CS/ECE 148–

Data Science Fundamentals

Model Selection and Regularization

UCLA Computer Science

So Far …

Interviewer: What's your biggest strength?

Me: I'm an expert in machine learning.

Inteviewer: What's 6 + 10?

Me: Zero.

Interviewer: Nowhere near, it's 16.

Me: It's 16.

Interviewer: Ok... What's 10 + 20?

Me: It's 16.

So Far …

Model Fitness

How does the model perform predicting?

Comparison of Two Models

How do we choose from two different models?

Evaluating Significance of Predictors

Does the outcome depend on the predictors?

How well do we know $\hat{f}$

The confidence intervals of our $\hat{f}$

Lecture Outline

Overfitting

Model Selection

Cross Validation

Bias vs Variance

Regularization: LASSO and Ridge

Regularization Methods: A Comparison

Multilinear model

In practice, it is unlikely that any response variable Y depends solely on one predictor *x*. Rather, we expect that is a function of multiple predictors $f(X_1, \dots, X_J)$. Using the notation we introduced in last lecture,

$$Y = y_1, \dots, y_n, \quad X = X_1, \dots, X_J \text{ and } X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$$

In this case, we can still assume a simple form for f -a multilinear form:

$$Y = f(X_1, \dots, X_J) + \epsilon = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_J X_J + \epsilon$$

Hence, $\hat{f}$, has the form

$$\hat{Y} = \hat{f}(X_1, \dots, X_J) + \epsilon = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \dots + \hat{\beta}_J X_J + \epsilon$$

Lecture Outline

Overfitting

Model Selection

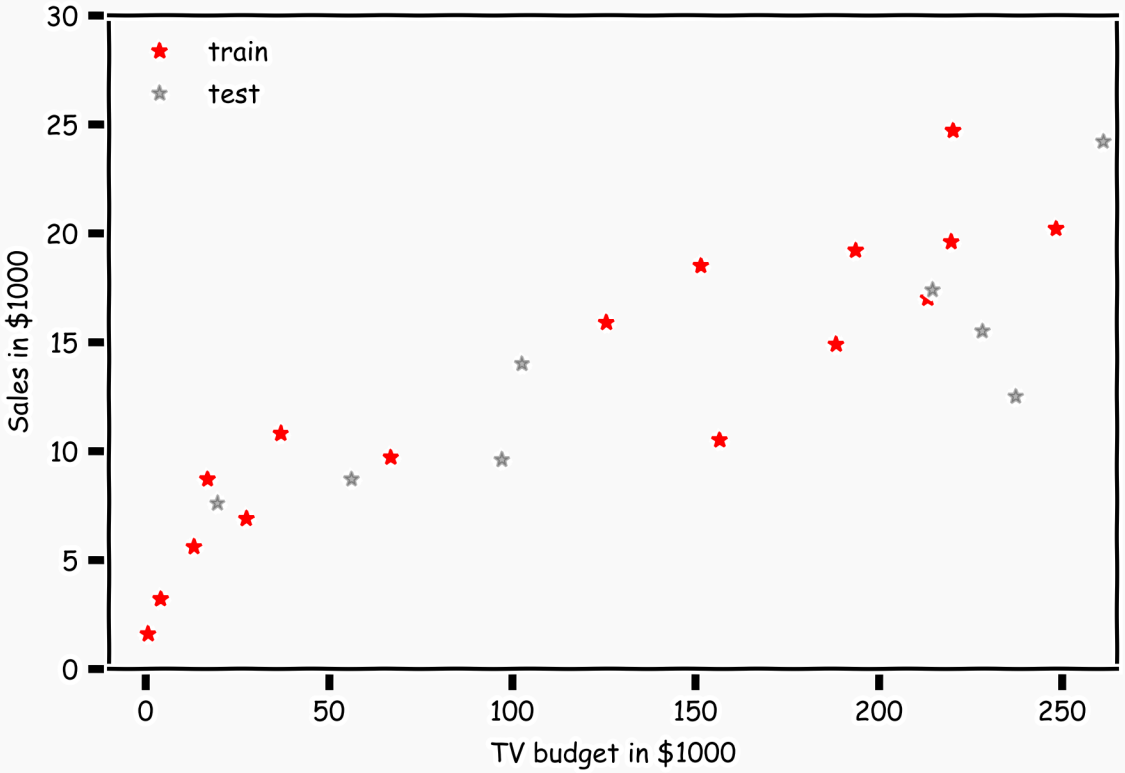
Cross Validation

Bias vs Variance

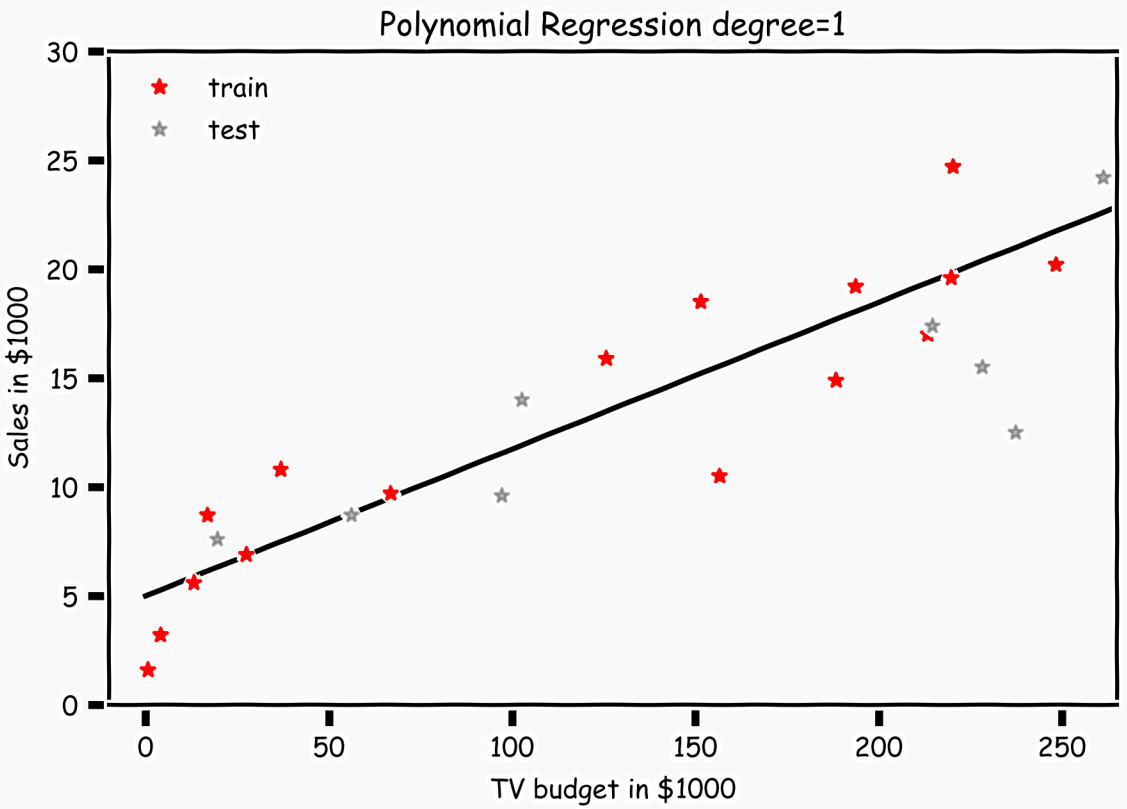
Regularization: LASSO and Ridge

Regularization Methods: A Comparison

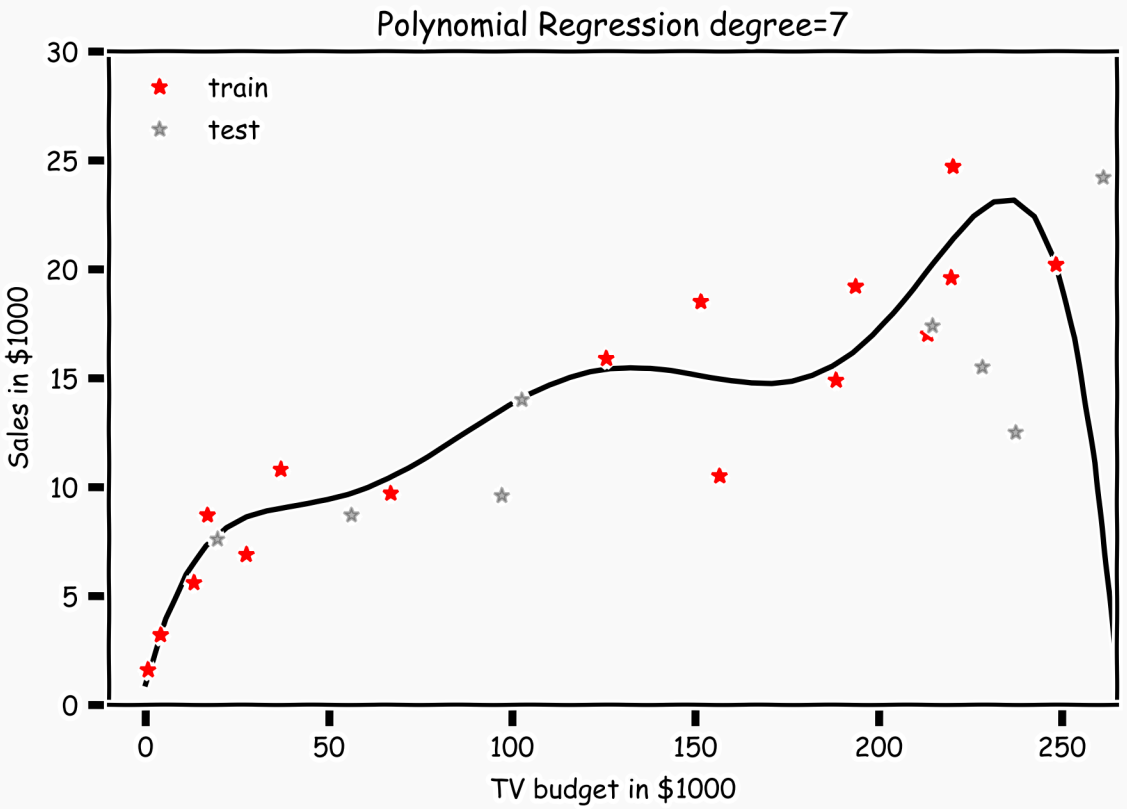
Overfitting



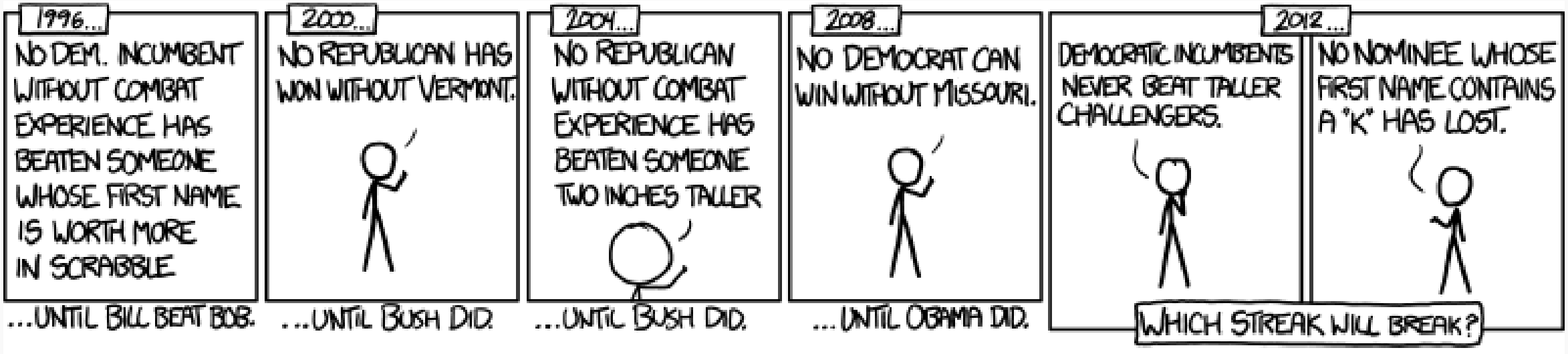
Overfitting



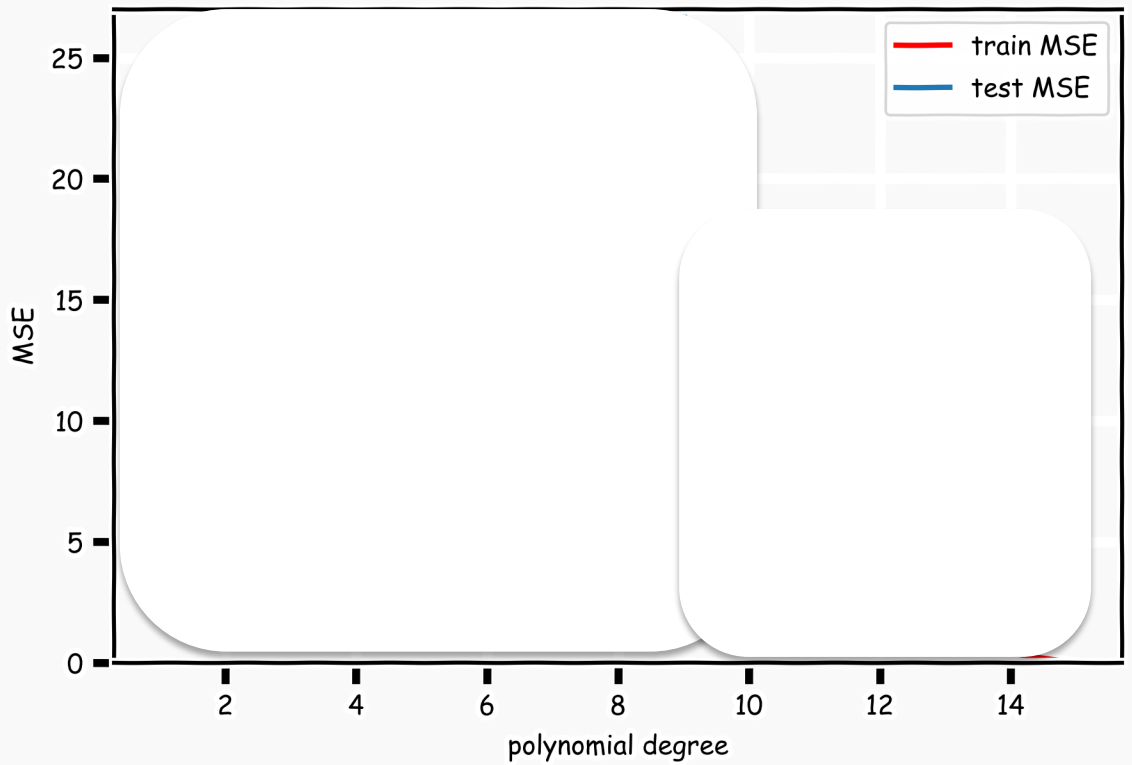
Overfitting



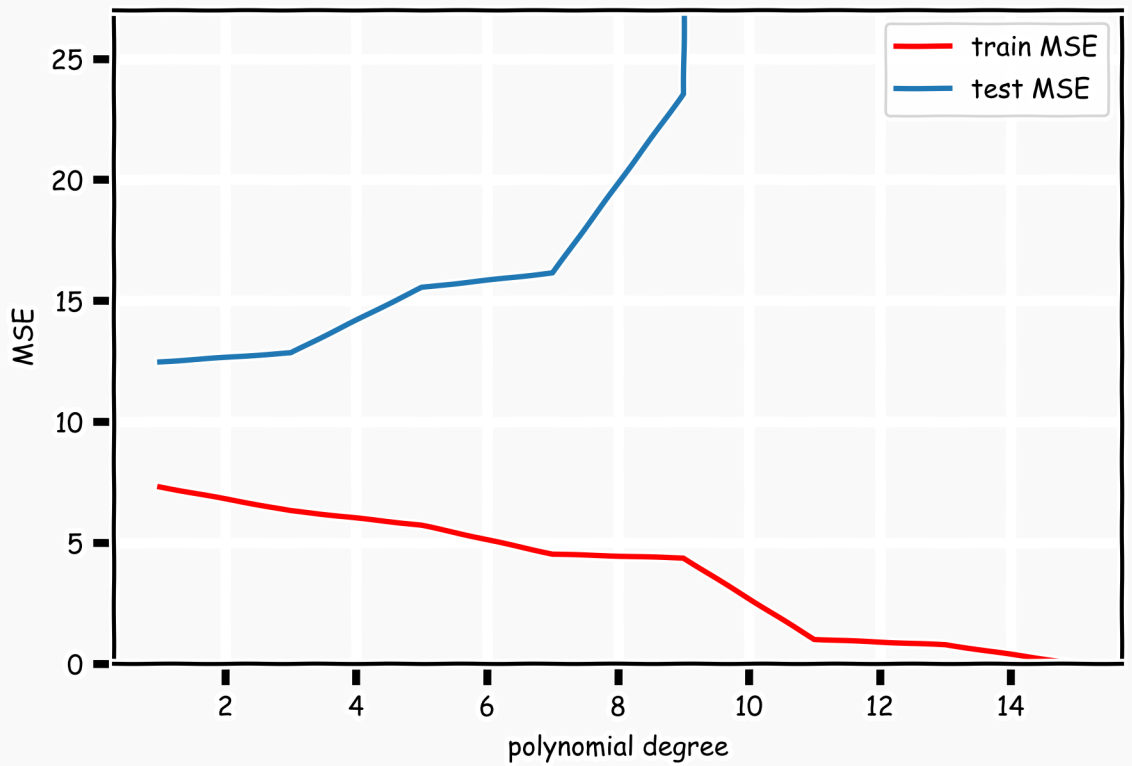
Overfitting



Validation


train MSE
test MSE
MSE
25
20
15
10
5
0
2
4
6
8
10
12
14
polynomial degree

Validation


train MSE
test MSE
MSE
0
5
10
15
20
25
2
4
6
8
10
12
14
polynomial degree

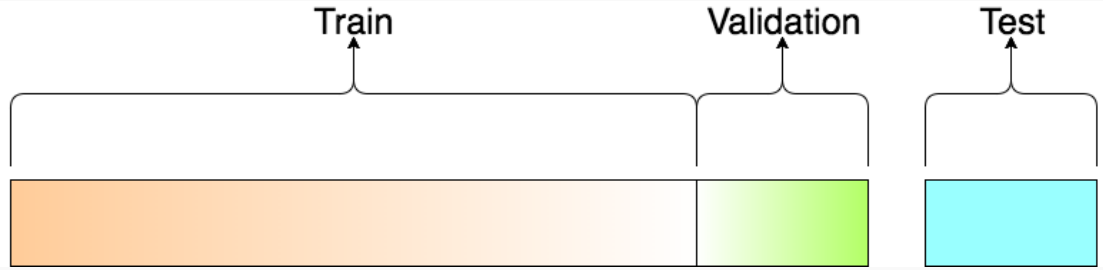
Train-Validation-Test

Question:

How would you report the performance of the model?

