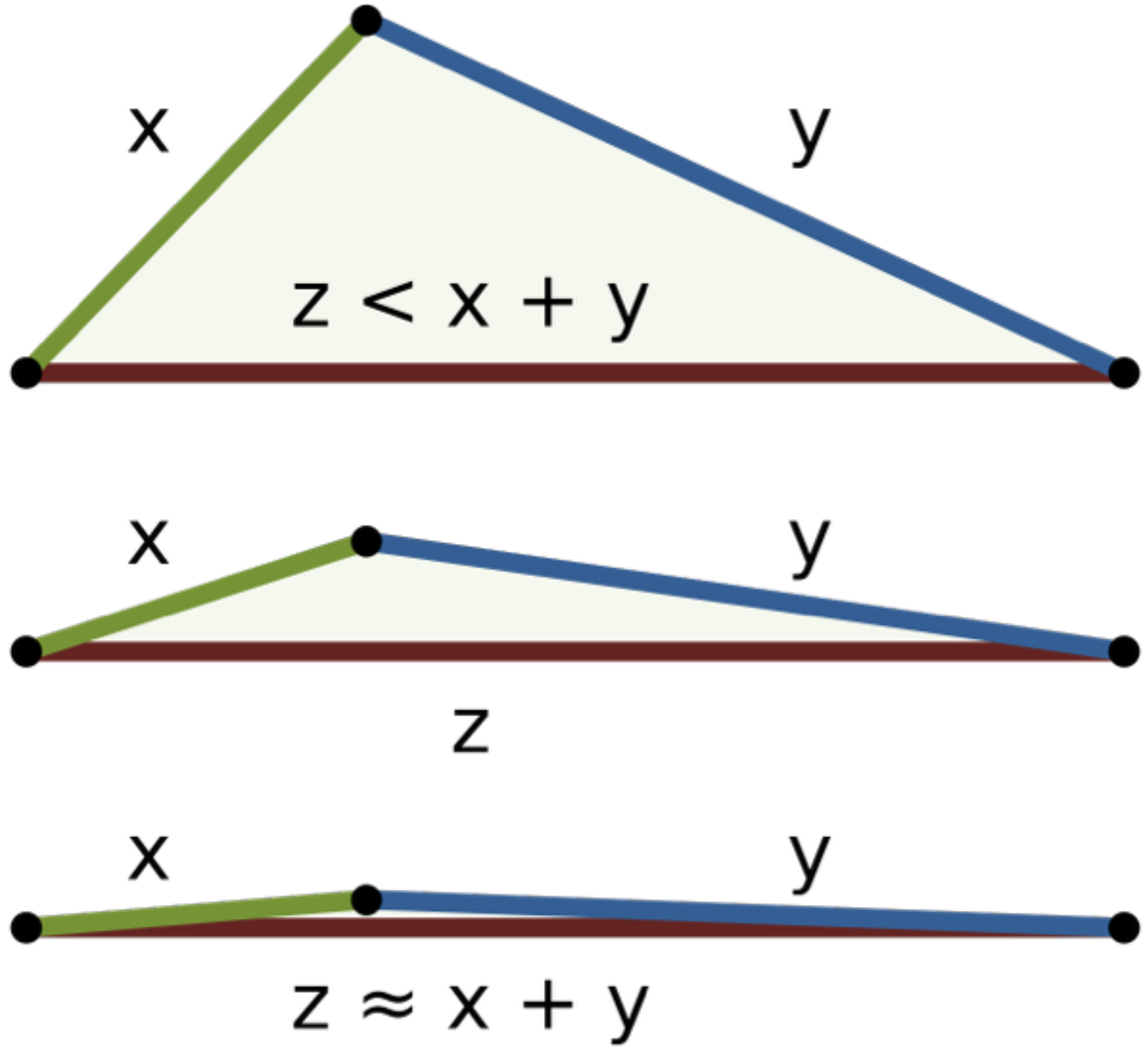
$$|x_1 - x_2| + |y_1 - y_2|$$

A vector's
L1 distance from
the origin

$$||\vec{x}||_1$$

Different Measures of Distance

A distance measure must satisfy the *triangle inequality*.



Distance Measures Are Called Norms

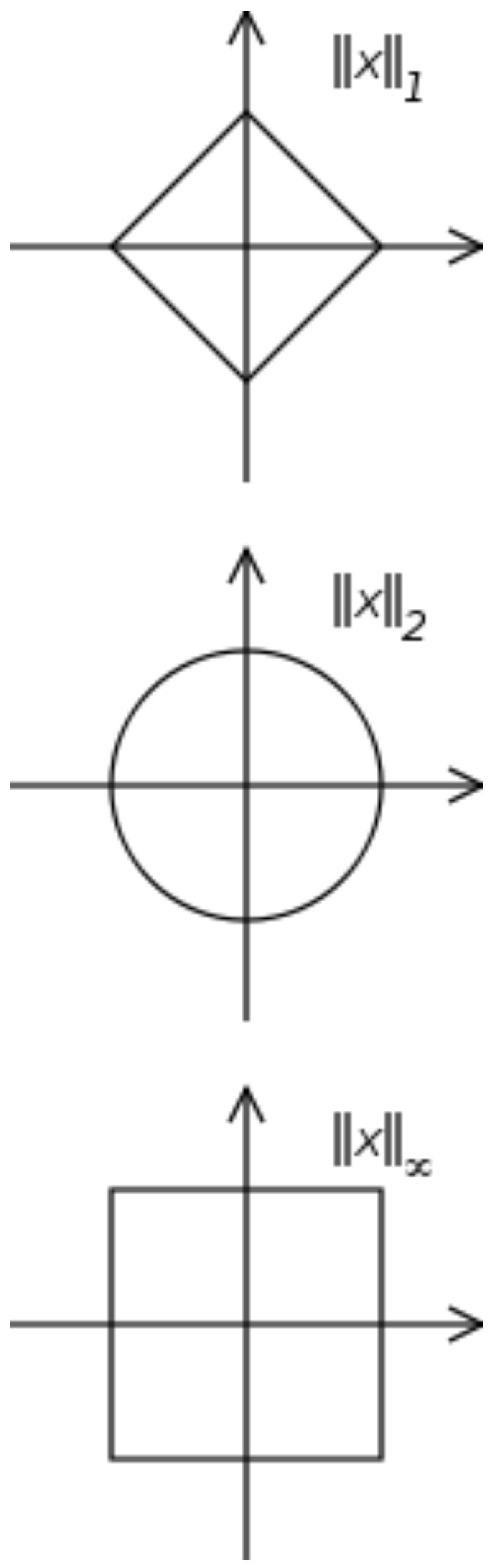
A norm assigns a positive length to a vector.

