CSCI4622 Machine Learning

2020 Spring

Improving Training

Geena Kim

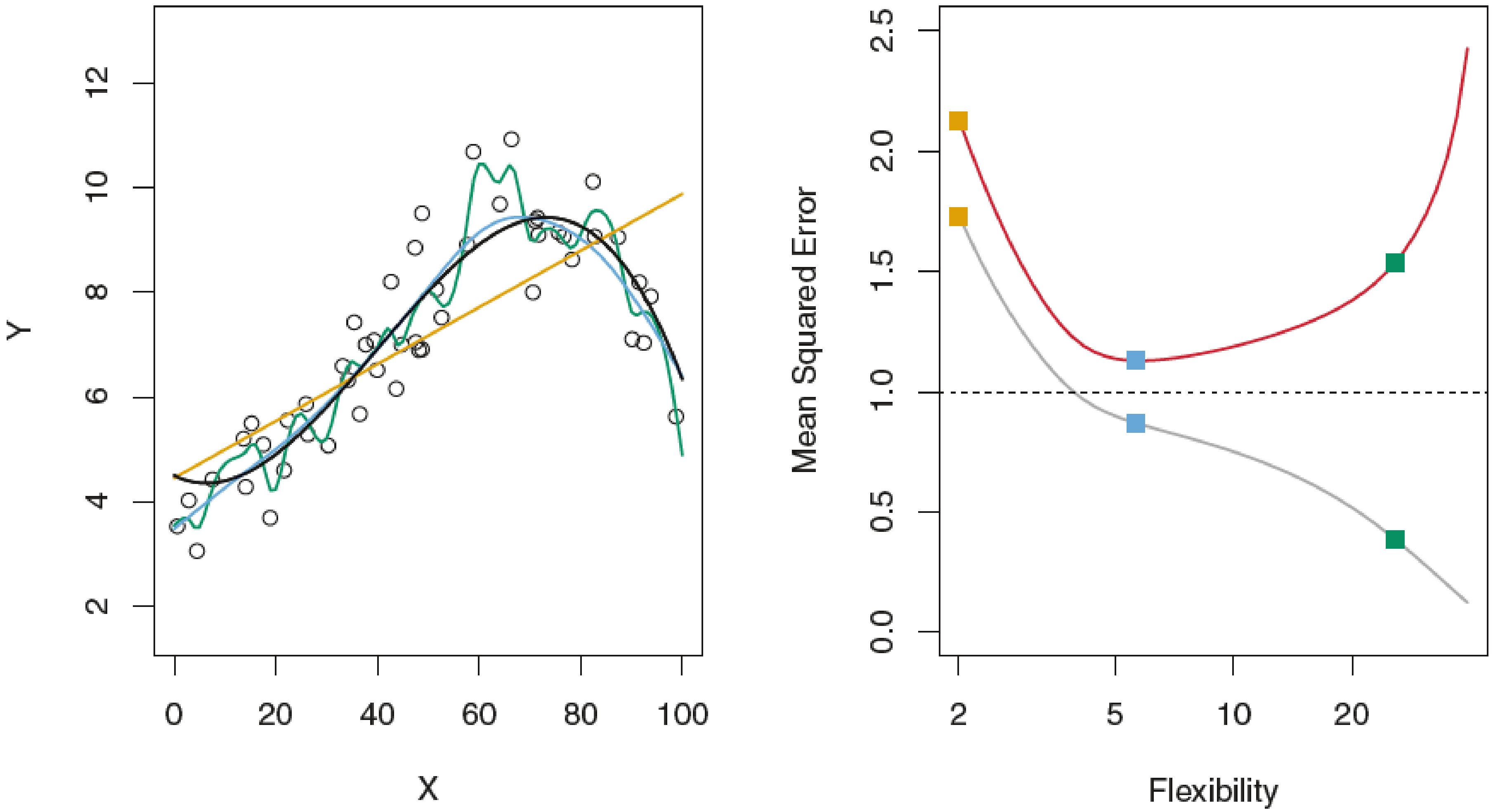
*Slide contents adopted from ISLR material

Better training

The Goals:

- Smallest generalization error
- Better test performance score

Generalization error



Where is the error coming from?