R^2_test = 0.52

R^2_train(degree=1) = 0.83



A validation set is part of the training set used for parameter selection as well as for avoiding overfitting of the machine learning model being developed. On the contrary, a test set is meant for **evaluating or testing the performance of a trained model**

Lecture Outline

Overfitting

Model Selection

Cross Validation

Bias vs Variance

Regularization: LASSO and Ridge

Regularization Methods: A Comparison

Model Selection

Model selection is the application of a principled method to determine the complexity of the model, e.g. choosing a subset of predictors, choosing the degree of the polynomial model etc.

A strong motivation for performing model selection is to avoid overfitting, which we saw can happen when:

- there are too many predictors:
  - the feature space has high dimensionality
  - the polynomial degree is too high
  - too many cross terms are considered
- the coefficients values are too **extreme (we have not seen this yet)**

Model Selection

Question:

How many different models when considering J predictors?

Model Selection

Example: 3 predictors (X_1, X_2, X_3)

- Models with 0 predictor:

  M0:

- Models with 1 predictor:

  M1: X_1

  M2: X_2

  M3: X_3

- Models with 2 predictors:

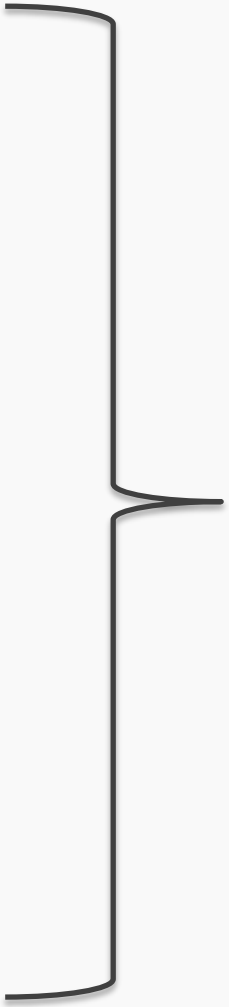
  M4: $\{X_1, X_2\}$

  M5: $\{X_2, X_3\}$

  M6: $\{X_3, X_1\}$

- Models with 3 predictors:

  M7: $\{X_1, X_2, X_3\}$



2^J Models

Stepwise Variable Selection and Cross Validation

Selecting (near)optimal subsets of predictors (including choosing the degree of polynomial models) through:

- stepwise variable selection - iteratively building an optimal subset of predictors by optimizing a fixed model evaluation metric each time,
- validation - selecting an optimal model by evaluating each model on validation set.

(We will also address the issue of discouraging extreme values in model parameters later.)

Stepwise Variable Selection: Forward method

In **forward selection**, we find an 'optimal' set of predictors by iterative building up our set.

1. Start with the empty set P0, construct the null model M0.

2. For $k = 1, \dots, J$:

2.1 Let M_{k-1} be the model constructed from the best set of $k-1$ predictors, P_{k-1}.

2.2 Select the predictor X_{n_k}, not in P_{k-1}, so that the model constructed from $P_k = X_{n_k} \cup P_{k-1}$ optimizes a fixed metric (this can be p -value, F -stat; validation MSE, R^2, or AIC/BIC on training set).

2.3 Let M_k denote the model constructed from the optimal P_k.

3. Select the model M amongst $\{M_0, M_1, \dots, M_J\}$ that optimizes a fixed metric (this can be validation MSE, R^2)

Stepwise Variable Selection: Forward method

Predict height based on Weight, Gender, Favorite number

Stepwise Variable Selection Computational Complexity

How many models did we evaluate?

- 1st step, ***J* Models**
- 2nd step, *J-1* **Models** (add 1 predictor out of *J-1* possible)
- 3rd step, *J-2* **Models** (add 1 predictor out of *J-2* possible)
- …