$$||\vec{x}||_1 = |x_1| + |x_2|$$ "L1-norm"

$$||\vec{x}||_2 = \sqrt{x_1^2 + x_2^2}$$ "L2-norm"

$$||\vec{x}||_p = (|x_1|^p + |x_2|^p)^{\frac{1}{p}}$$ "p-norm"

$$||\vec{x}||_\infty = \max(|x_1|, |x_2|)$$ "∞-norm"



Norms

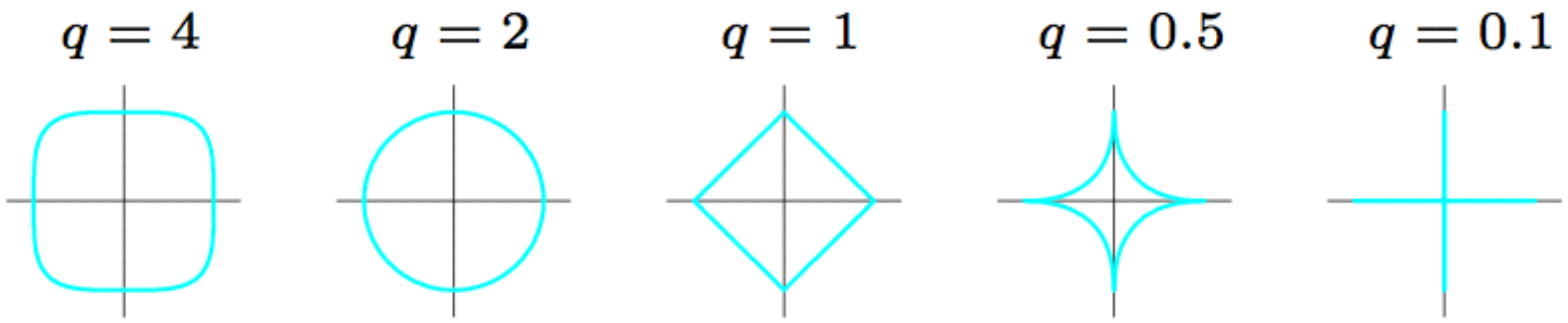


FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

Hastie, Travor et al. The Elements of Statistical Learning. Vol. 2. No. 1. New York: Springer, 2009

Norms and Sparsity

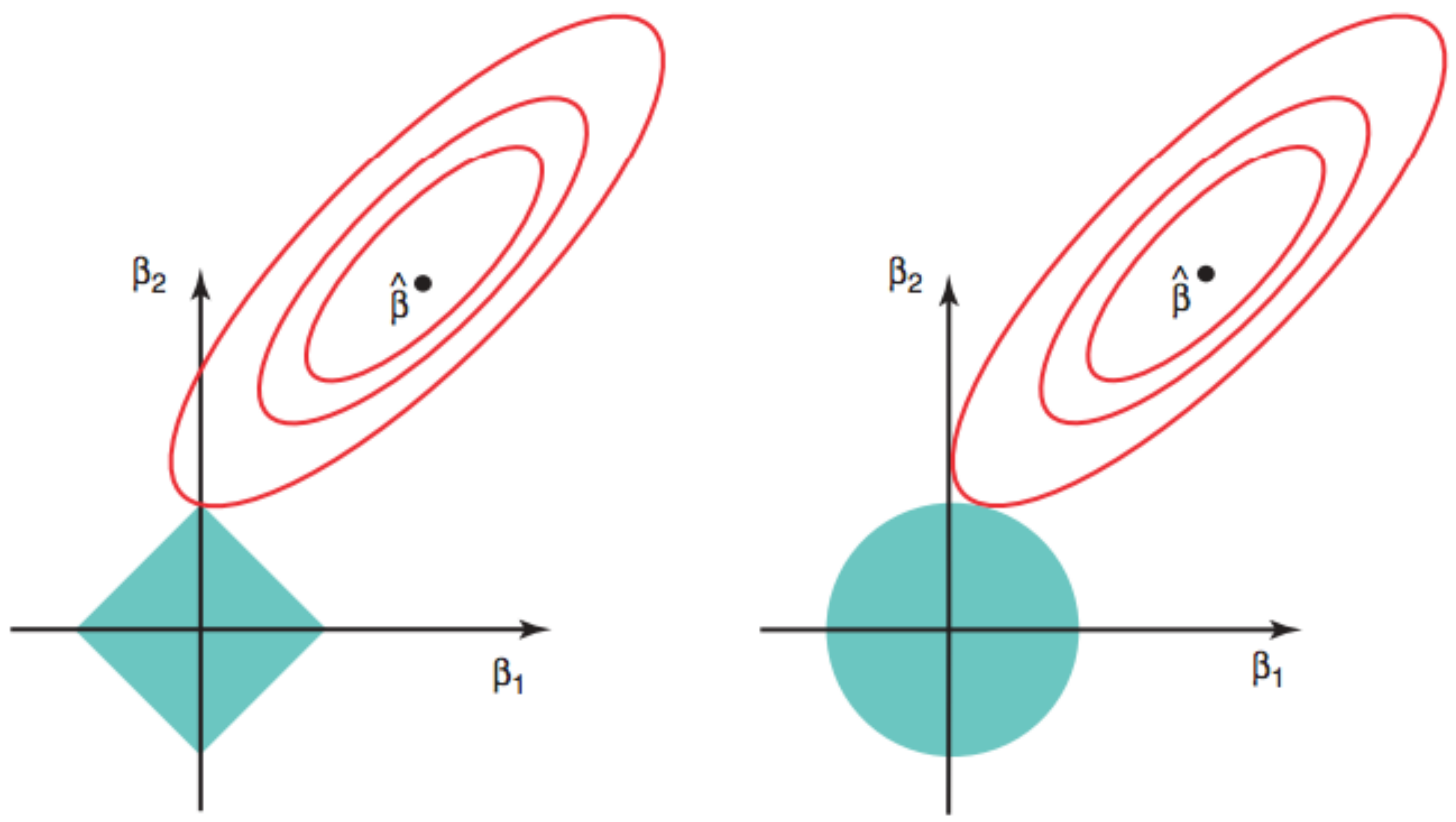
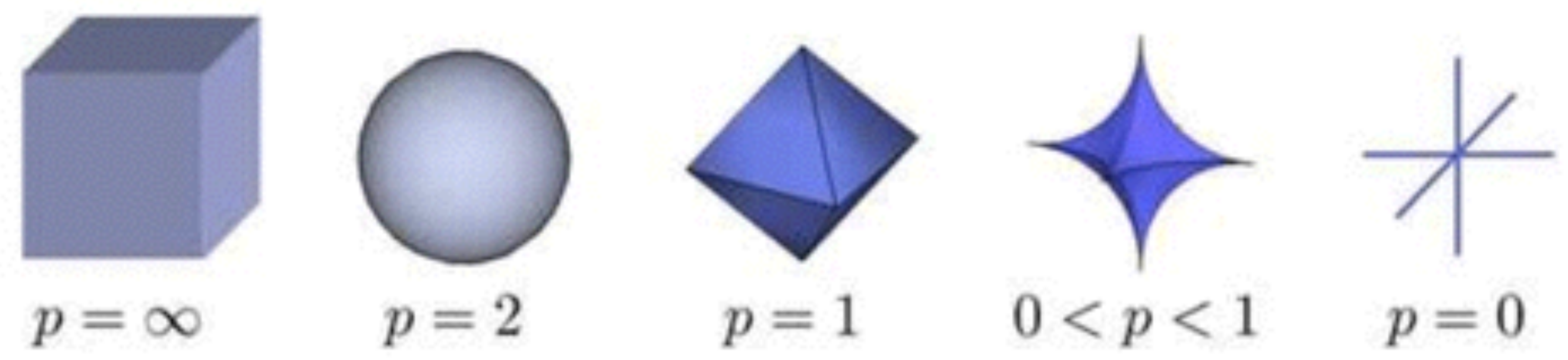