E.g. In regression…

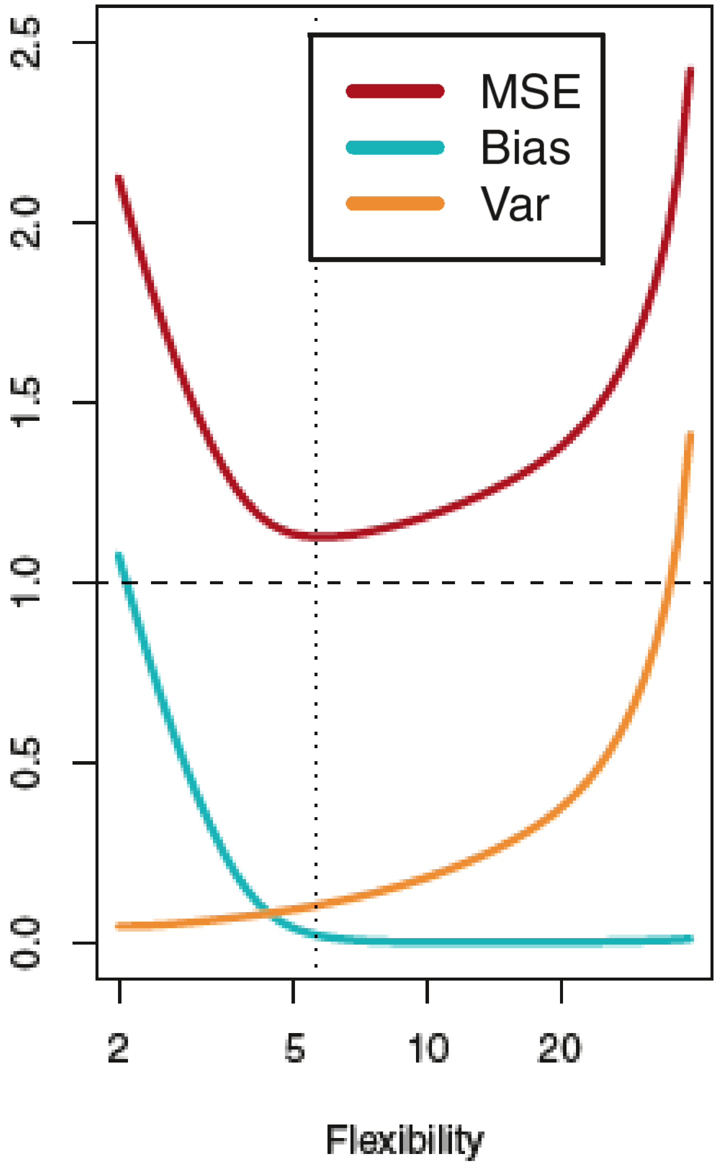
$$y = f(x) + \epsilon$$

$$MSE = \mathbb{E}\left[(y - \hat{f}_S(x))^2\right]$$

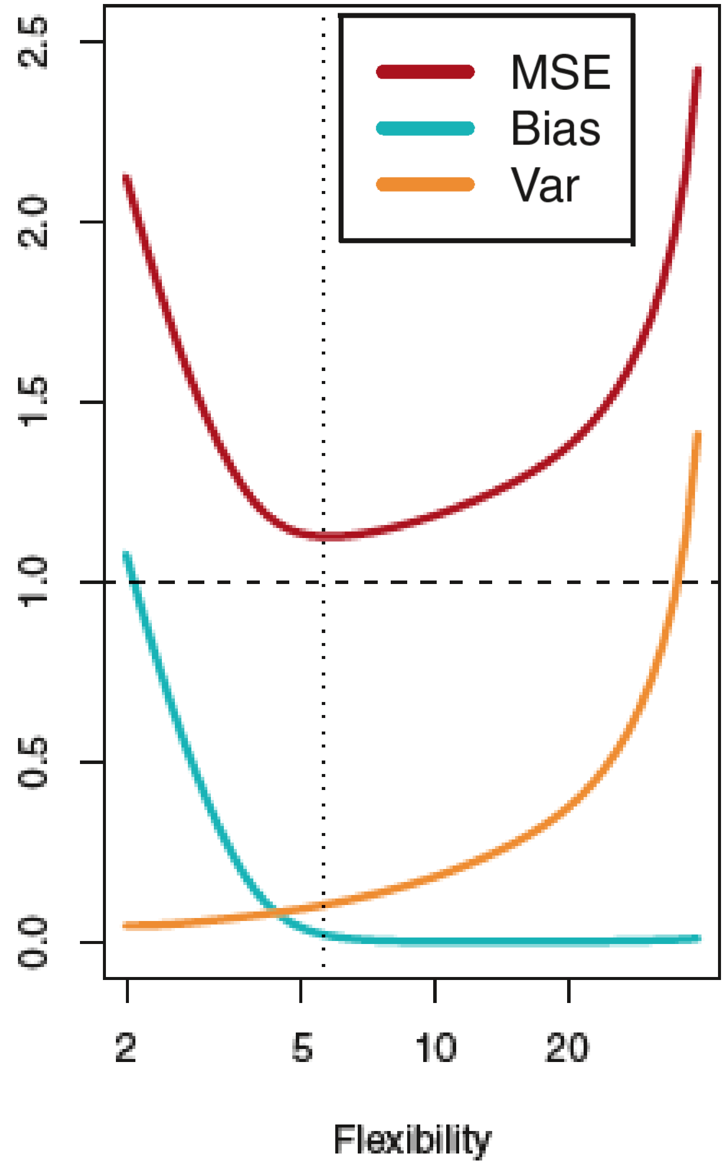
$$= Var(f(x) - \hat{f}_S(x)) + Var(\epsilon) + \left(\mathbb{E}[f(x)] - \mathbb{E}[\hat{f}_S(x)]\right)^2$$