$$O(J^2) \ll 2^J \text{ for large } J$$

Lecture Outline

Overfitting

Model Selection

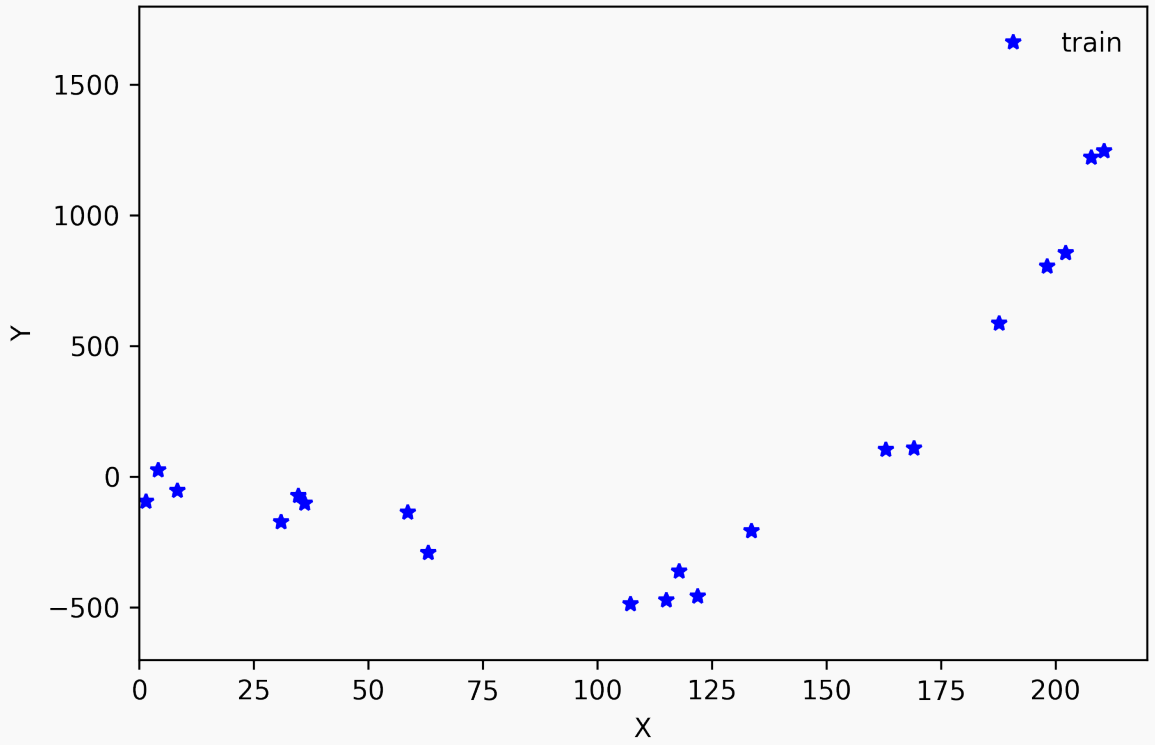
Cross Validation

Bias vs Variance

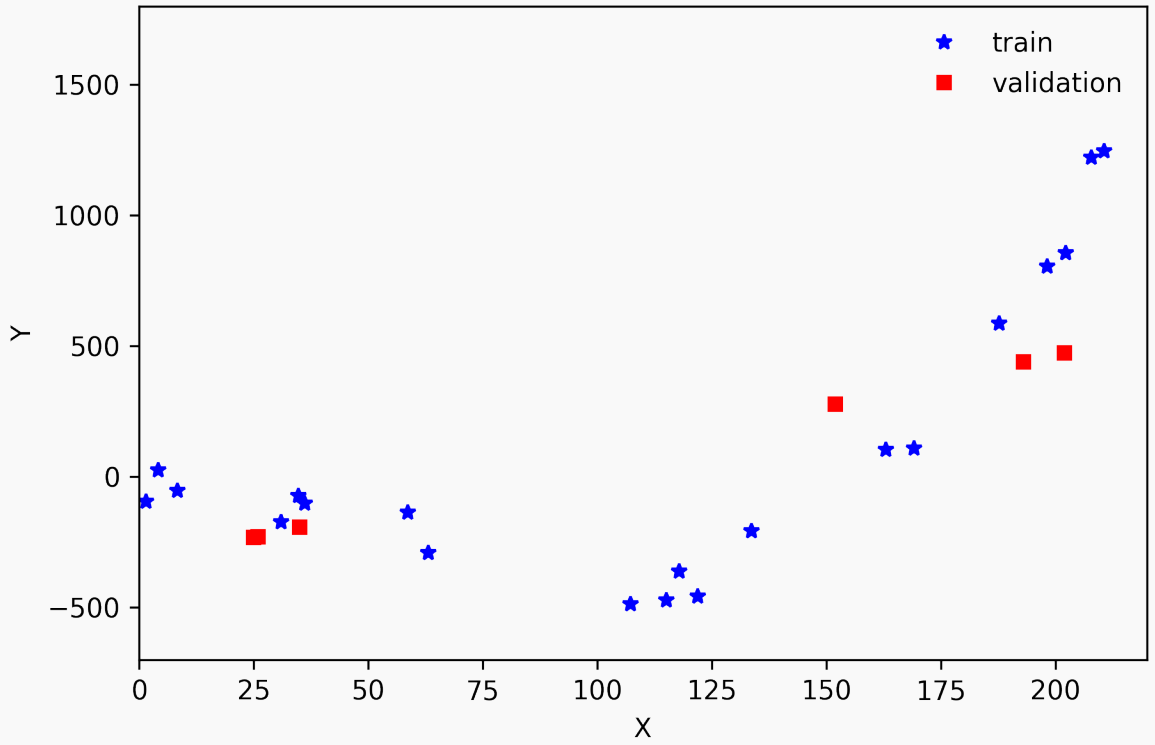
Regularization: LASSO and Ridge

Regularization Methods: A Comparison

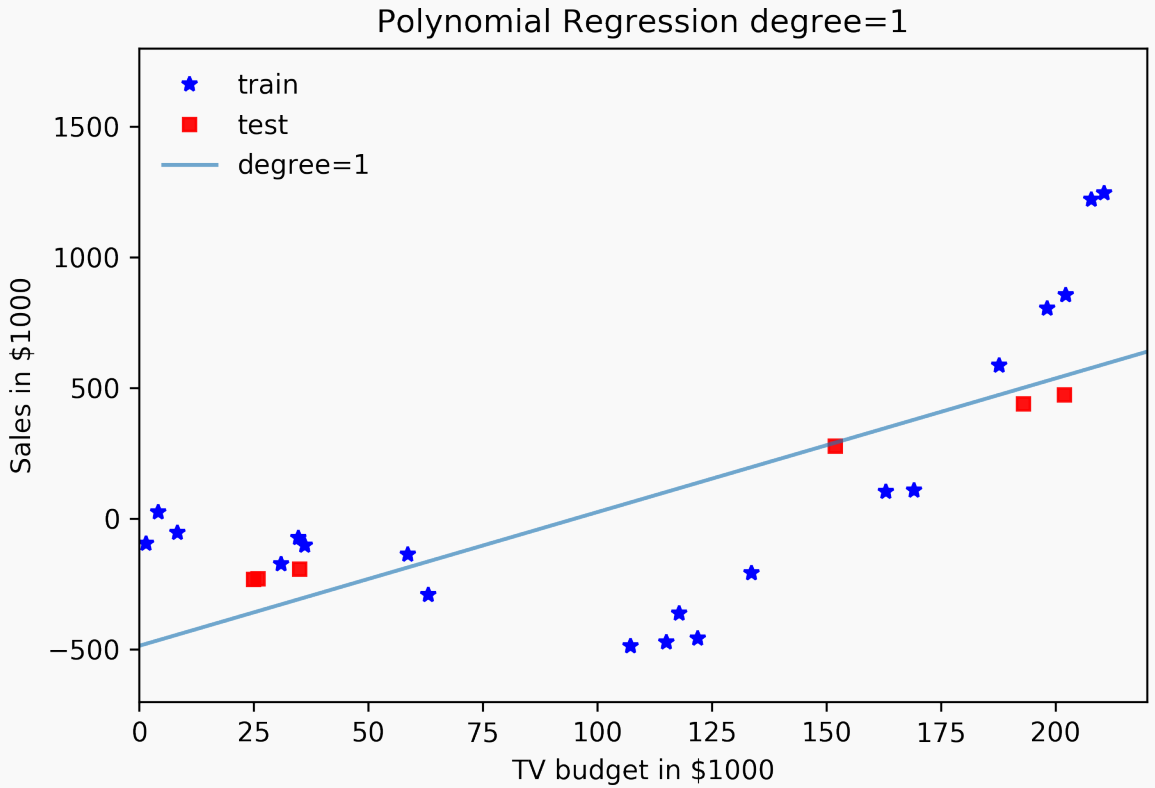
Cross Validation



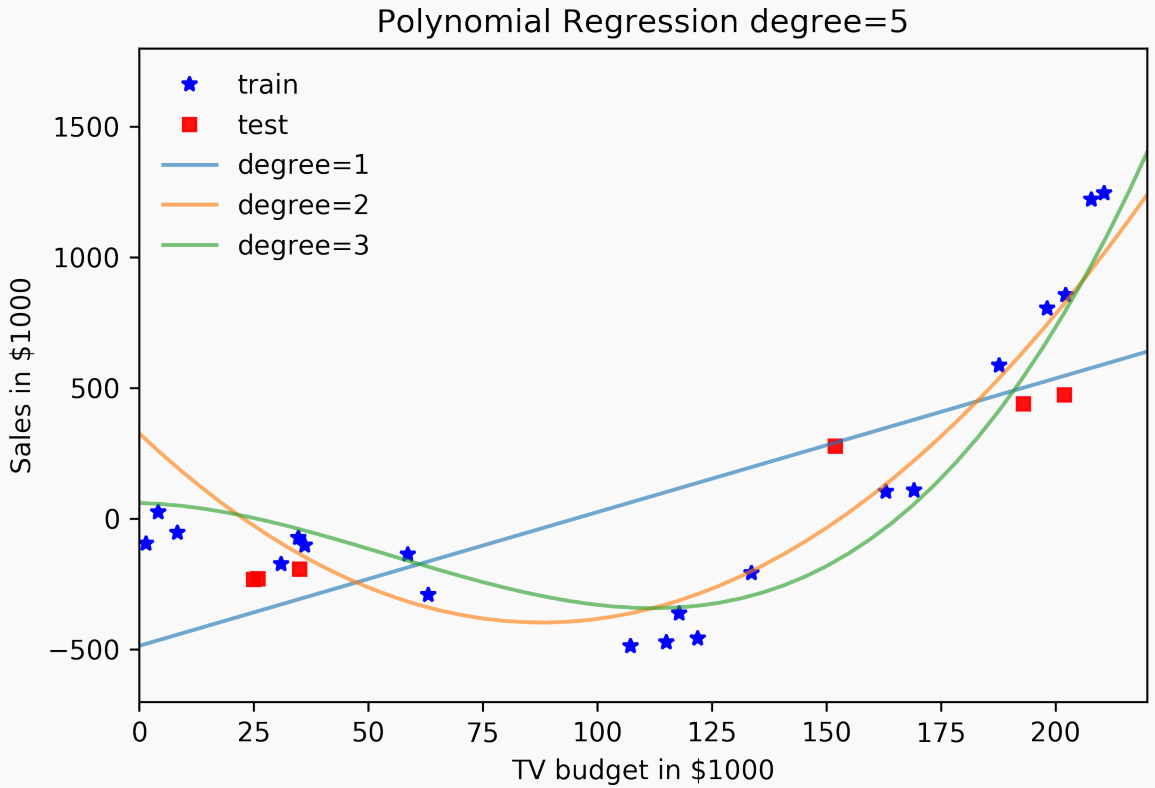
Cross Validation



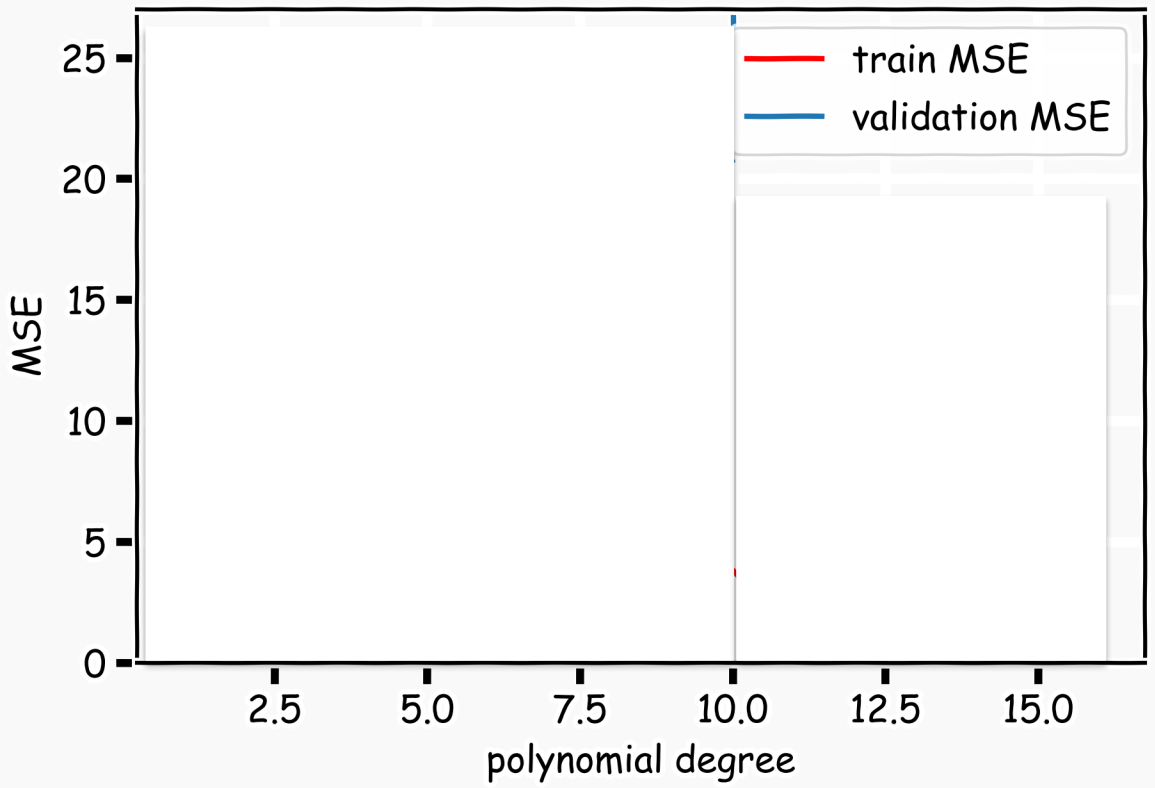
Cross Validation



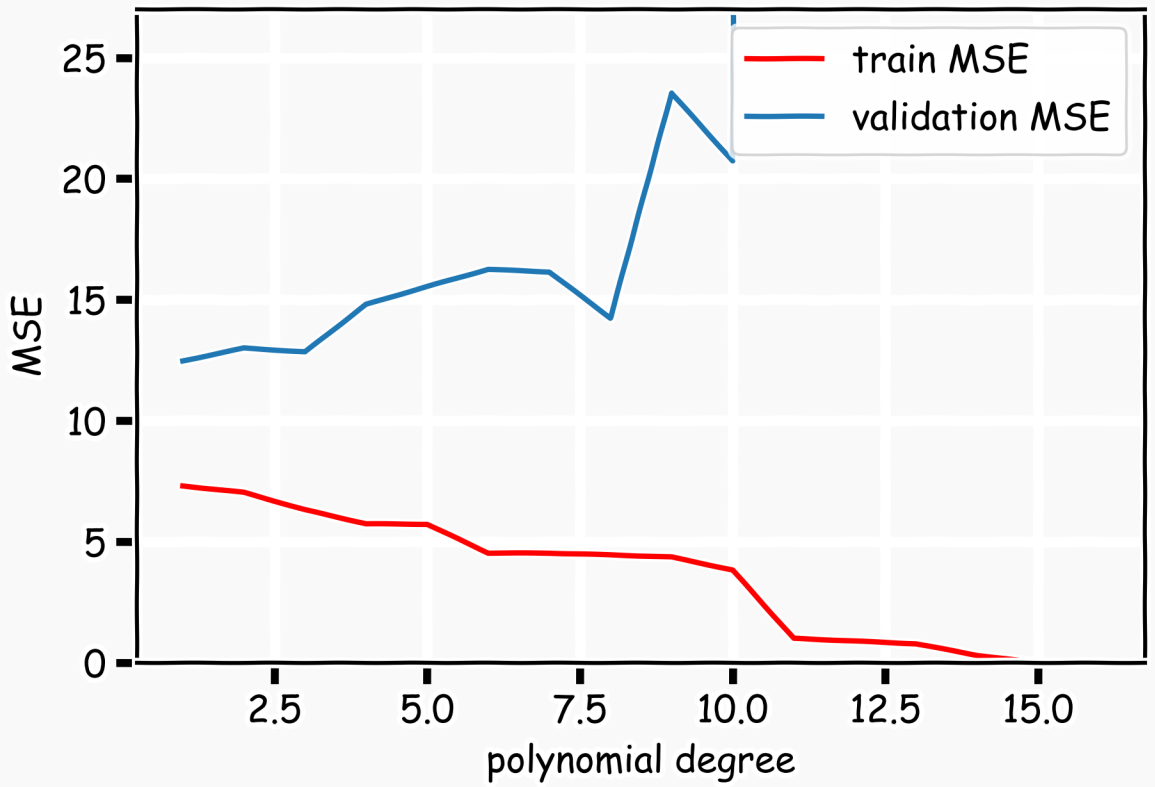
Cross Validation



Validation


train MSE
validation MSE
MSE
25
20
15
10
5
0
2.5
5.0
7.5
10.0
12.5
15.0
polynomial degree

Validation


train MSE
validation MSE
MSE
25
20
15
10
5
0
2.5
5.0
7.5
10.0
12.5
15.0
polynomial degree

Cross Validation: Motivation

Using a single validation set to select amongst multiple models can be problematic - **there is the possibility of overfitting to the validation set**.

One solution to the problems raised by using a single validation set is to evaluate each model on **multiple** validation sets and average the validation performance.

One can randomly split the training set into training and validation multiple times **but** randomly creating these sets can create the scenario where important features of the data never appear in our random draws.

Cross Validation

K-Fold Cross Validation

Given a data set $\{X_1, \dots, X_n\}$, where each $\{X_1, \dots, X_n\}$ contains *J* features.

To ensure that every observation in the dataset is included in at least one training set and at least one validation set we use the **K-fold validation**:

- split the data into *K* uniformly sized chunks, $\{C_1, \dots, C_K\}$
- we create *K* number of training/validation splits, using one of the *K* chunks for validation and the rest for training.

We fit the model on each training set, denoted $\hat{f}_{C_{-i}}$, and evaluate it on the corresponding validation set, $\hat{f}_{C_{-i}}(C_i)$. The ***cross validation is the performance*** of the model averaged across all validation sets:

$$CV(\text{Model}) = \frac{1}{K}\sum_{i=1}^{K} L(\hat{f}_{C_{-i}}(C_i))$$

where *L* is a loss function.

Leave-One-Out

*Or using the **leave one out*** method:

- validation set: $\{X_i\}$
- training set: $X_{-i} = \{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$

for $i = 1, \dots, n$:

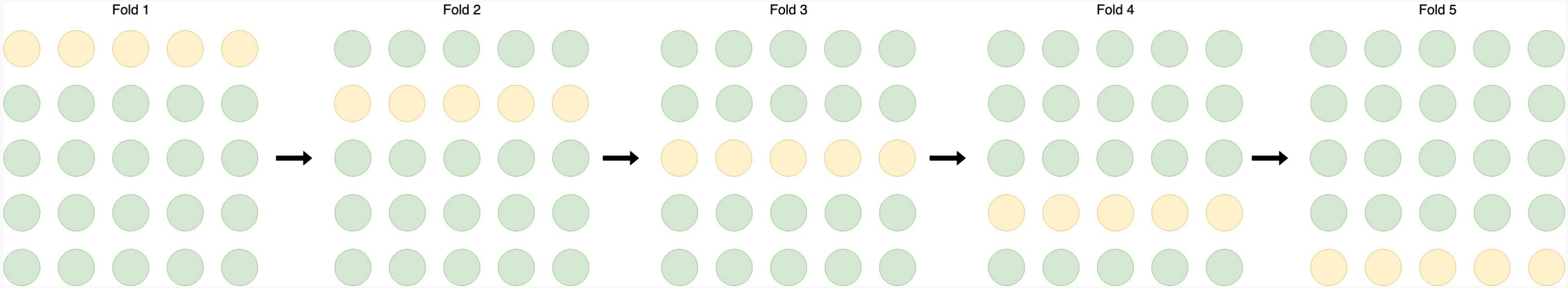
We fit the model on each training set, denoted $\hat{f}_{X_{-i}}$, and evaluate it on the corresponding validation set, $\hat{f}_{X_{-i}}(X_i)$.

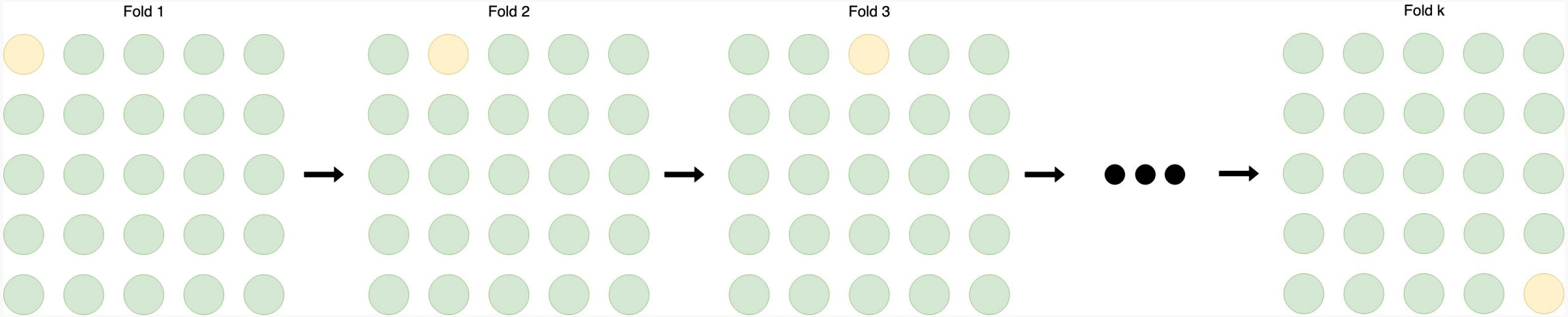
The ***cross validation score*** is the performance of the model averaged across all validation sets:

$$CV(\text{Model}) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{f}_{X_{-i}}(X_i))$$

where *L* is a loss function.

Cross Validation





Leave-one-out validation is a special type of cross-validation where N=k

Predictor Selection: Cross Validation

Question: What is the right ratio of train/validate/test, how do I choose K?

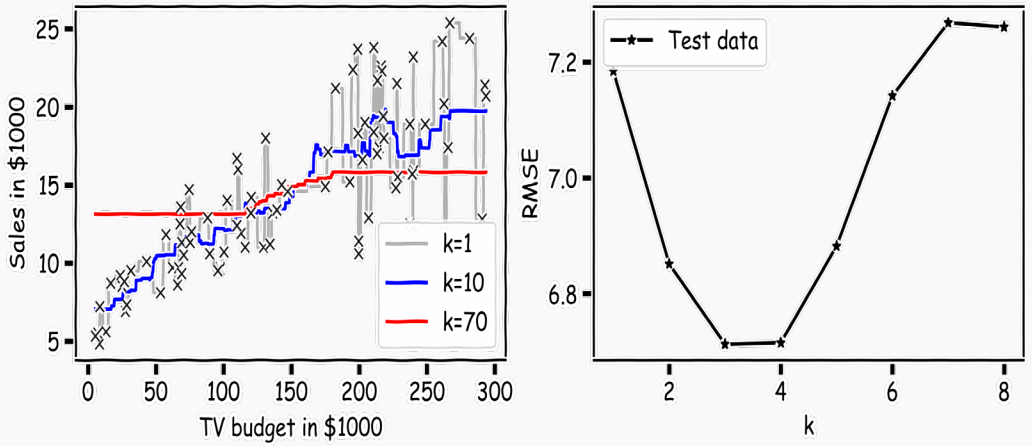
Question: What is the difference in multiple predictors and polynomial regression in model selection?

We can frame the problem of degree selection for polynomial models as a predictor selection problem:

which of the predictors $\{x, x^2, \dots, x^m\}$, should we select for modeling?

kNN Revisited

Recall our first simple, intuitive, non-parametric model for regression – the kNN model. We saw that it is vitally important to select an appropriate *k* for the data.



If the *k* is too small then the model is very sensitive to noise (since a new prediction is based on very few observed neighbors), and if the *k* is too large, the model tends towards making constant predictions.