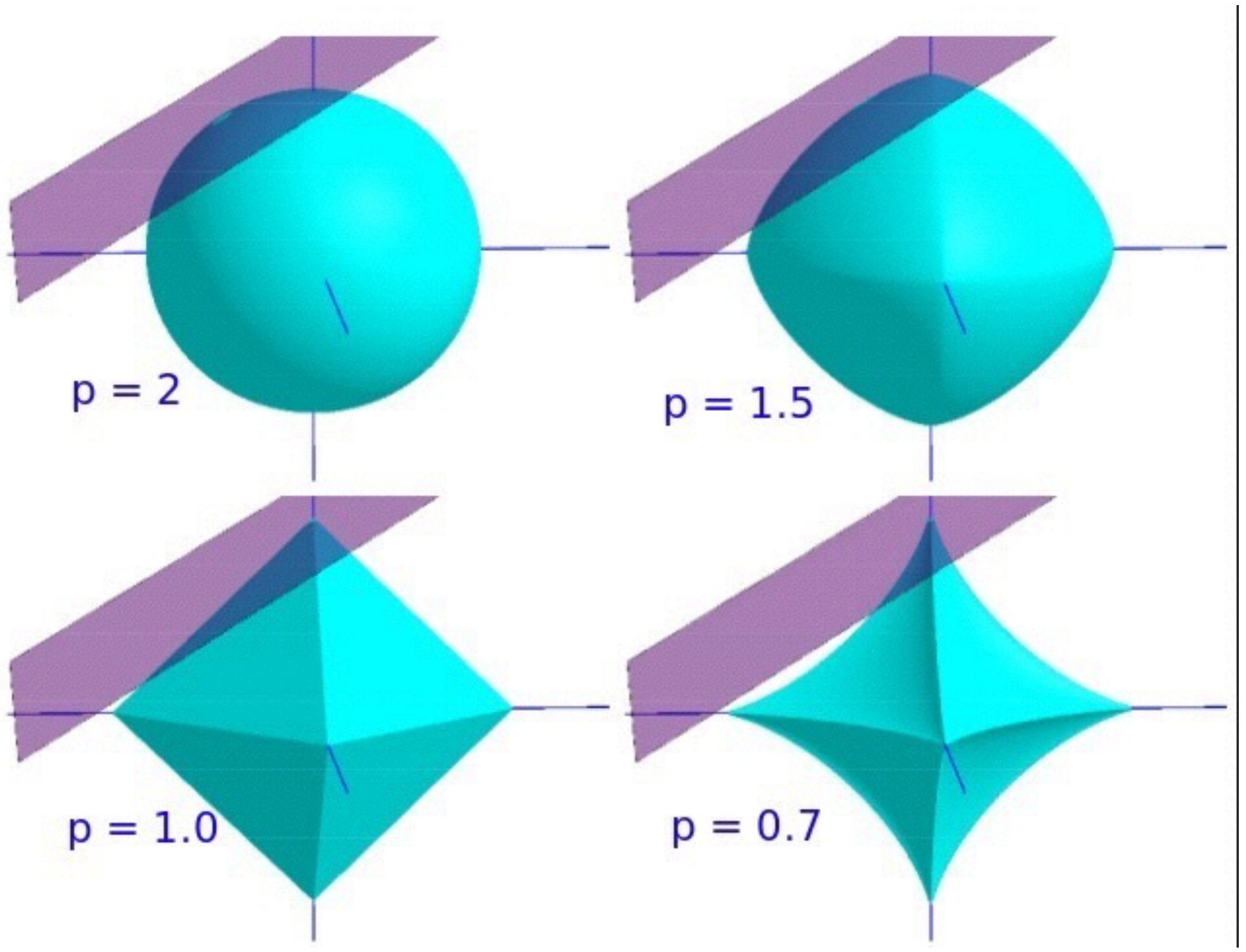
FIGURE 6.7, ISL (8th printing 2017)

L1 penalty says: “Find the best model whose coefficients are inside this diamond.” (Lasso)

L2 penalty says: “Find the best model whose coefficients are inside this circle.” (Ridge)

Norms and Sparsity





Norms with sharper corners on the axes yield sparser solutions.