$$+ \mathbb{E}^2[\epsilon] + 2\mathbb{E}[\epsilon]\mathbb{E}[f(x)] - 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)]$$

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$



How do we know which term to drop/include?



- Parameters
- Design parameters

What features to include?

Method 1. Best subset method

- The idea: test all possible combinations
- Curse of dimensionality!

Method 2. Regularization

Regularization

Original loss function

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2$$

Let's penalize some terms that are not necessary

With a L2 regularization

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad \lambda \geq 0$$

L2 regularization (Ridge)

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Also called Ridge regression

What does the lambda (λ) do?

L2 regularization

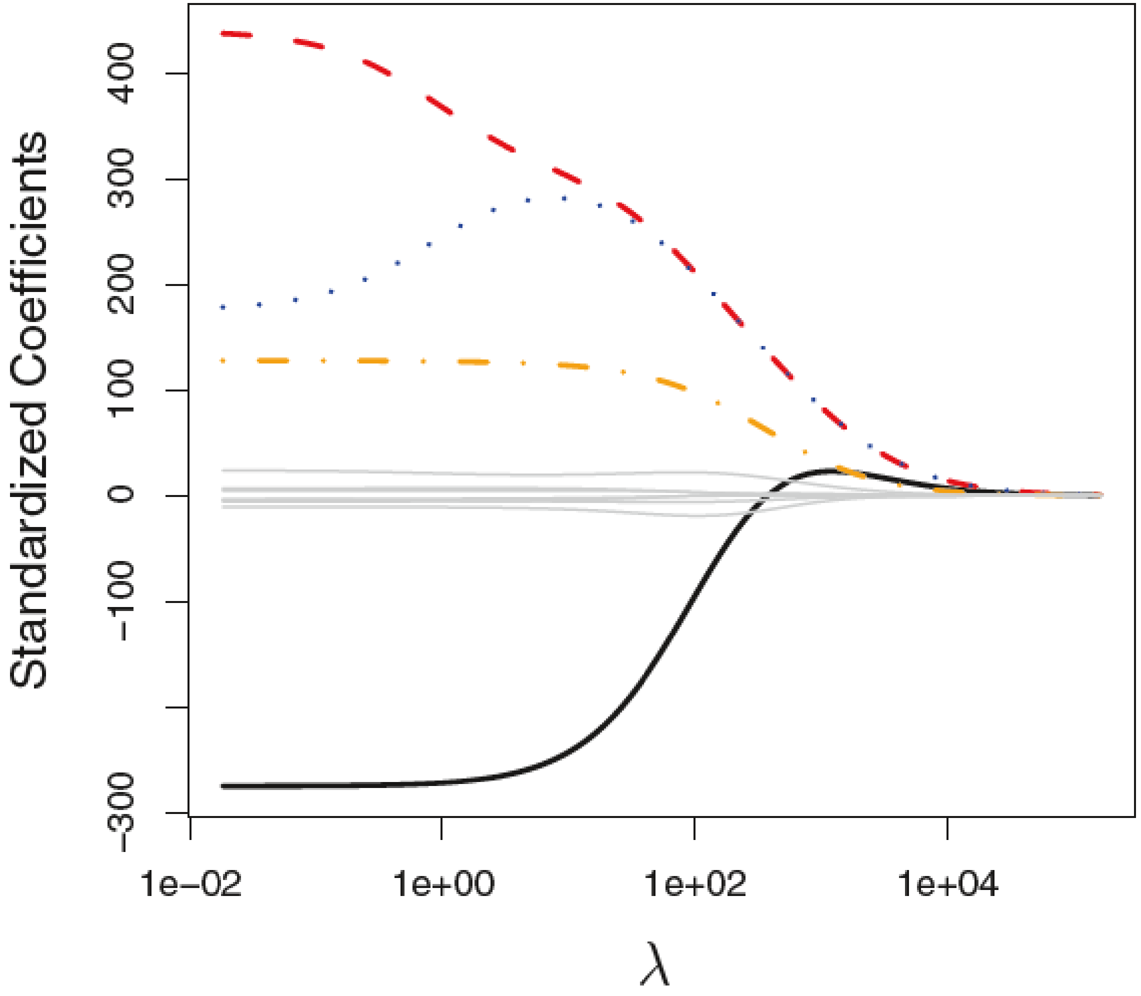
What does the lambda (λ) do?