A principled way to choose *k* is **through K-fold cross validation.**

Lecture Outline

Overfitting

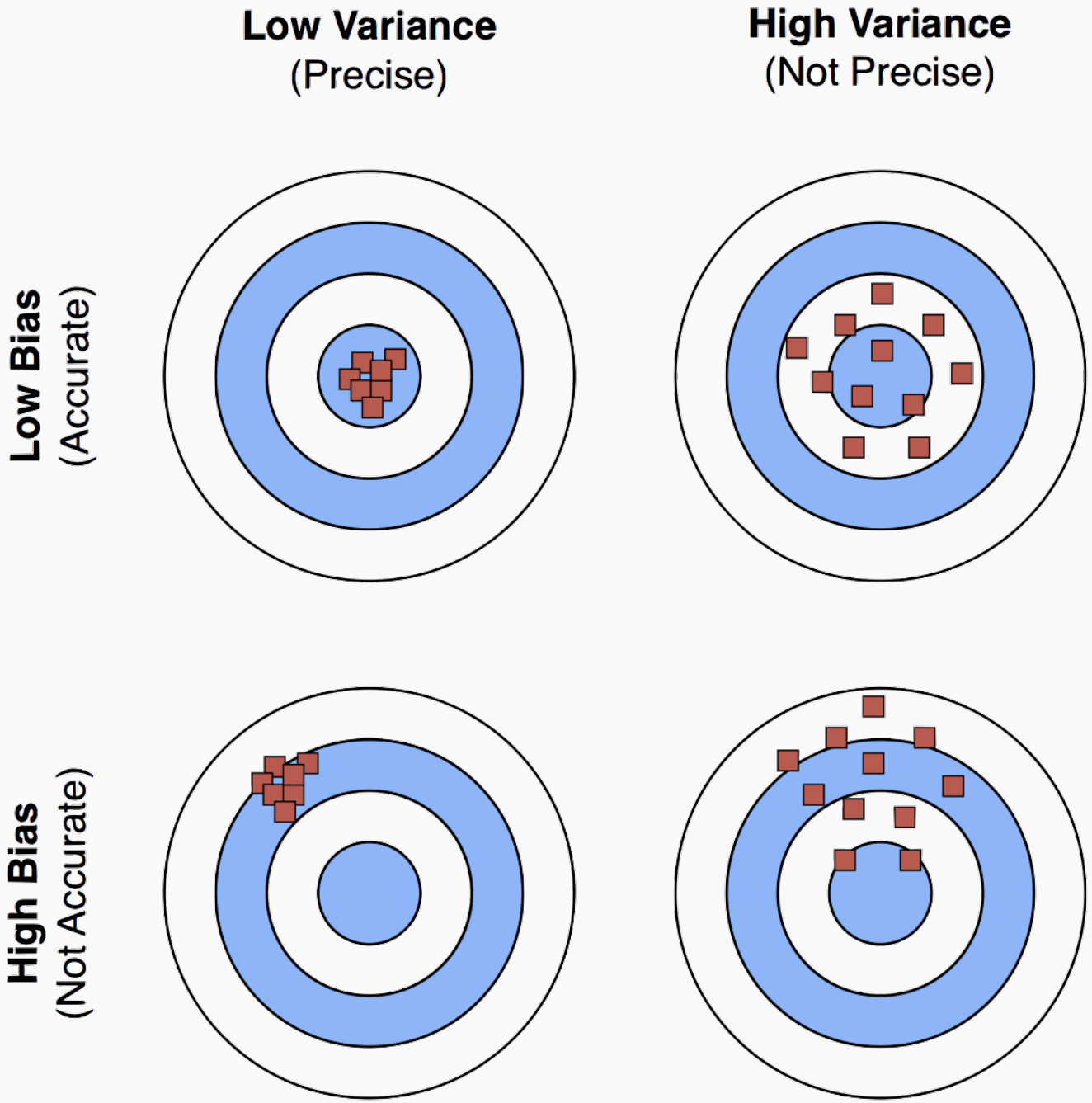
Model Selection

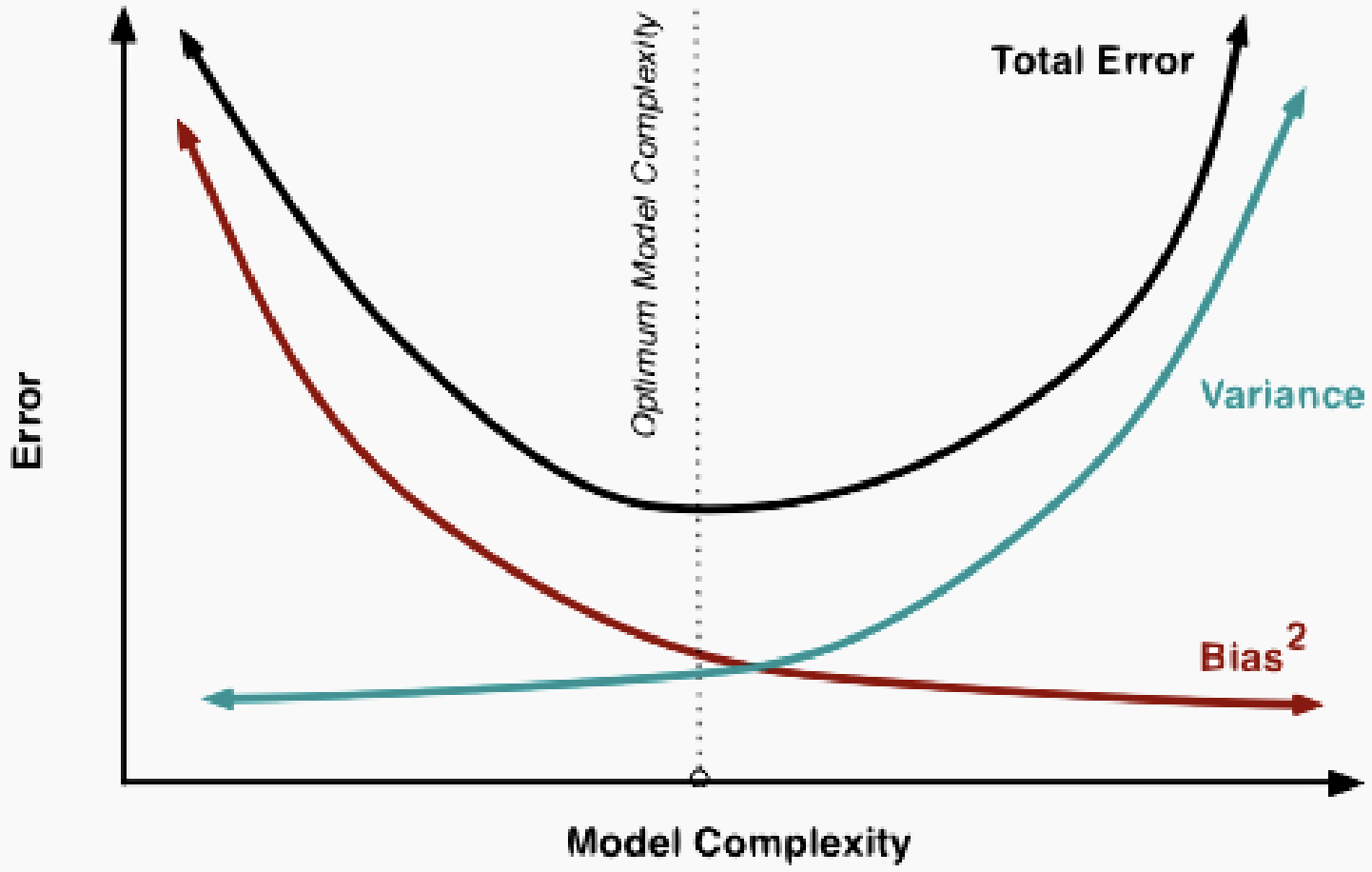
Cross Validation

Bias vs Variance

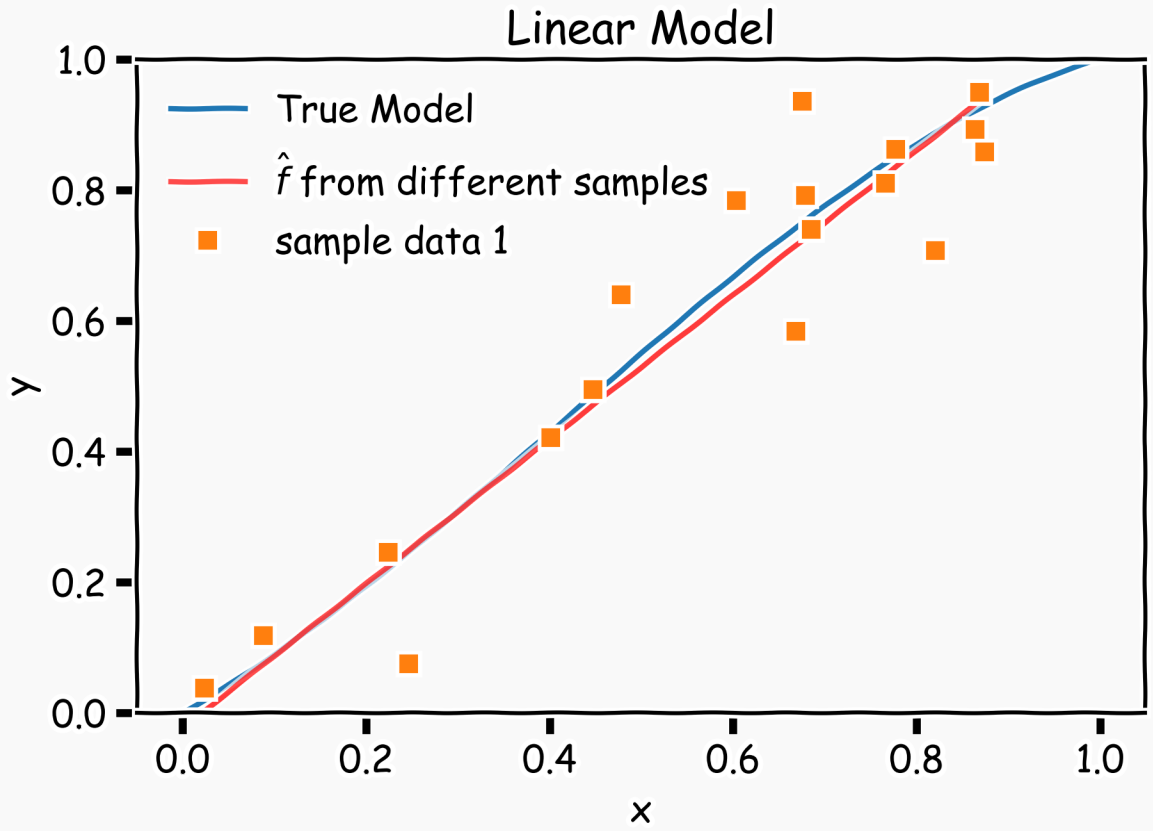
Regularization: LASSO and Ridge

Regularization Methods: A Comparison

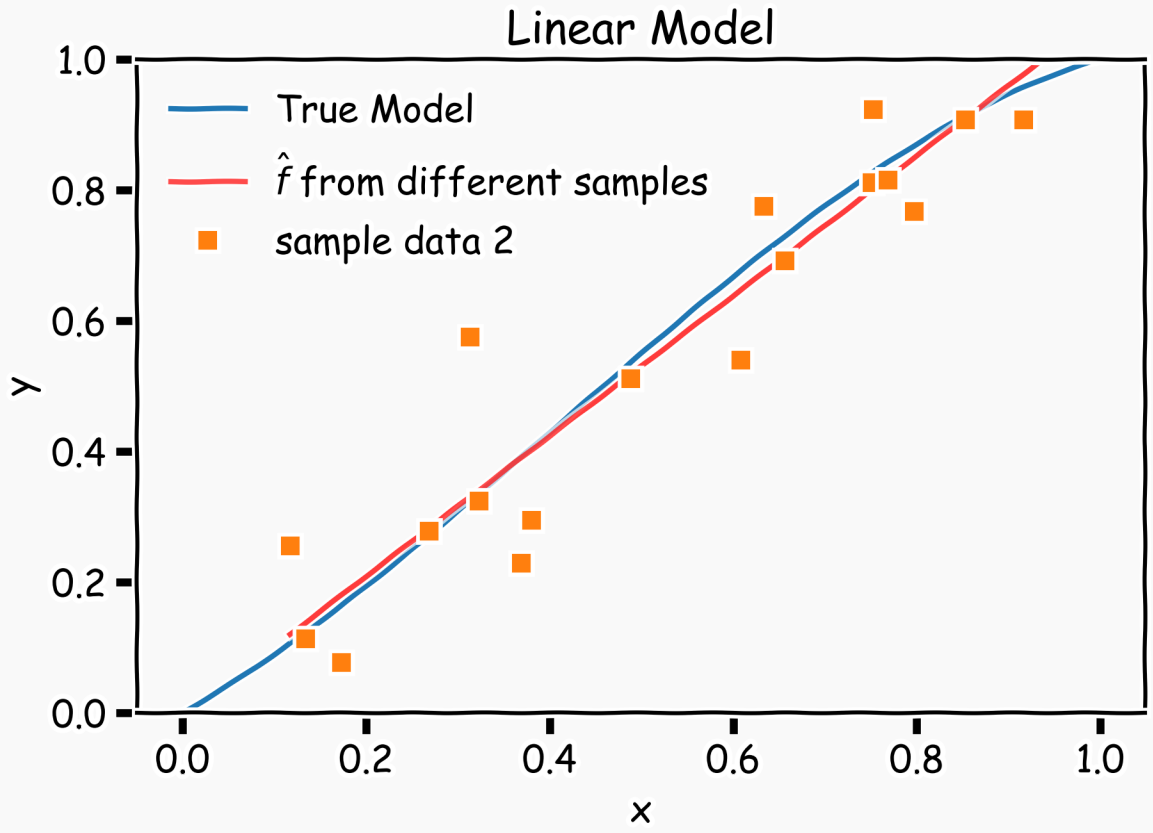

Low Variance
(Precise)
High Variance
(Not Precise)
Low Bias
(Accurate)
High Bias
(Not Accurate)