Lasso vs. Ridge Penalty

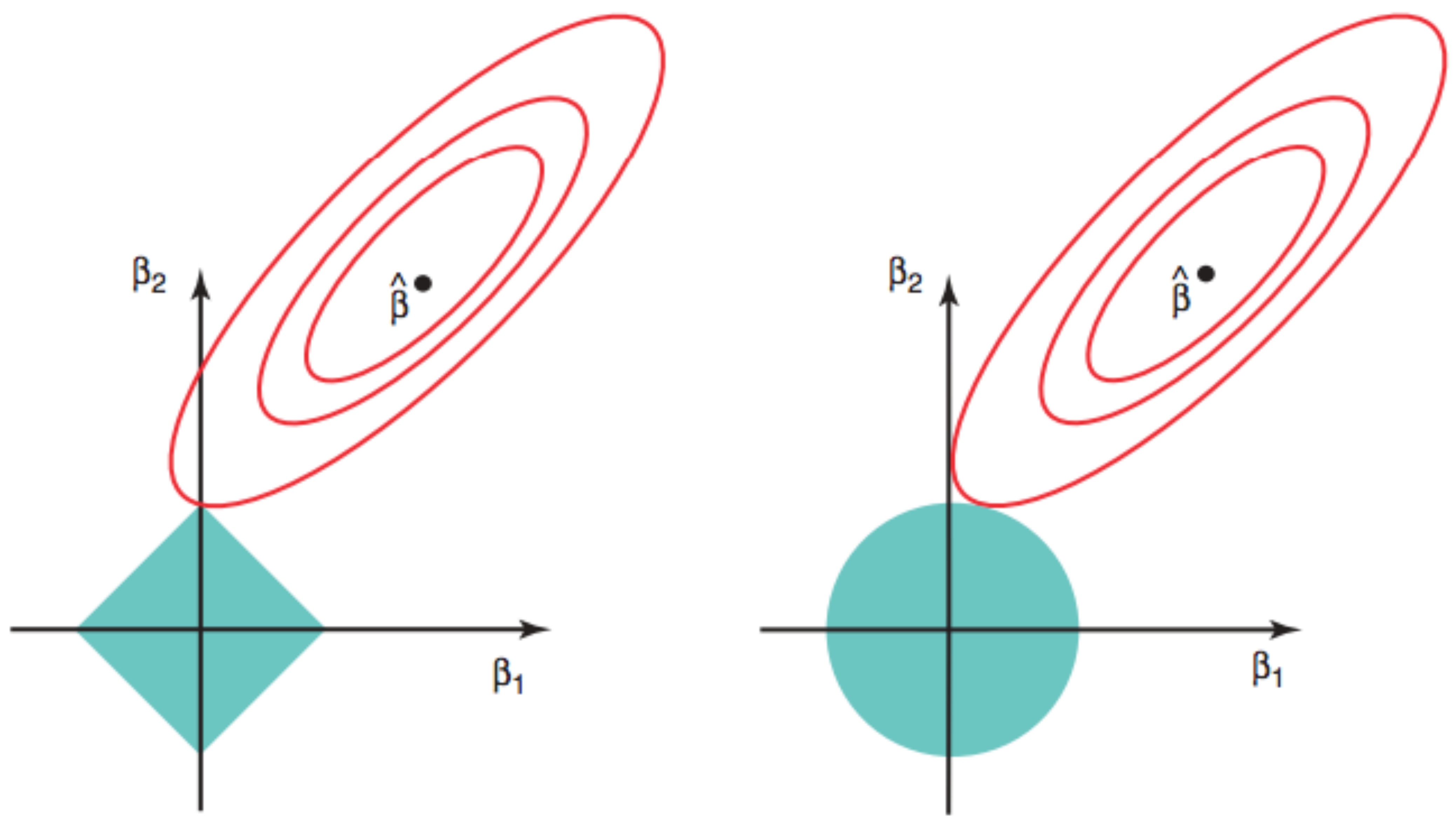


FIGURE 6.7, ISL (8th printing 2017)

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_1$$

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

Lasso finds coefficients that are exactly zero.