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

λ	$\lvert\beta\rvert$	Total Loss (L)	Original Loss (L_0)

L2 regularization

What does the lambda (λ) do?

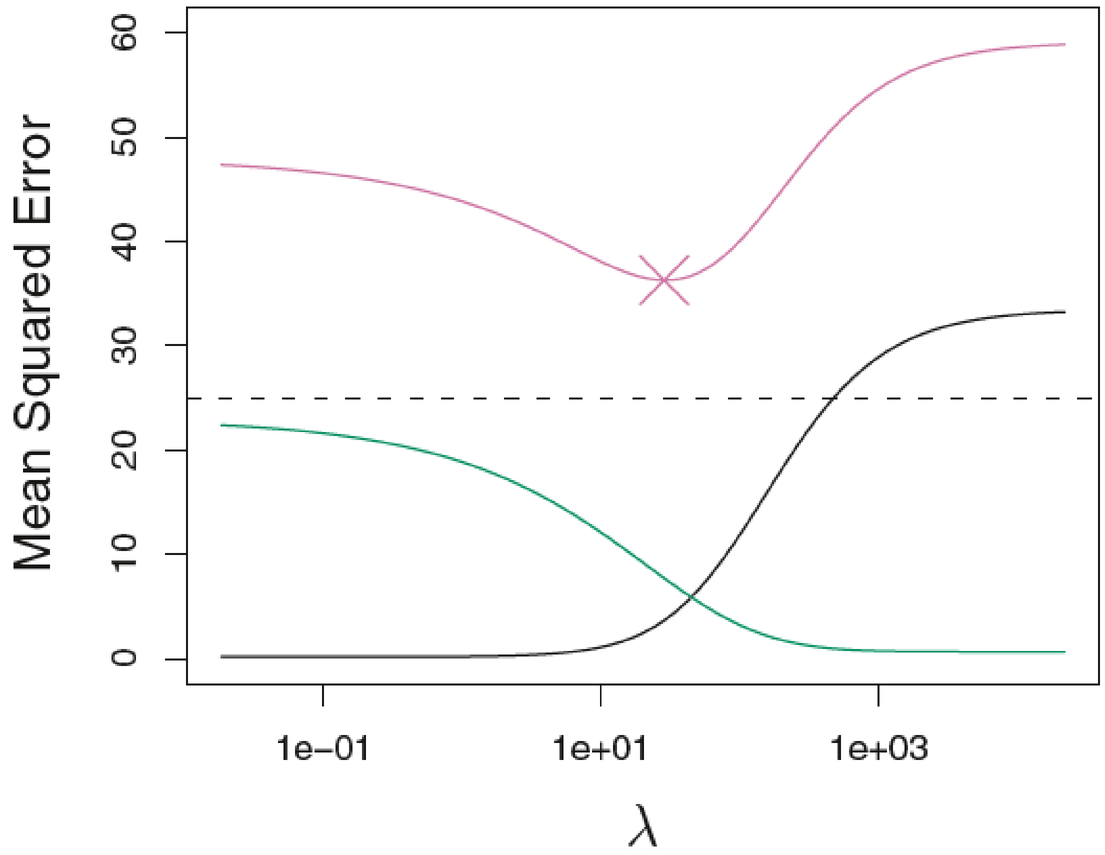


λ vs. $|\beta|_2$

λ vs. β_j

L2 regularization

What does the lambda (λ) do?