Error
Optimum Model Complexity
Total Error
Variance
Bias^2
Model Complexity

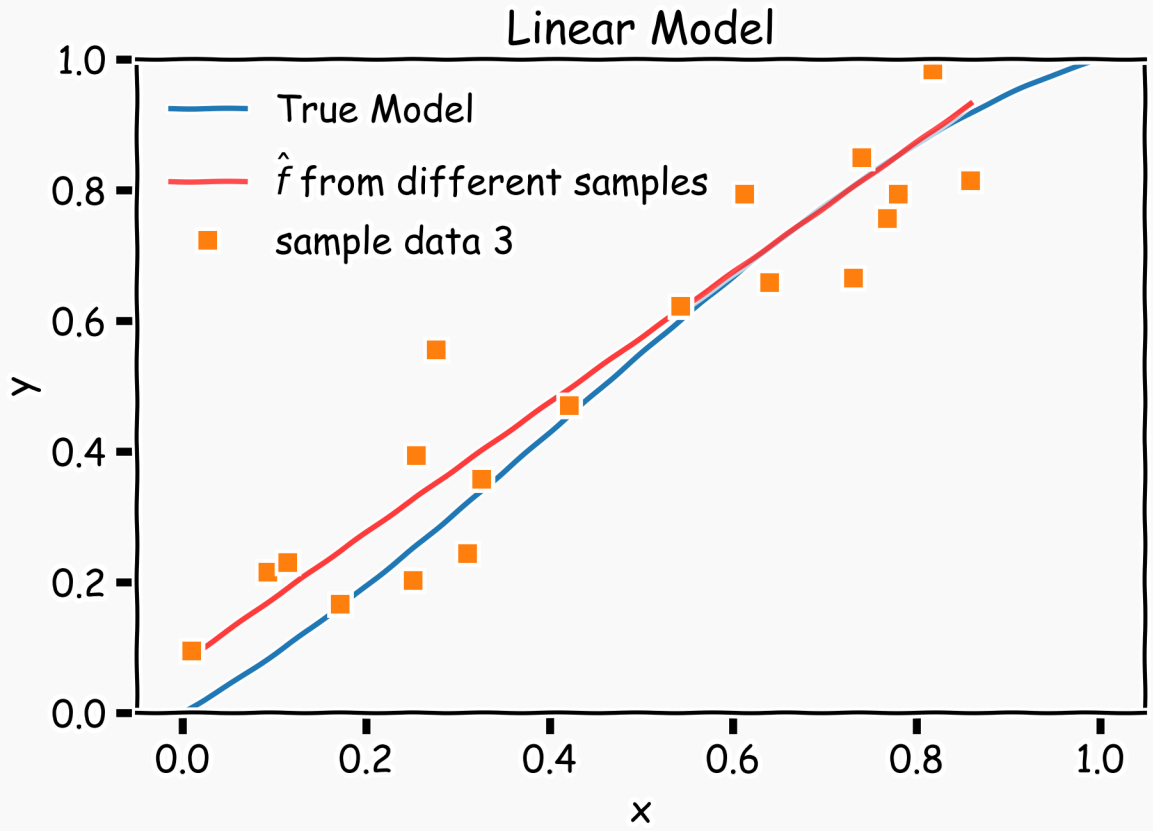
Bias vs Variance



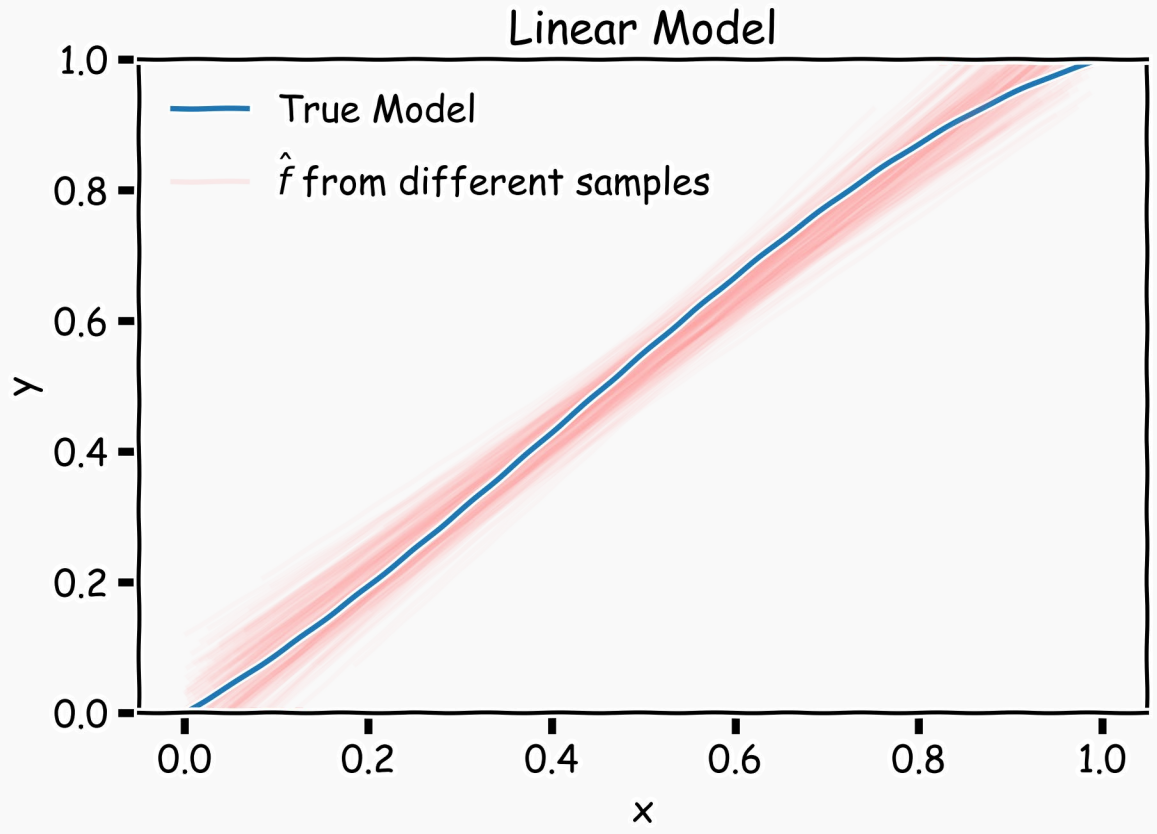
Bias vs Variance



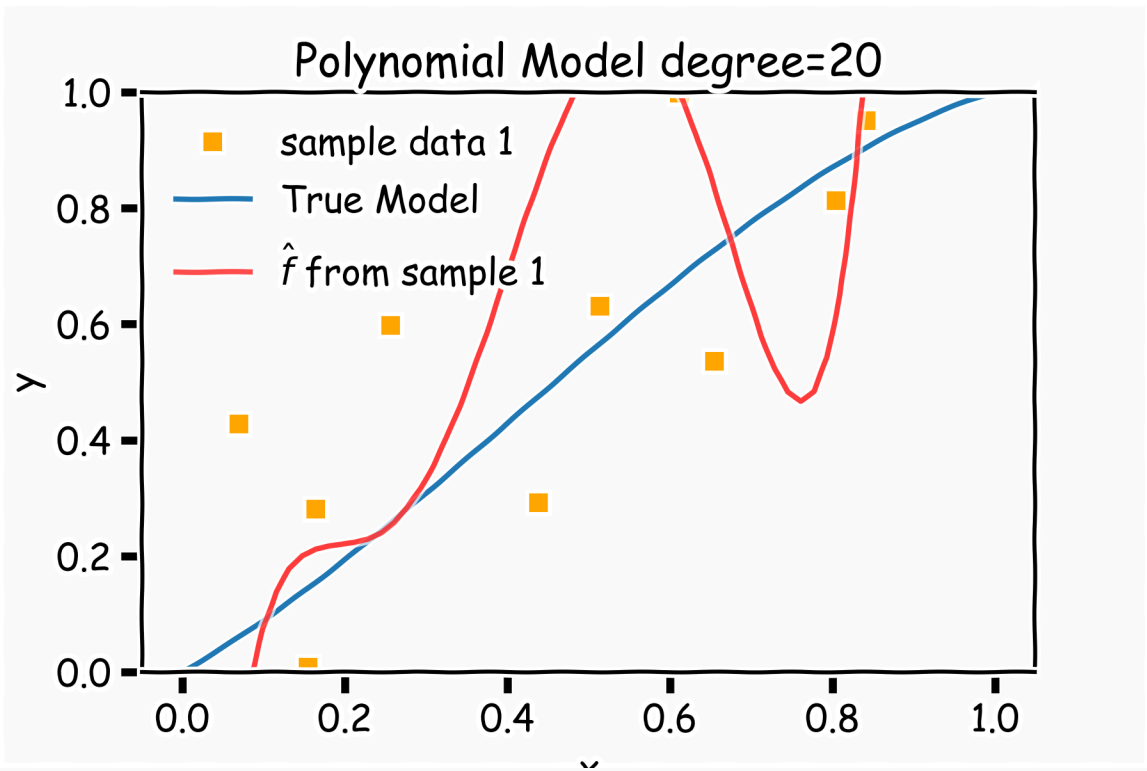
Bias vs Variance



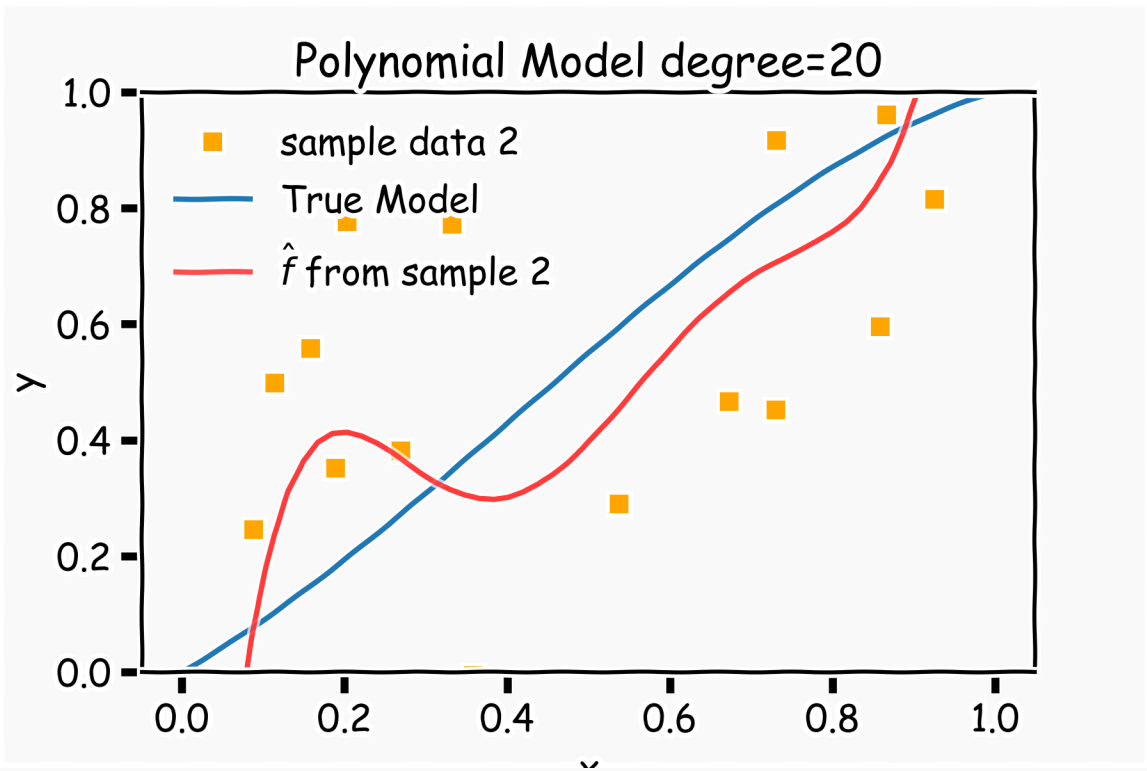
Linear models: 20 data points per line 2000 simulations



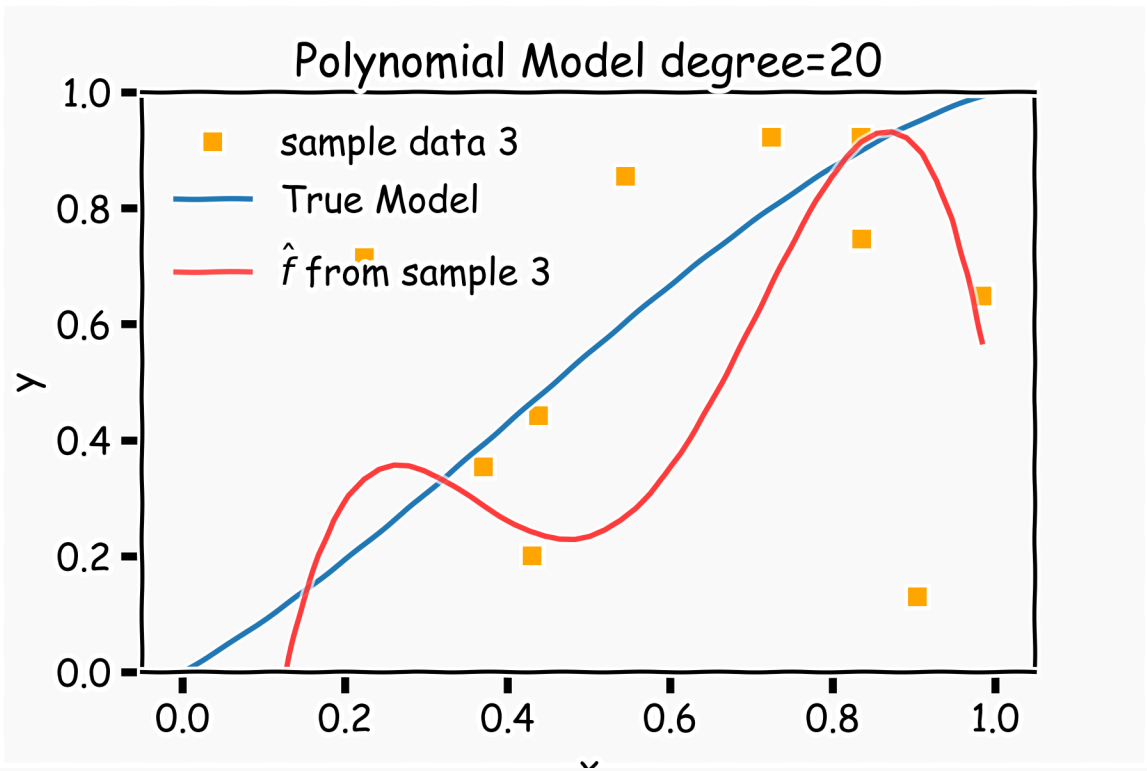
Bias vs Variance



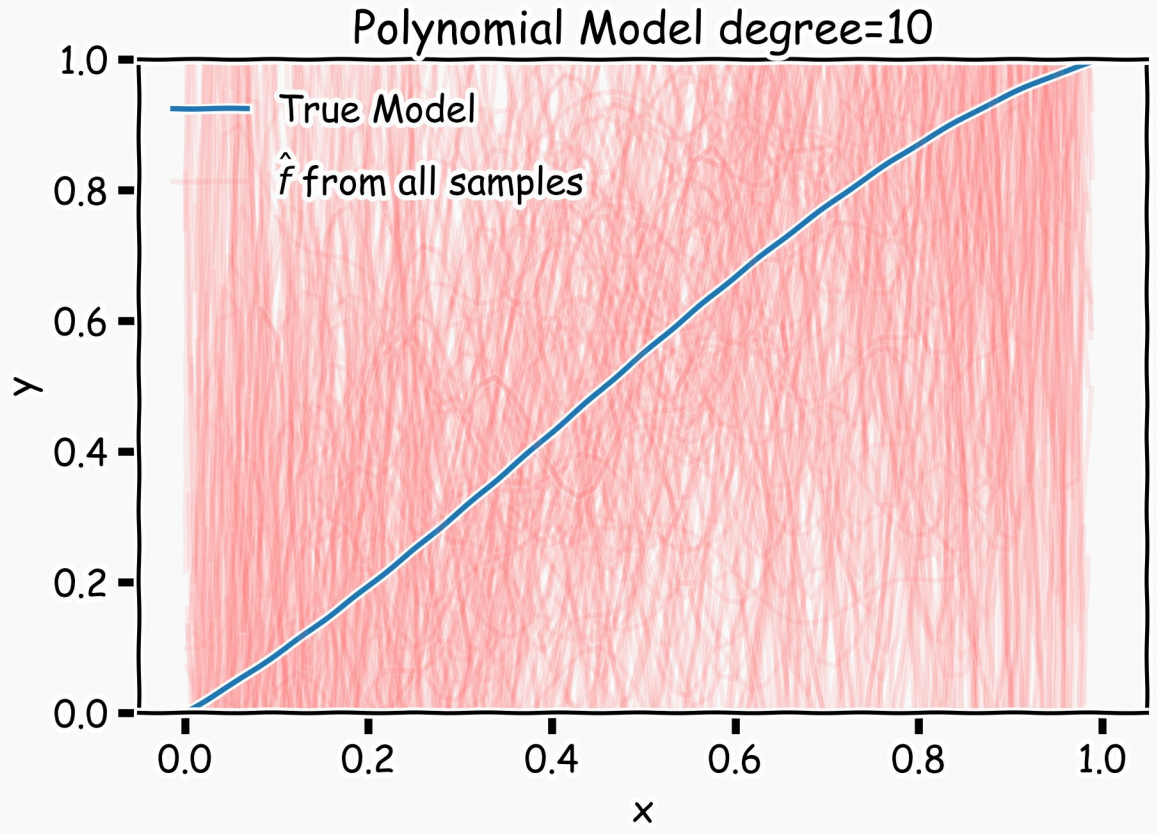
Bias vs Variance



Bias vs Variance



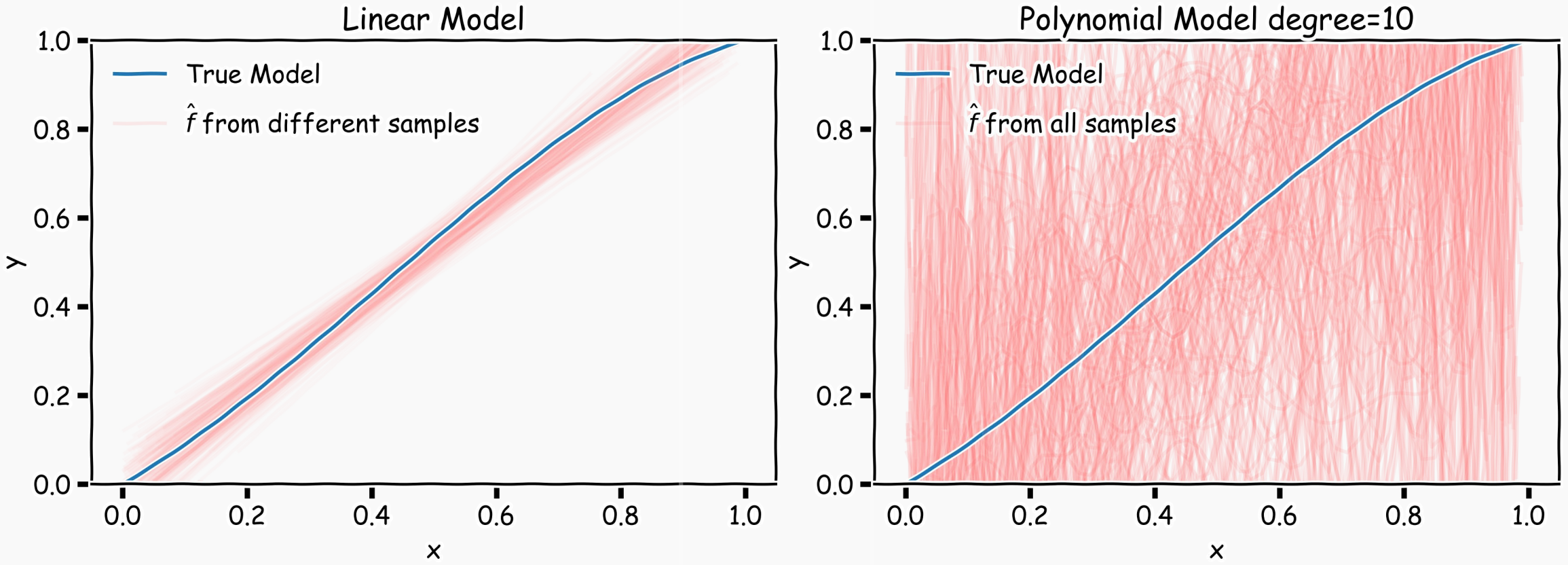
Poly 10 degree models : 20 data points per line 2000 simulations



Bias vs Variance

Left: 2000 best fit straight lines, each fitted on a different 20 point training set.

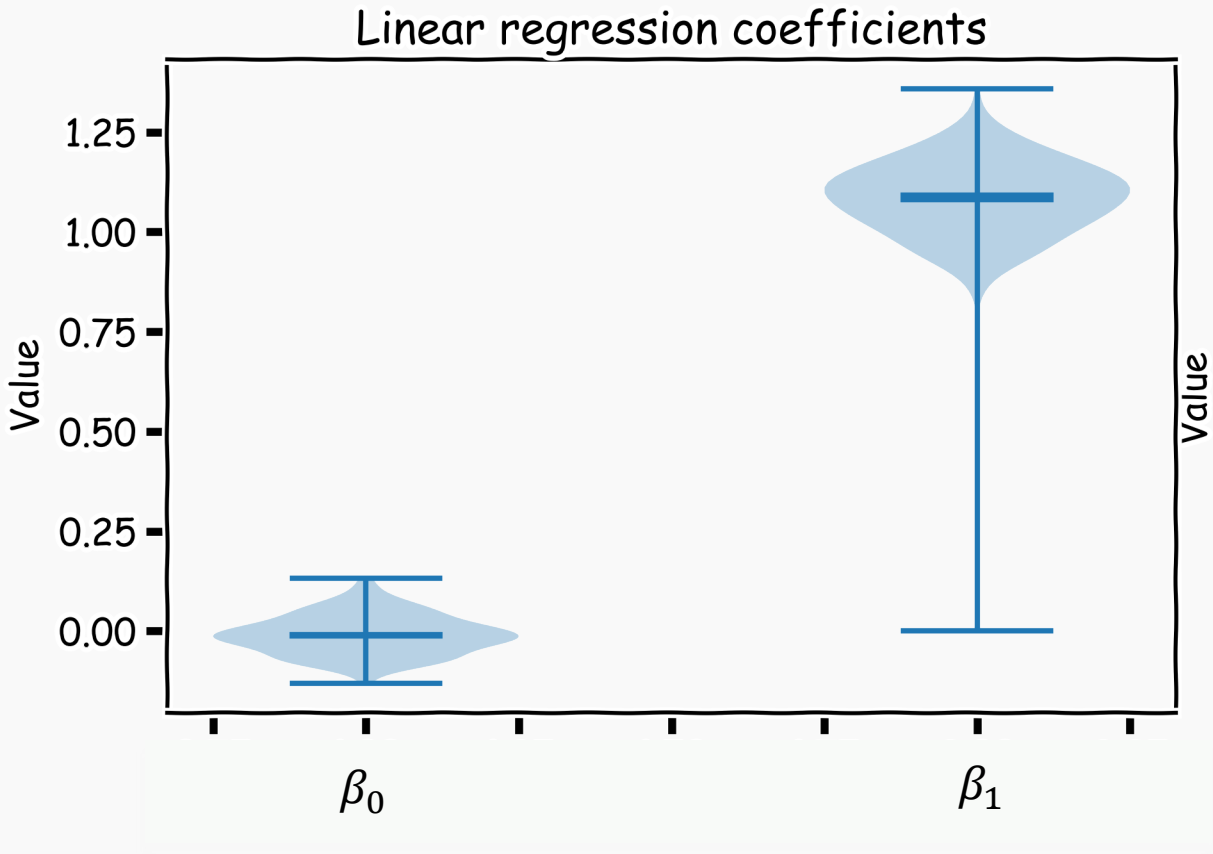
Right: Best-fit models using degree 10 polynomial



Bias vs Variance

Left: Linear regression coefficients

Right: Poly regression of order 10 coefficients



Lecture Outline

Overfitting

Model Selection

Cross Validation

Bias vs Variance

Regularization: LASSO and Ridge

Regularization Methods: A Comparison

Regularization: An Overview

The idea of regularization revolves around modifying the loss function L; in particular, we add a regularization term that penalizes some specified properties of the model parameters

$$L_{reg}(\beta) = L(\beta) + \lambda R(\beta),$$

where λ is a scalar that gives the weight (or importance) of the regularization term.