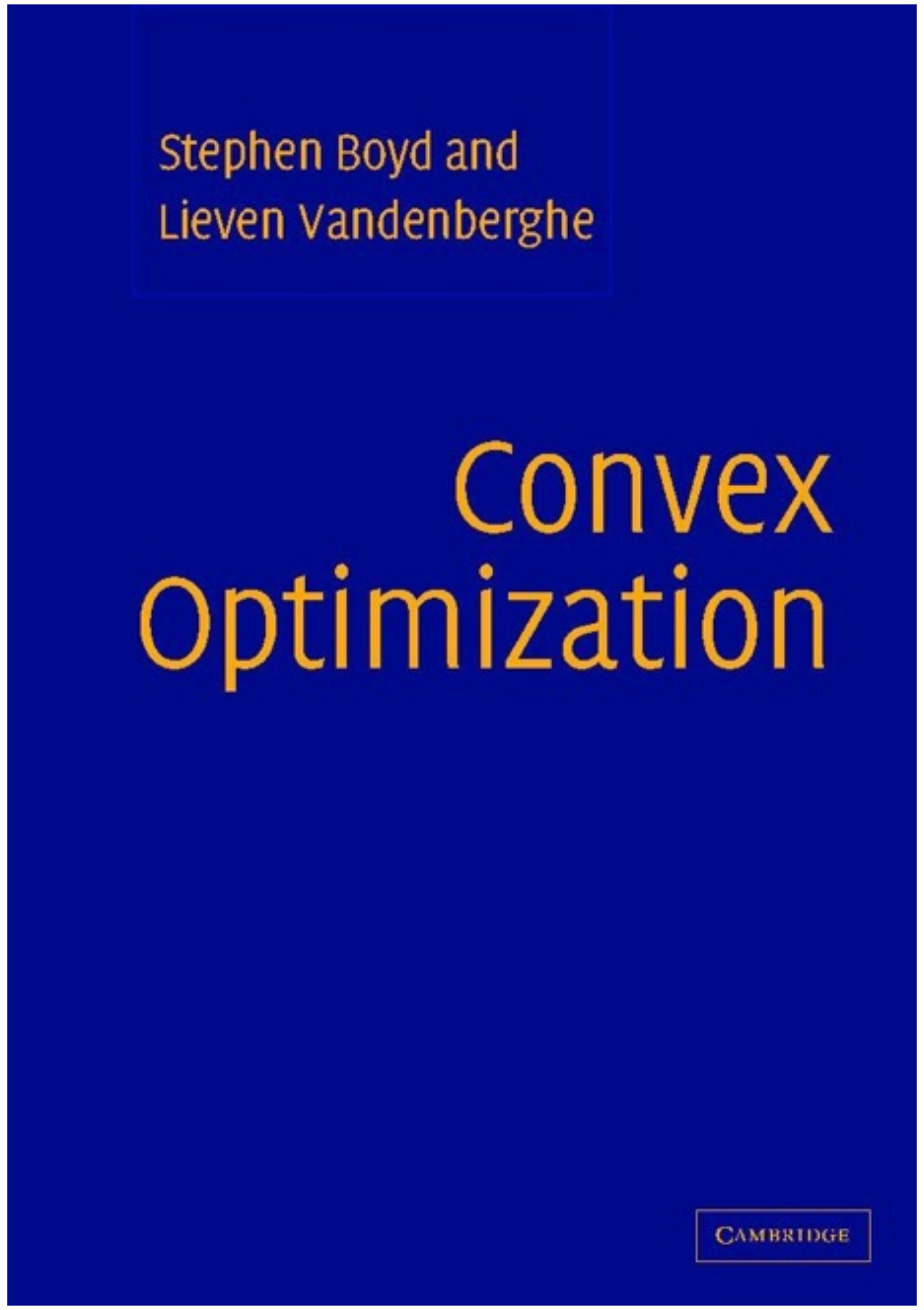
Why not use a p-norm with p<1?

By the logic “pointier norms = sparser coefficients”, why not use an even pointier norm, i.e. p-norm with $0 < p < 1$?

Convexity

Popular Stanford course: EE 364

convex convex convex concave concave



Applied mathematicians have figured out efficient ways to maximize and minimize convex functions.

L2 and L1 norms are both convex.

A p-norm with $0 < p < 1$ is not convex.

The Lasso

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_1$$

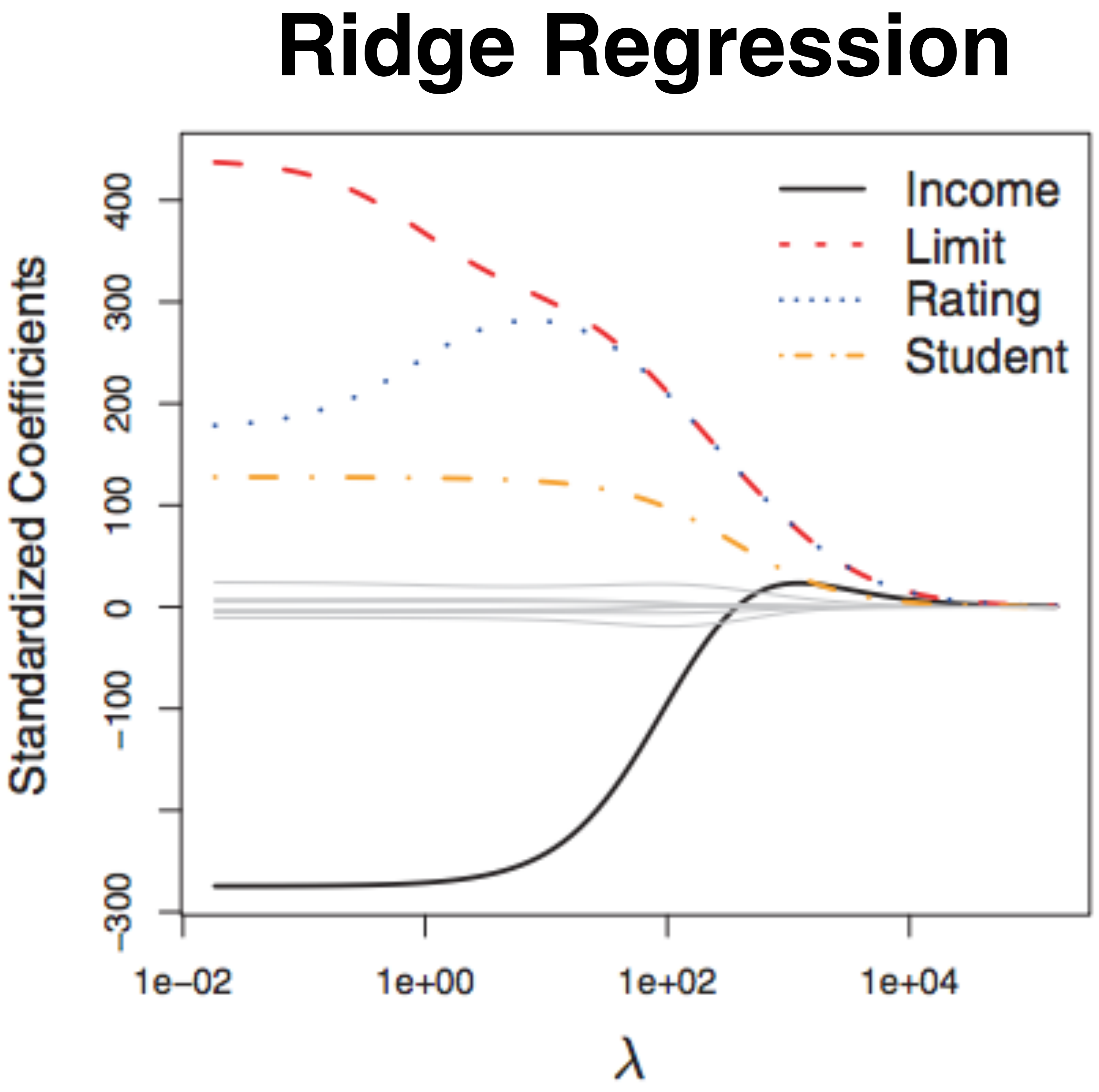


FIGURE 6.4, ISL (8th printing 2017)