λ vs. MSE (L_0)

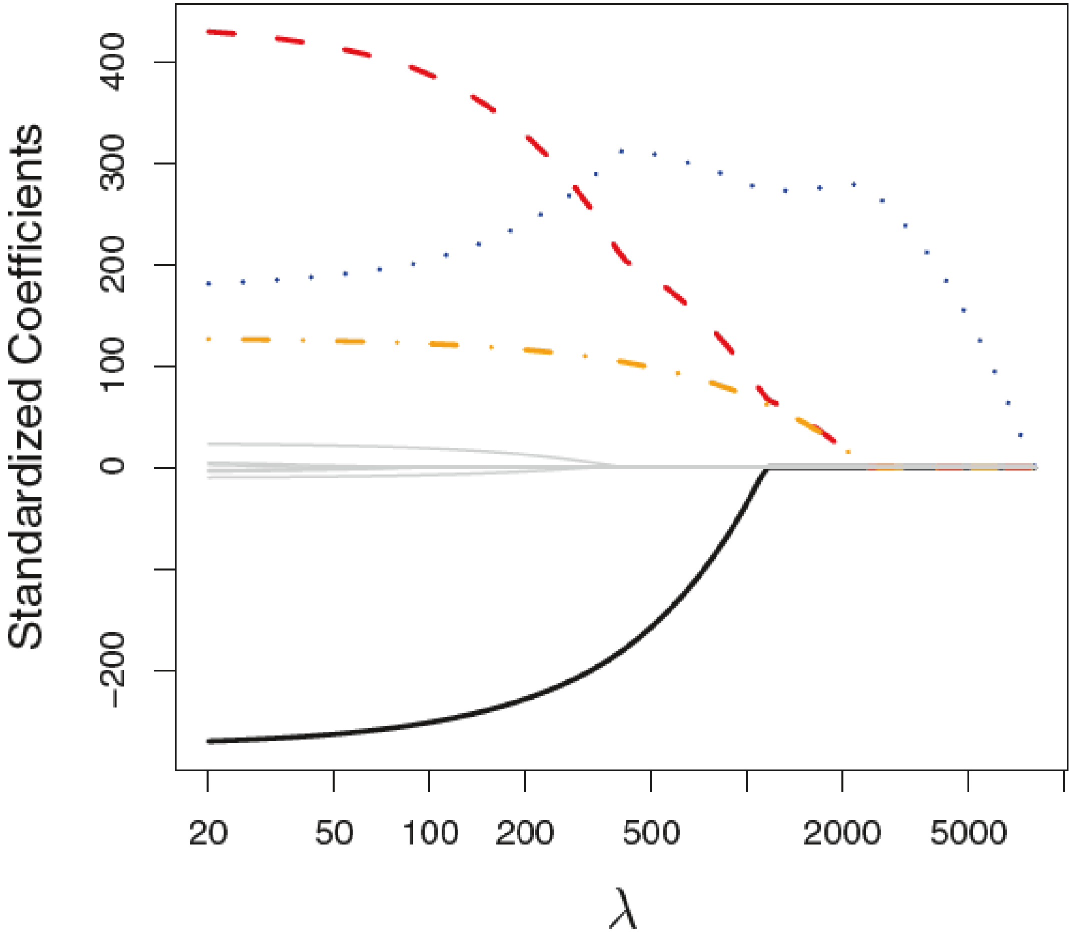
λ vs. bias and variance

L1 regularization (Lasso)