Fitting the model using the modified loss function L_{reg} would result in model parameters with desirable properties (specified by R).

LASSO Regression

Since we wish to discourage extreme values in model parameter, we need to choose a regularization term that penalizes parameter magnitudes. For our loss function, we will again use MSE.

Together our regularized loss function is:

$$L_{LASSO}(\beta) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i|^2 + \lambda \sum_{j=1}^{J}|\beta_j|.$$

Note that $\sum_{j=1}^{J}|\beta_j|$ is the $\boldsymbol{l_1}$ norm of the vector $\boldsymbol{\beta}$

$$\sum_{j=1}^{J}|\beta_j| = \|\boldsymbol{\beta}\|_1$$

Ridge Regression

Alternatively, we can choose a regularization term that penalizes the squares of the parameter magnitudes. Then, our regularized loss function is:

$$L_{Ridge}(\beta) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i|^2 + \lambda \sum_{j=1}^{J} \beta_j^2.$$

Note that $\sum_{j=1}^{J} |\beta_j|^2$ is the $\boldsymbol{l_2}$ norm of the vector $\boldsymbol{\beta}$

$$\sum_{j=1}^{J} \beta_j^2 = \|\boldsymbol{\beta}\|_2^2$$

L1 vs L2

Choosing λ

In both ridge and LASSO regression, we see that the larger our choice of the **regularization parameter** λ, the more heavily we penalize large values in β,

- If λ is close to zero, we recover the MSE, i.e. ridge and LASSO regression is just ordinary regression.
- If λ is sufficiently large, the MSE term in the regularized loss function will be insignificant and the regularization term will force β_{ridge} and β_{LASSO} to be close to zero.

To avoid ad-hoc choices, we should select λ using cross-validation.

Ridge - Computational complexity

Solution to ridge regression:

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

The solution of the Ridge/Lasso regression involves three steps

- Select λ
- Find the minimum of the ridge/Lasso regression cost function (using linear algebra) as with the multiple regression and record the R^2 **on the test set**.
- Find the λ that gives the largest R^2

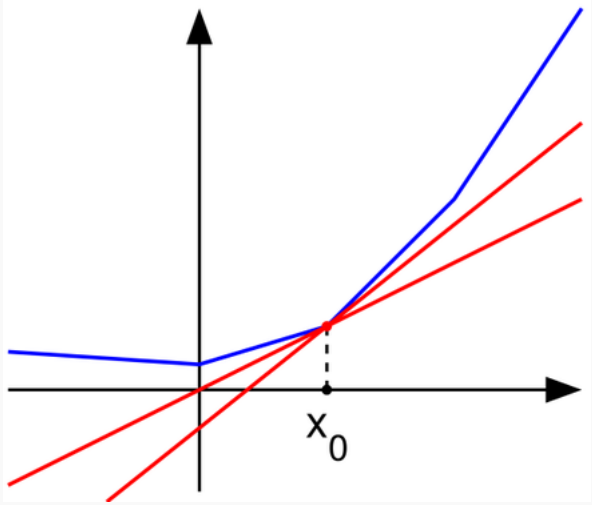
Ridge, LASSO - Computational complexity

Solution to ridge regression:

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

The solution to the LASSO regression:

LASSO has no conventional analytical solution, as the L1 norm has no derivative at 0. We can, however, use the concept of subdifferential or subgradient to find a manageable expression.

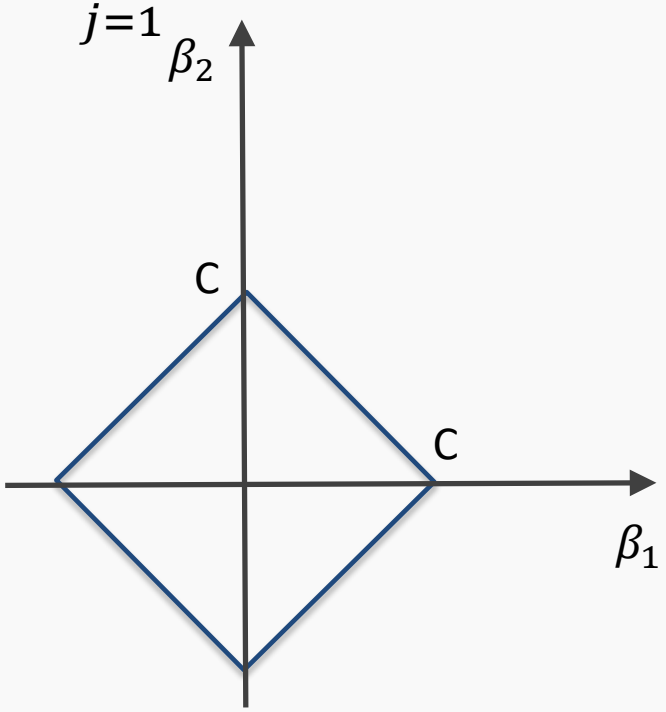


The Geometry of Regularization (LASSO)

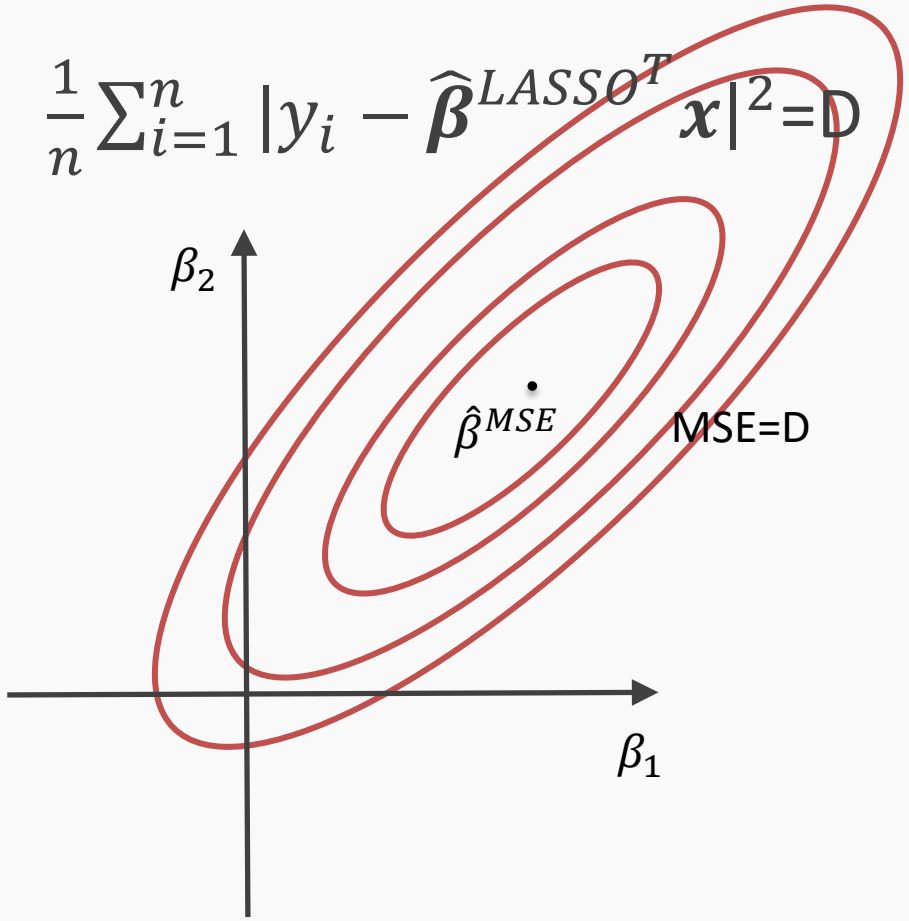
$$L_{LASSO}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T \boldsymbol{x}|^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

$$\widehat{\boldsymbol{\beta}}^{LASSO} = \operatorname{argmin} L_{LASSO}(\boldsymbol{\beta})$$

$$\lambda \sum_{j=1}^{J} |\hat{\beta}_j^{LASSO}| = C$$



$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{\boldsymbol{\beta}}^{LASSO^T} \boldsymbol{x}|^2 = D$$

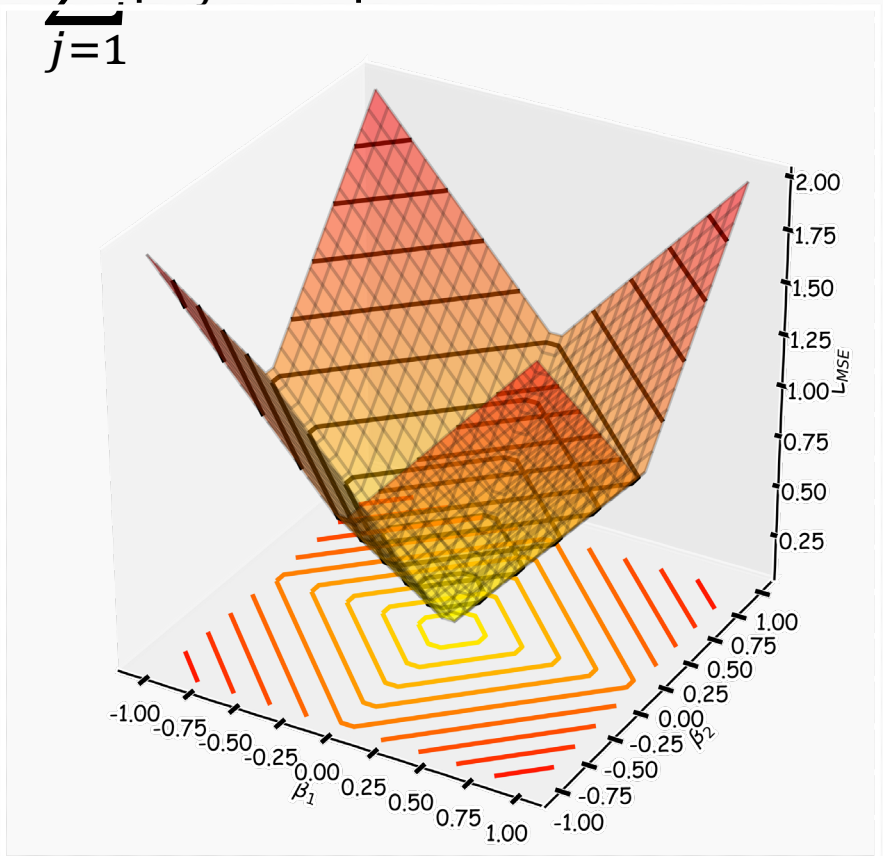


The Geometry of Regularization (LASSO)

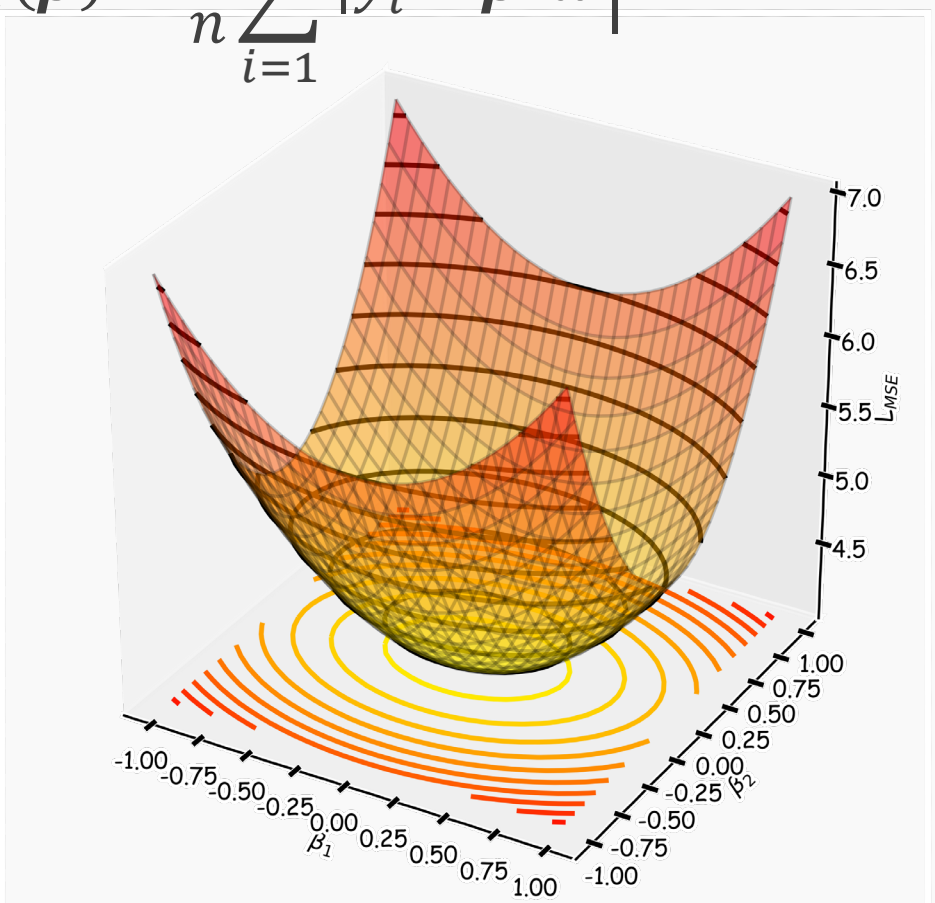
$$L_{LASSO}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T \boldsymbol{x}\Big|^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

$$\widehat{\boldsymbol{\beta}}^{LASSO} = \operatorname{argmin} L_{LASSO}(\boldsymbol{\beta})$$