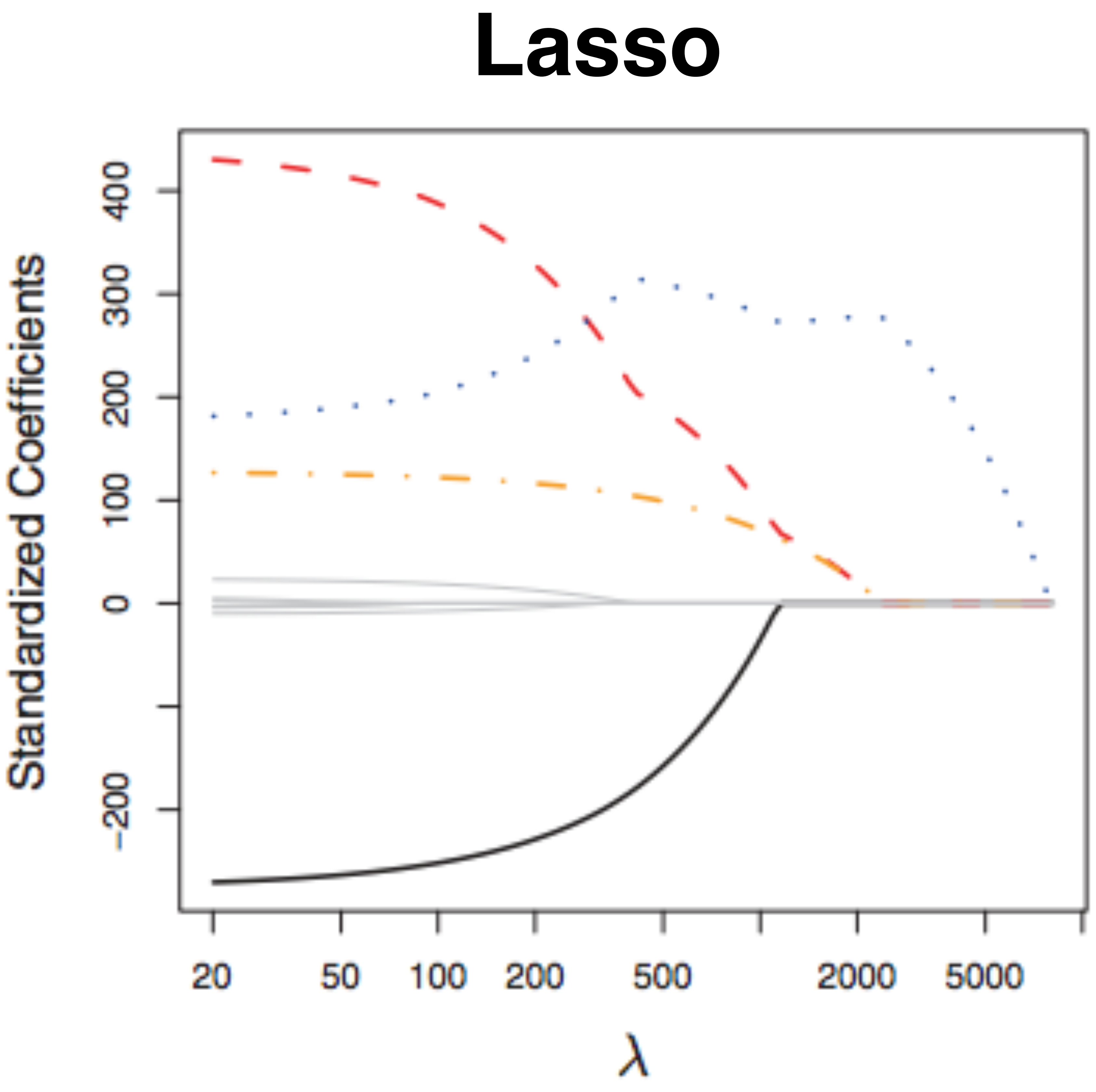


FIGURE 6.6, ISL (8th printing 2017)