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

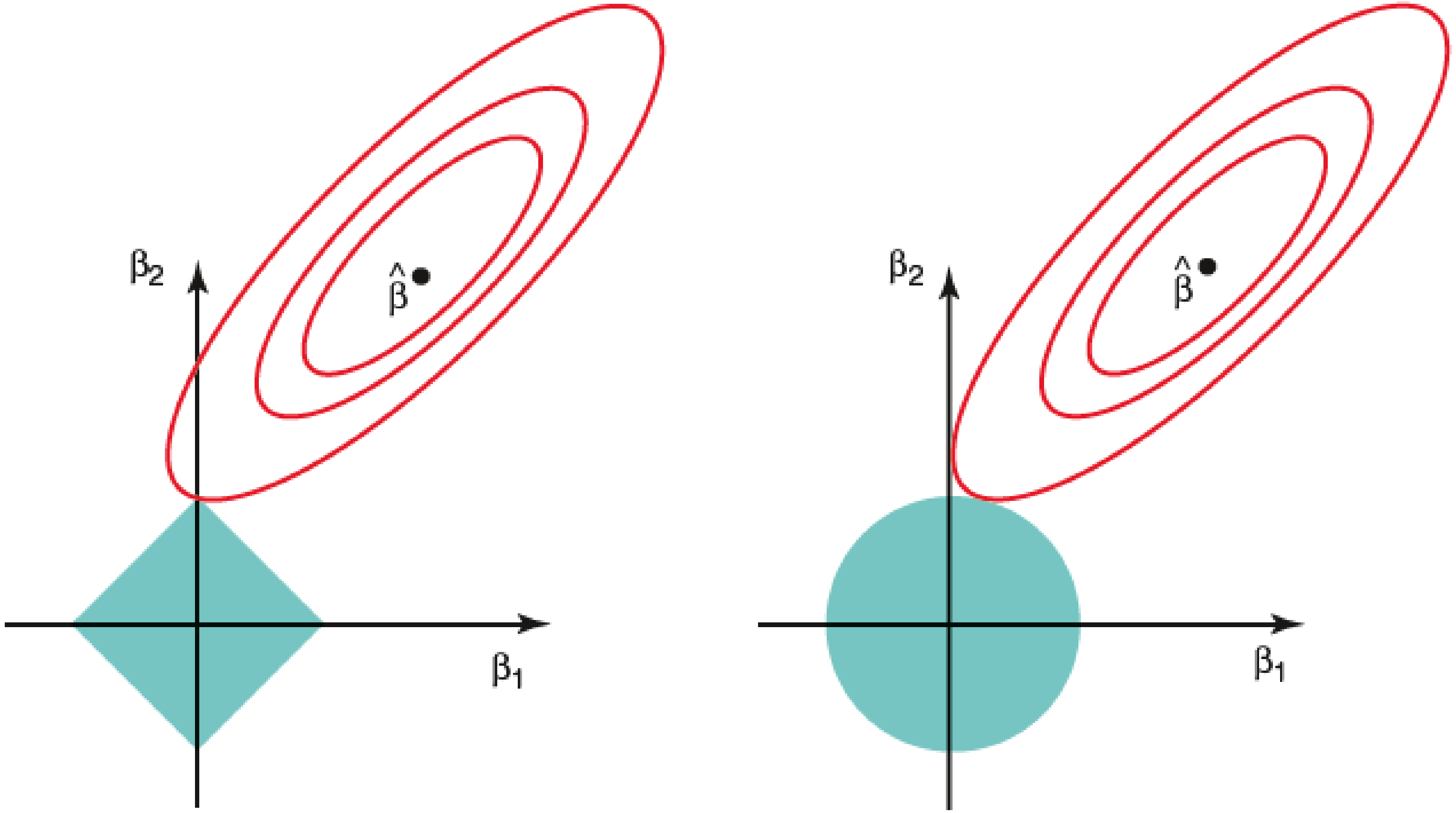
What does the lambda (λ) do?

Lasso can make certain β 0. Why?



Ridge and Lasso

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



Elastic Net

$$\mathcal{L} = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 \right)$$

- Elastic Net is a convex combination of Ridge and Lasso
- Elastic Net > Ridge > Lasso

What features to include?

Method 1. Best subset method

- The idea: test all possible combinations
- Curse of dimensionality!

Method 2. Regularization

- The idea: Penalize unnecessary complexity/features
- Hyperparameter lambda
- Ridge (L2), Lasso (L1), Elastic Net (L1+L2)

TIP: normalize the columns

Method 3. Cross-Validation

Model validation during the training

The general idea:

- Split dataset into Train, Validation, Test
- Train using train data with a hyperparam(s) fixed
- Tune the hyperparameter(s) with validation
- When tuning is done, test with the test data

- How do I know my validation dataset was good or bad?

Cross-Validation