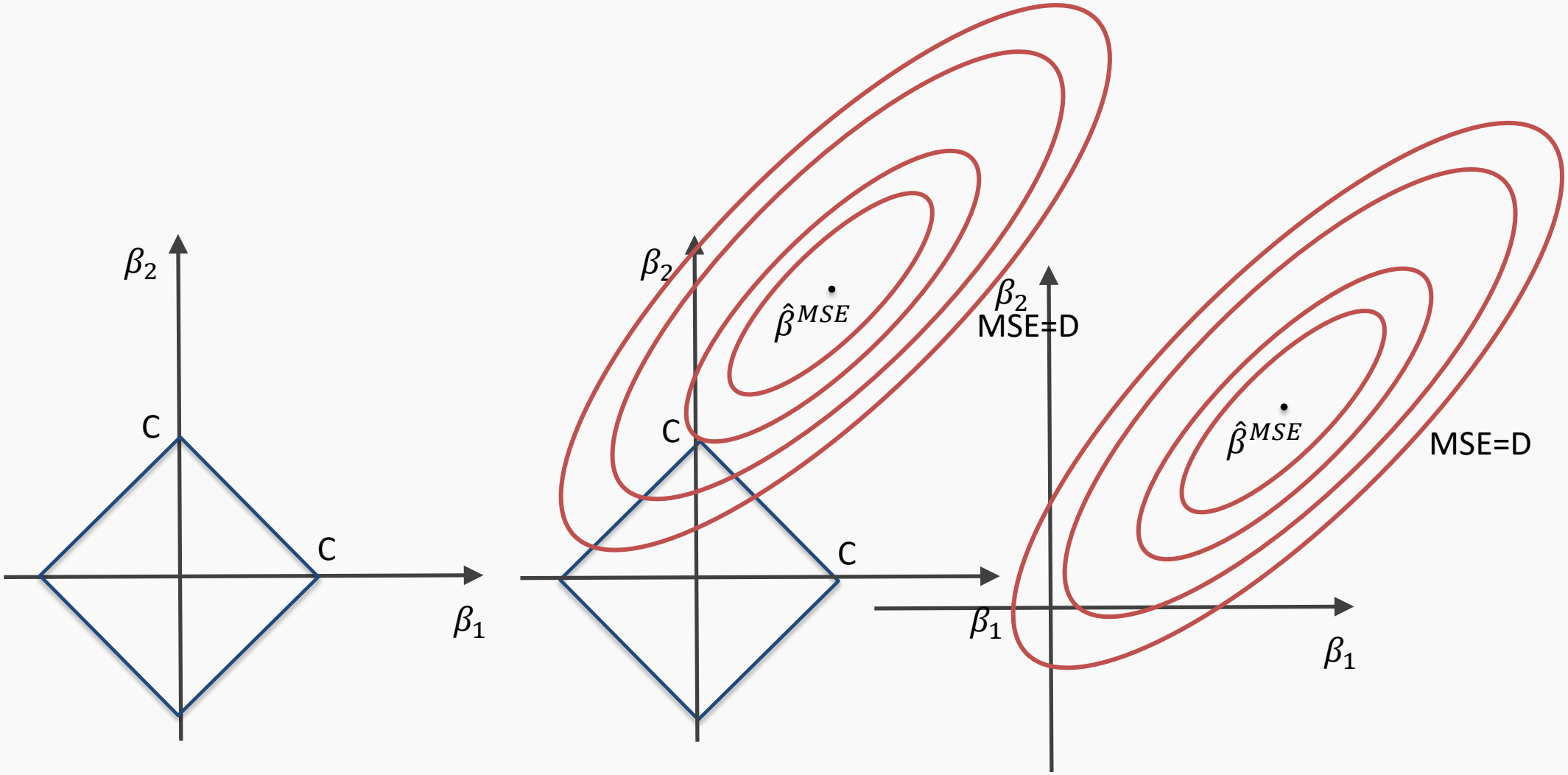
$$L_1 = \lambda \sum_{j=1}^{J} \left| \hat{\beta}_j^{LASSO} \right|$$



$$L_{MSE}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T \boldsymbol{x}\Big|^2$$



The Geometry of Regularization (LASSO)

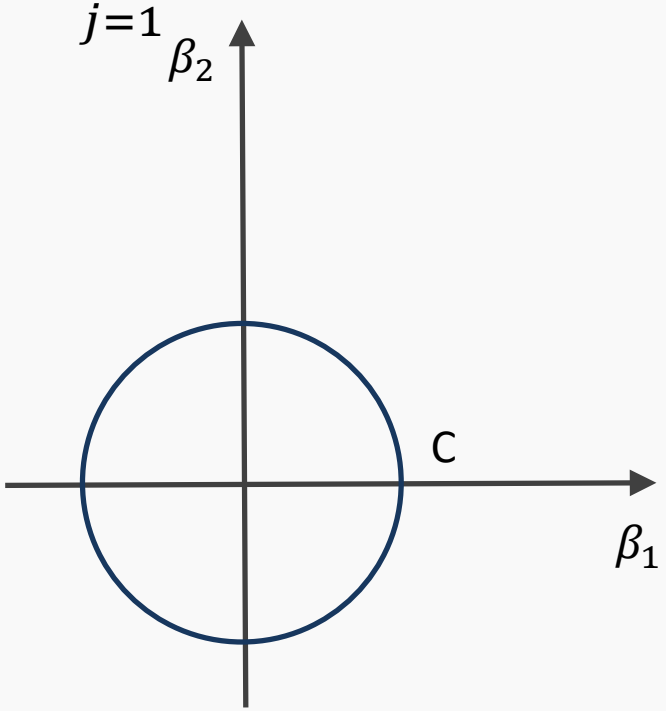


The Geometry of Regularization (Ridge)

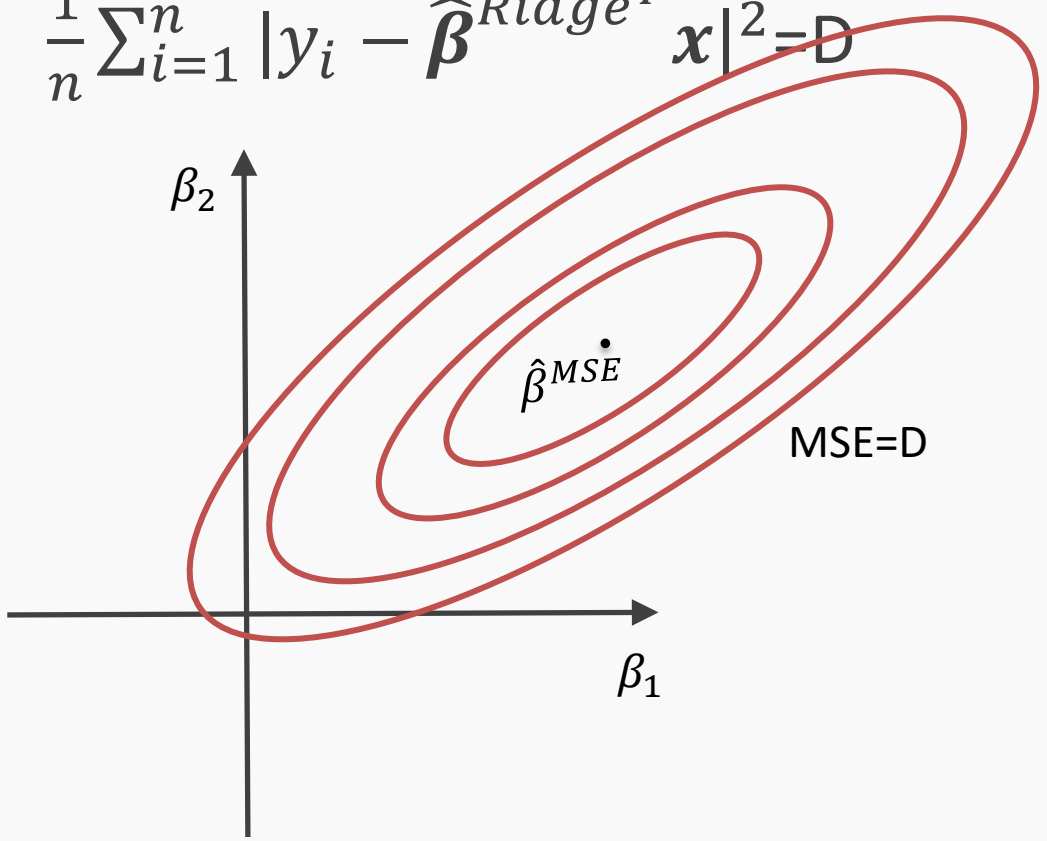
$$L_{Ridge}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T \boldsymbol{x}|^2 + \lambda \sum_{j=1}^{J} (\beta_j)^2$$

$$\widehat{\boldsymbol{\beta}}^{Ridge} = \operatorname{argmin} L_{Ridge}(\boldsymbol{\beta})$$

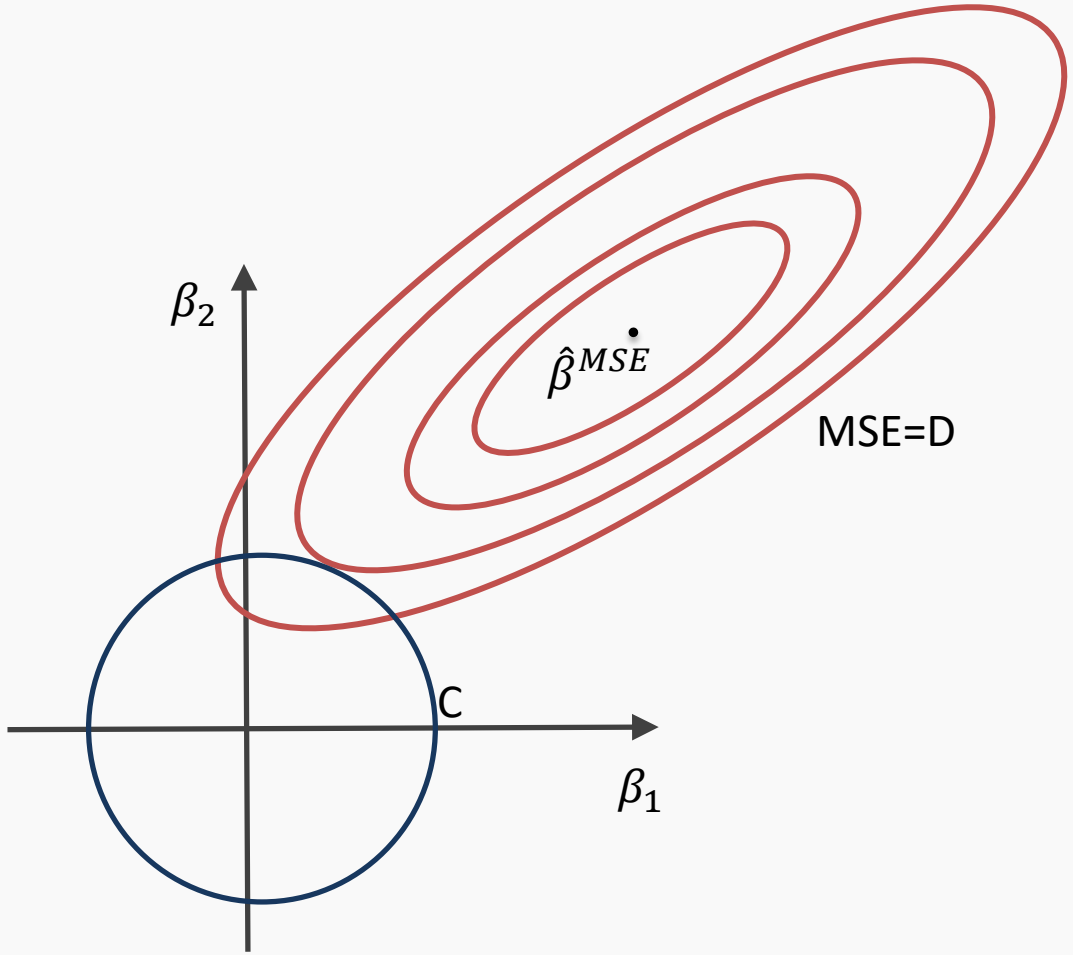
$$\lambda \sum_{j=1}^{J} \left|\hat{\beta}_j^{Ridge}\right|^2 = C$$



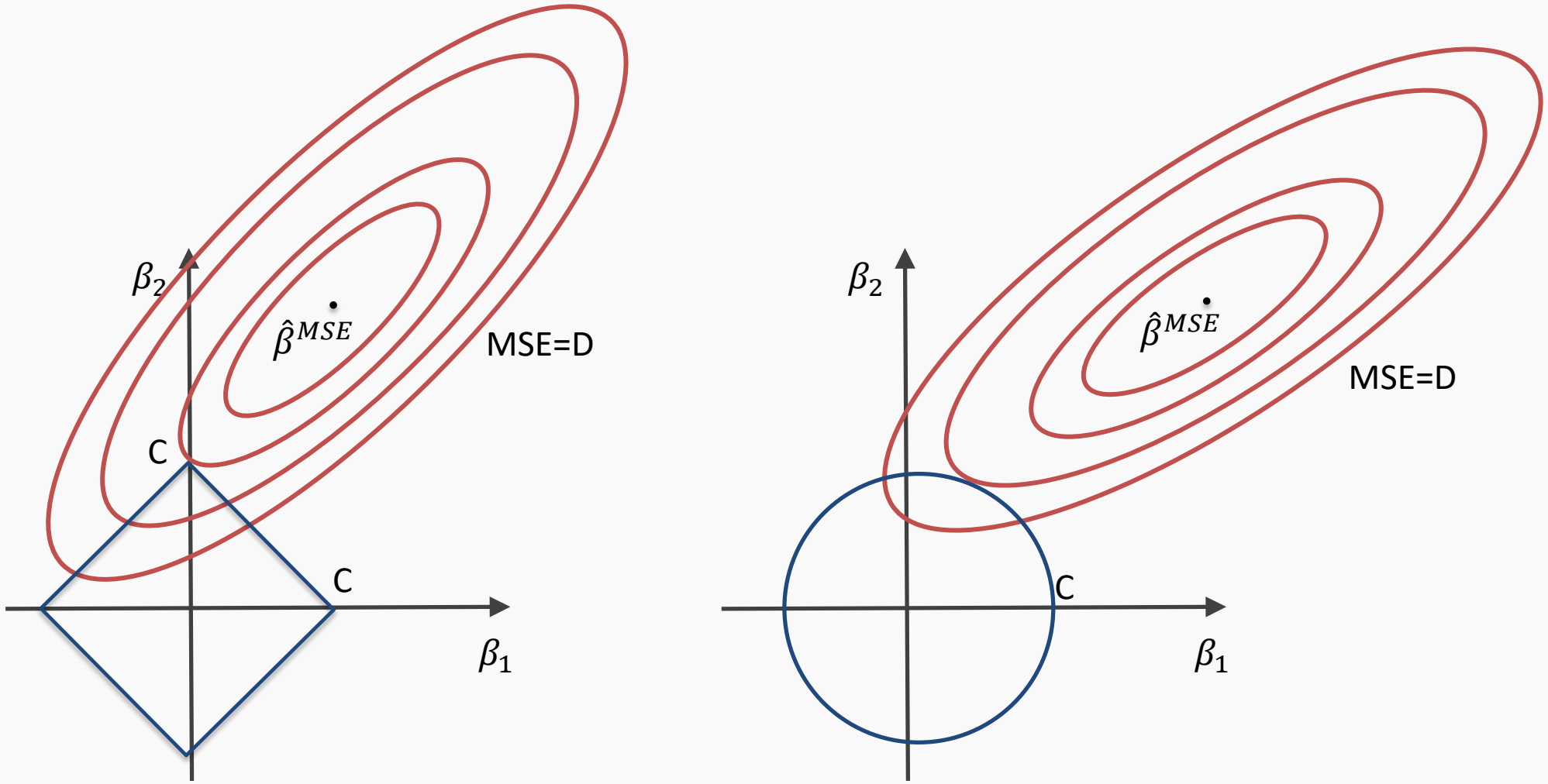
$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{\boldsymbol{\beta}}^{Ridge^T} \boldsymbol{x}|^2 = D$$



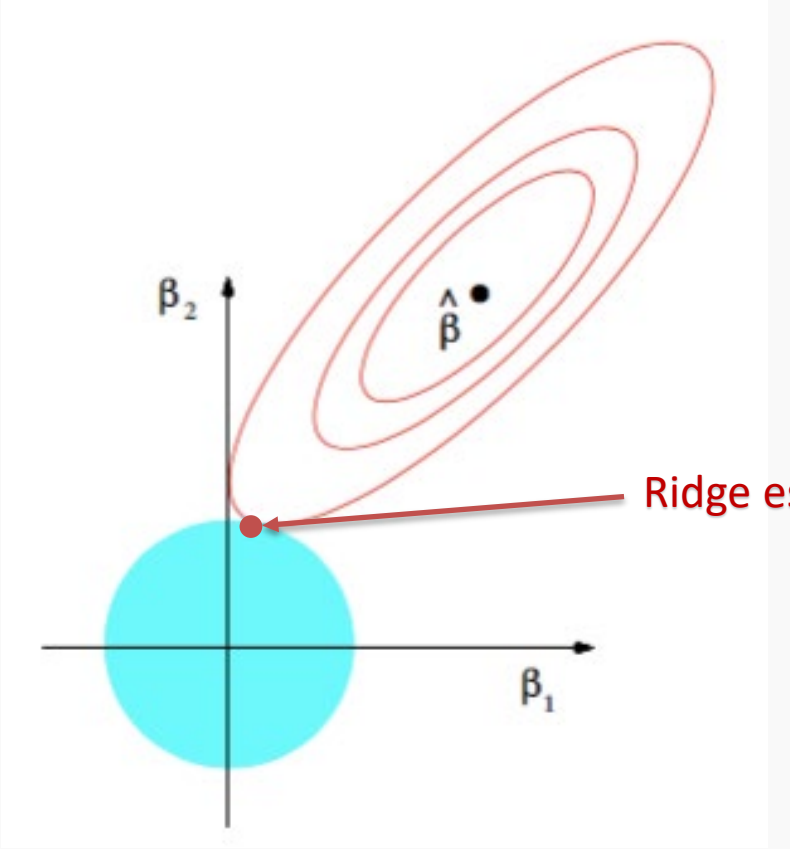
The Geometry of Regularization (Ridge)