The Lasso

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2$$

Ridge Regression

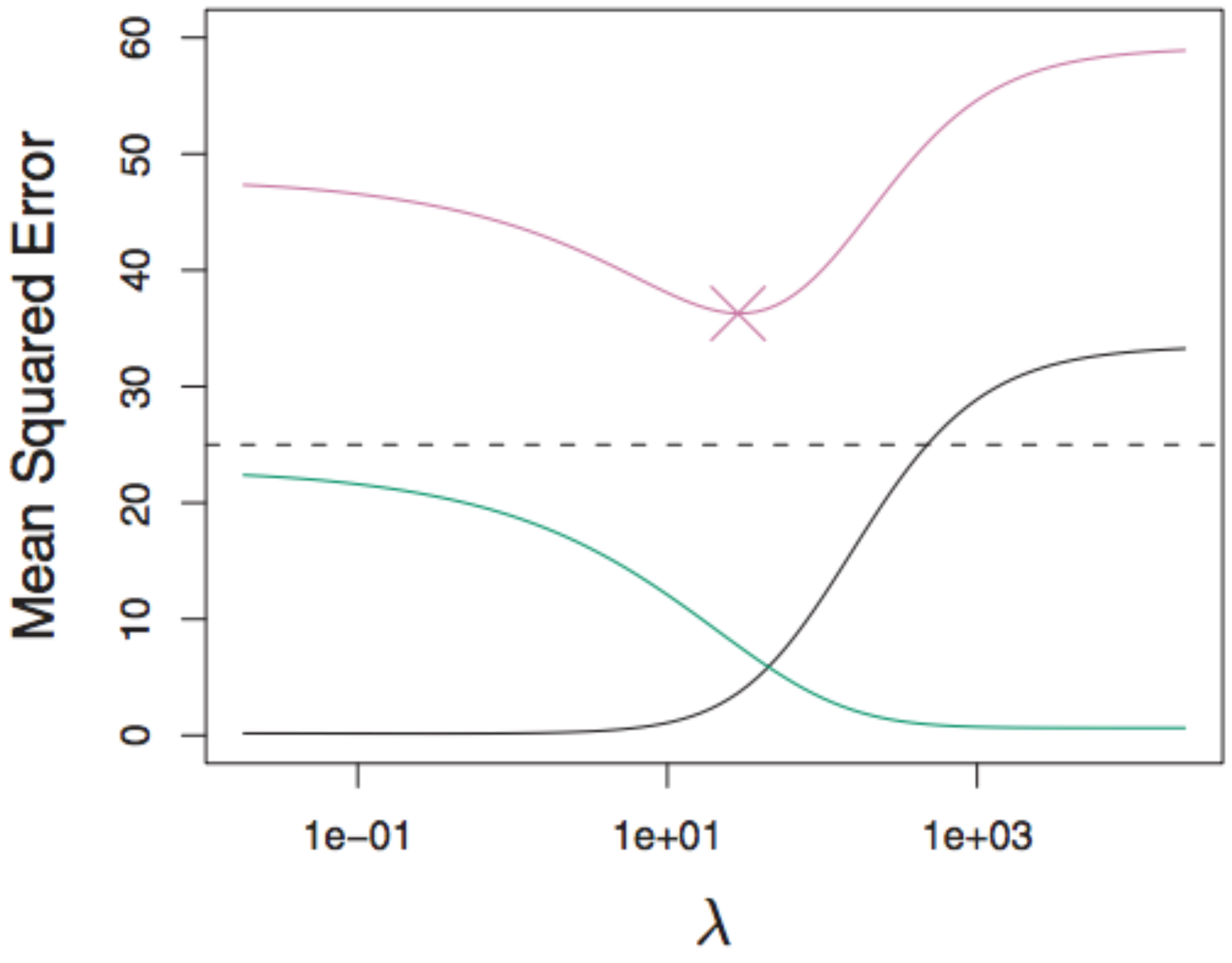


FIGURE 6.5, ISL (8th printing 2017)

Lasso

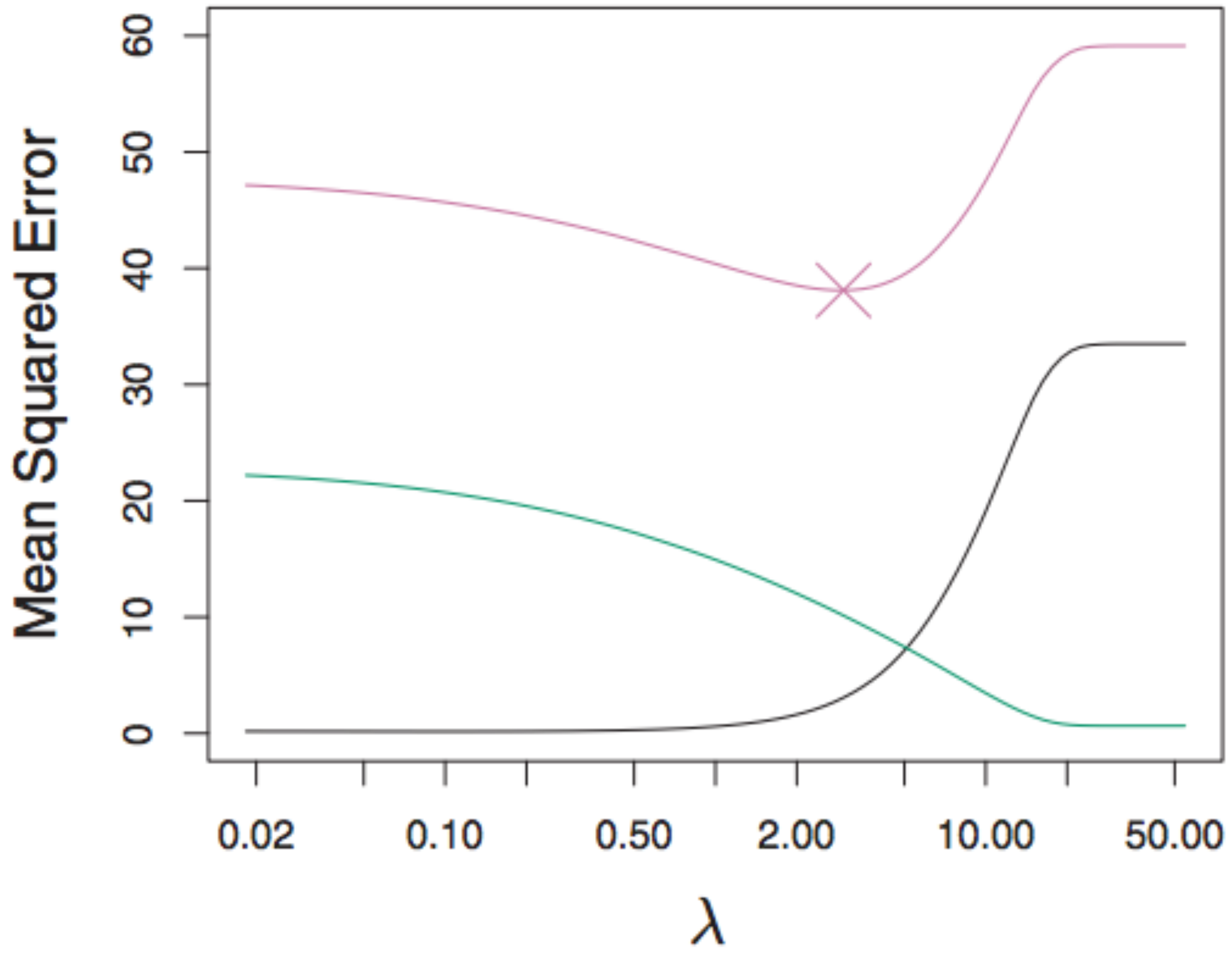


FIGURE 6.8, ISL (8th printing 2017)

Like ridge, lasso reduces model variance as λ increases.

Ridge vs. Lasso: Which is better?

Neither is universally better than the other.

Ridge regression will perform better when the response is a function of many predictors, all with coefficients roughly of similar magnitude.

Lasso will perform better when a small number of predictors have substantial coefficients, and the remaining are small or equal zero.