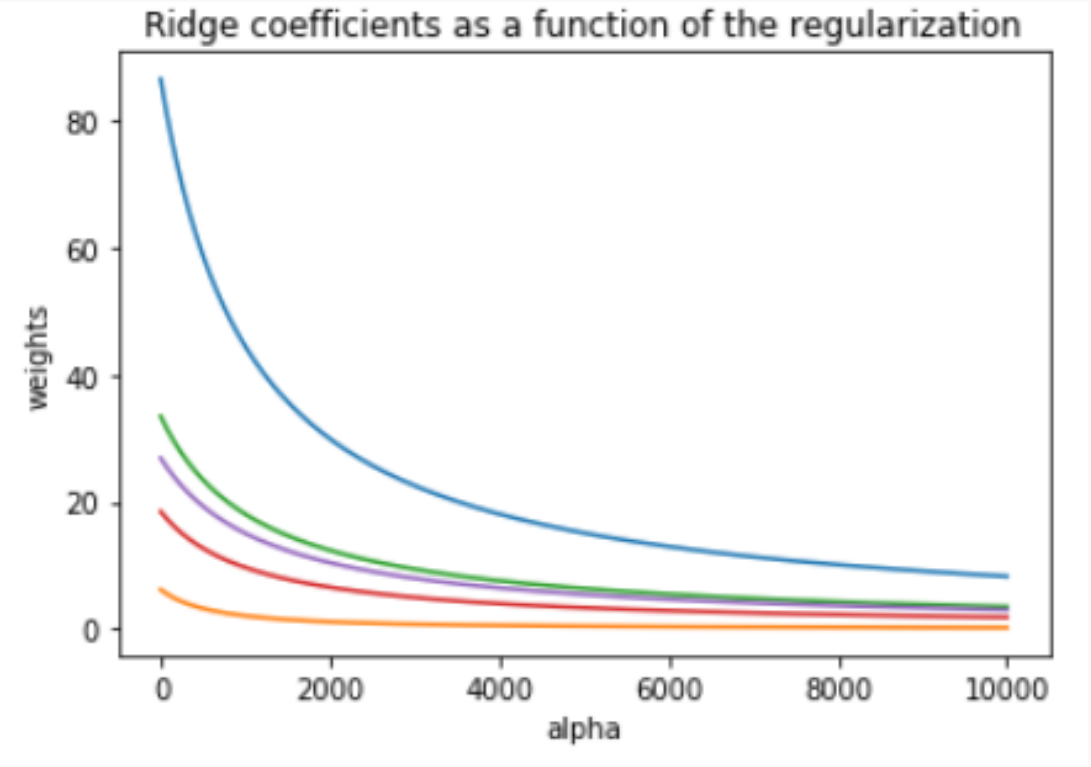
The Geometry of Regularization



Ridge visualized

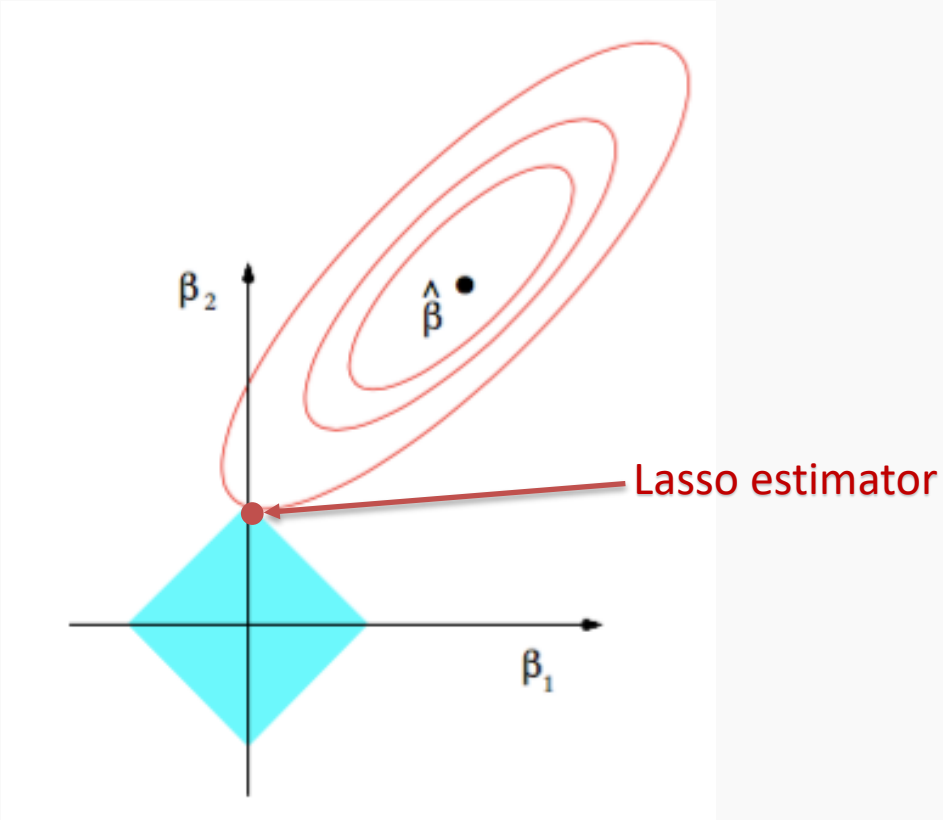


The ridge estimator is where the constraint and the loss intersect.

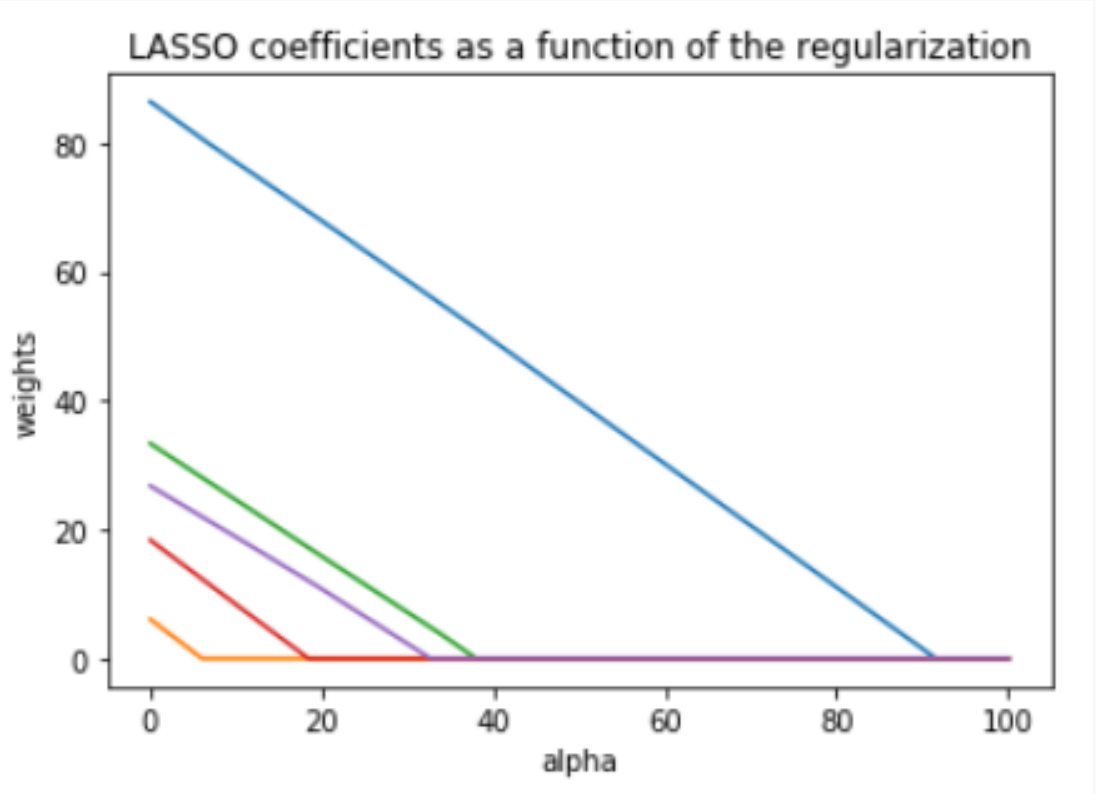


The values of the coefficients decrease as lambda increases, but they are not nullified.

LASSO visualized



The Lasso estimator tends to zero out parameters as the OLS loss can easily intersect with the constraint on one of the axis.

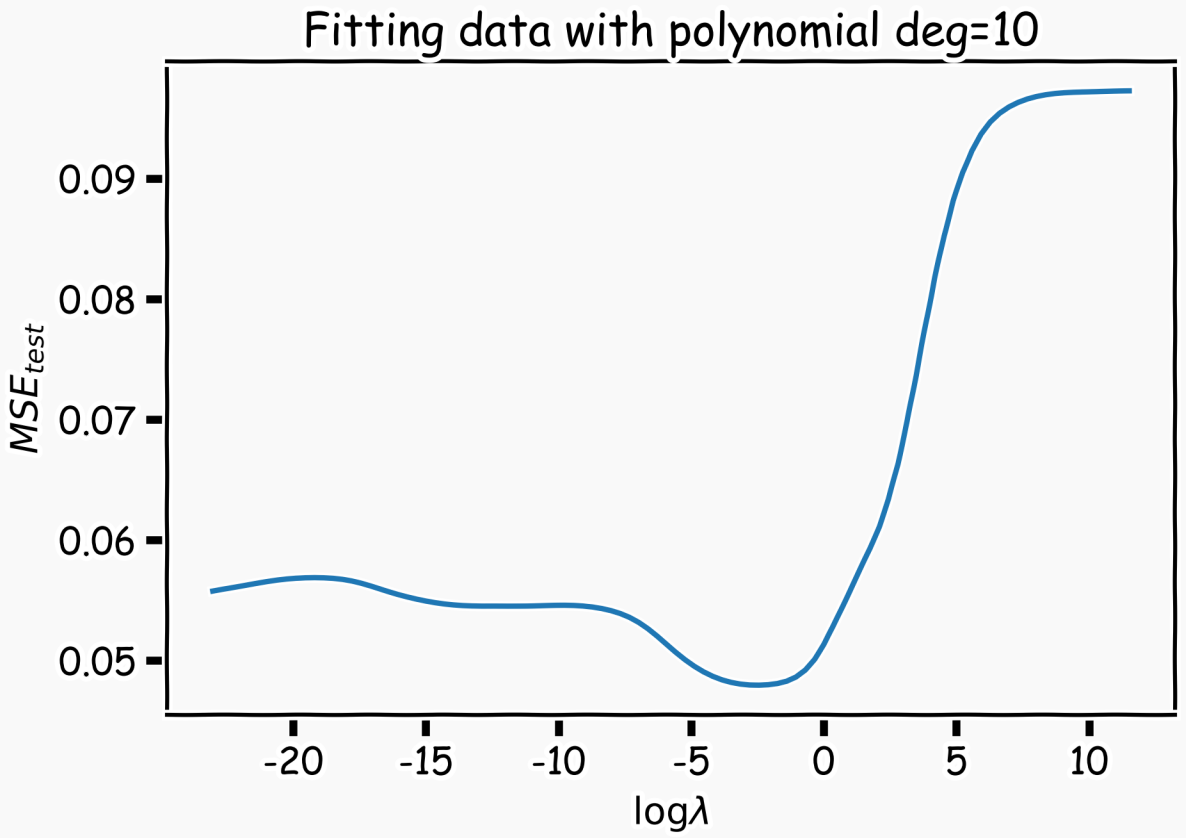


The values of the coefficients decrease as lambda increases, and are nullified fast.

Ridge regularization with **validation** only: step by step

1. split data into $\{\{X,Y\}_{train},\{X,Y\}_{validation},\{X,Y\}_{test}\}$
2. for λ in $\{\lambda_{min},\dots\lambda_{max}\}$:
    1. determine the β that minimizes the L_{ridge}, $\hat{\beta}_{Ridge}(\lambda)=\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}+\lambda I\right)^{-1}X^{T}Y$, using the train data.
    2. record $L_{MSE}(\lambda)$ using validation data.
3. select the λ that minimizes the loss on the validation data,
$$\lambda_{ridge}=\mathrm{argmin}_{\lambda}\,L_{MSE}(\lambda)$$
4. Refit the model using both train and validation data, $\{\{X,Y\}_{train},\{X,Y\}_{validation}\}$, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
5. report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with **validation** only: step by step



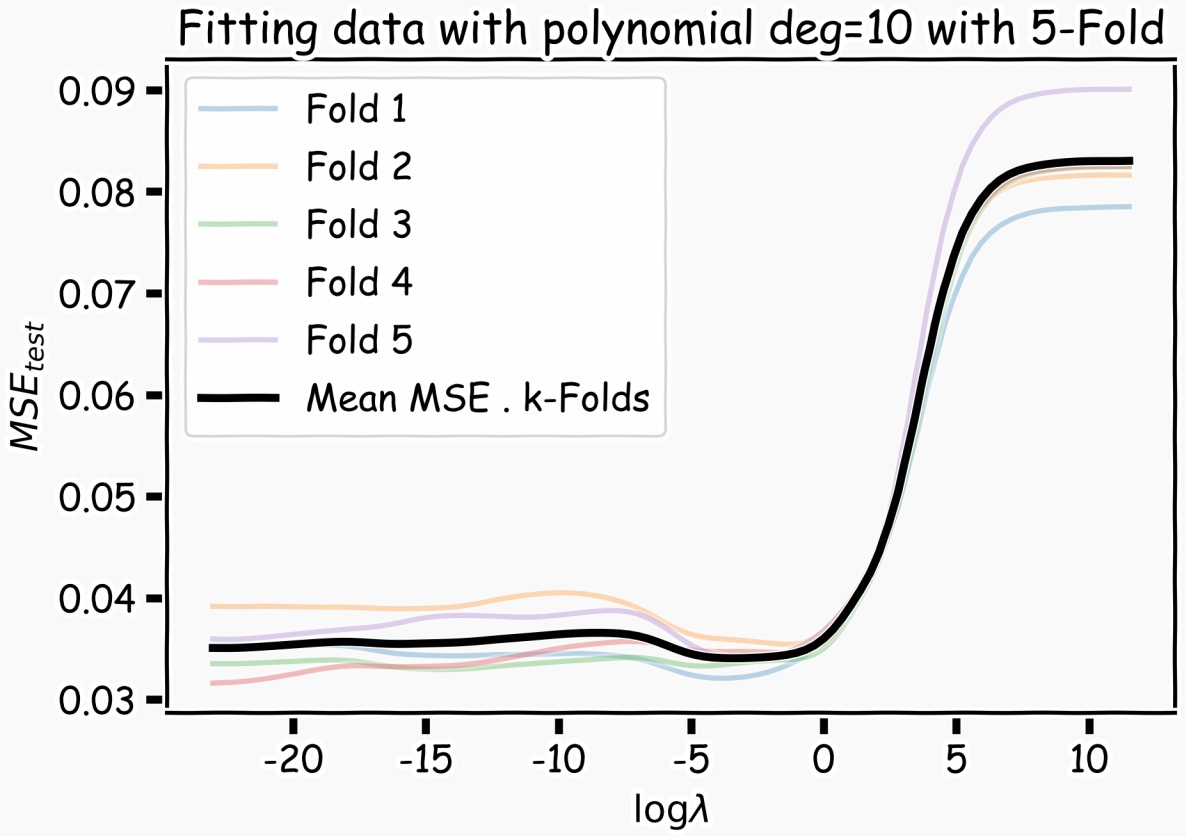
Ridge regularization with **CV**: step by step

	λ_1	λ_2	…	λ_n
k_1	L_{11}	L_{12}	..	…
k_2	L_{21}	…	..	…
…	..	…	..	…
k_n	…	…	…	…
E[]	$\bar{L}_1$	$\bar{L}_2$	…	$\bar{L}_n$

1. remove $\{X,Y\}_{test}$ from data
2. split the rest of data into K folds, $\{\{X,Y\}_{train}^{-k}, \{X,Y\}_{val}^{k}\}$
3. for k in $\{1,\dots,K\}$
    1. for λ in $\{\lambda_0,\dots,\lambda_n\}$:
        A. determine the β that minimizes the L_{ridge}, $\hat{\beta}_{ridge}(\lambda,k) = \left(\mathrm{X^T X} + \lambda I\right)^{-1} X^T Y$, using the train data of the fold, $\{X,Y\}_{train}^{-k}$.
        B. record $L_{MSE}(\lambda,k)$ using the validation data of the fold $\{X,Y\}_{val}^{k}$

        At this point we have a 2-D matrix, rows are for different k, and columns are for different λ values.
4. Average the $L_{MSE}(\lambda,k)$ for each λ, $\bar{L}_{MSE}(\lambda)$.
5. Find the λ that minimizes the $\bar{L}_{MSE}(\lambda)$, resulting to λ_{ridge}.
6. Refit the model using the full training data, $\{\{X,Y\}_{train}, \{X,Y\}_{val}\}$, resulting to $\hat{\beta}_{ridge}(\lambda_{ridge})$
7. report MSE or R^2 on $\{X,Y\}_{test}$ given the $\hat{\beta}_{ridge}(\lambda_{ridge})$

Ridge regularization with **validation** only: step by step



Variable Selection as Regularization

Since LASSO regression tend to produce zero estimates for a number of model parameters - we say that LASSO solutions are **sparse** - we consider LASSO to be a method for variable selection.

Many prefer using LASSO for variable selection (as well as for suppressing extreme parameter values) rather than stepwise selection, as LASSO avoids the statistic problems that arises in stepwise selection.

Question: What are the pros and cons of the two approaches?