The Lasso

Pro: Computationally feasible, reduces variance in linear regression when $p > n$ or n is not much larger than p, allows for a unique solution when $p > n$, performs feature selection (offers interpretability)

Con: Not as good as ridge when all predictors have significant and roughly similarly sized coefficients

Bayesian Prior for Ridge & Lasso

We can view ridge regression and the lasso through a Bayesian lens.

Bayesian probability is an interpretation of the concept of probability. Instead of a fixed frequency, probability is interpreted to represent a state of knowledge or as quantification of a personal belief.

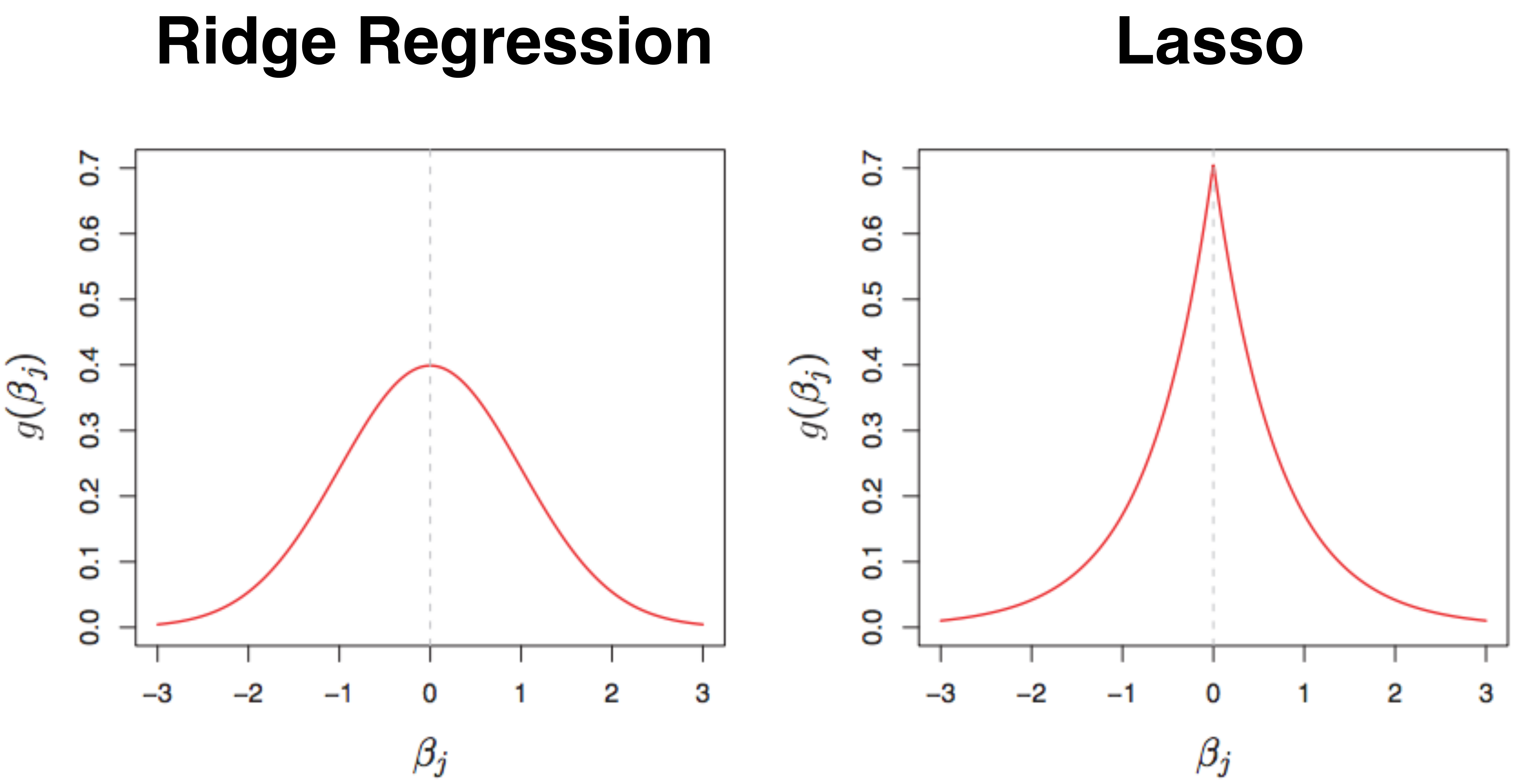
Bayesian Prior for Ridge & Lasso

A *prior distribution* expresses one's belief about a quantity before evidence is taken into account.

For model coefficients, a Bayesian viewpoint says that β has a prior $\Pr(\beta)$.

Bayesian Prior for Ridge & Lasso

Normal distribution



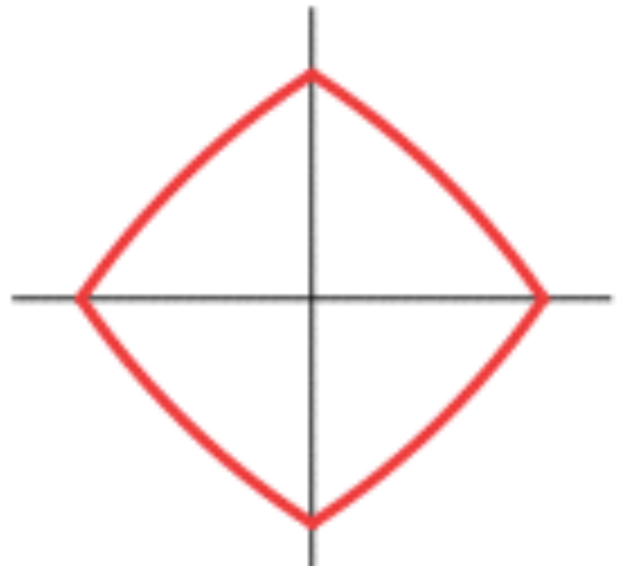
Laplace distribution

FIGURE 6.11, ISL (8th printing 2017)

Elastic Net

Elastic Net = Ridge + Lasso

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2 + \lambda_2||\vec{\beta}||_2^2 + \lambda_1||\vec{\beta}||_1$$



Shrinkage Methods in `sklearn`

```
from sklearn.linear_model import LogisticRegression

ridge = LogisticRegression(penalty='l2', C=1.0)

lasso = LogisticRegression(penalty='l1', C=1.0)
```