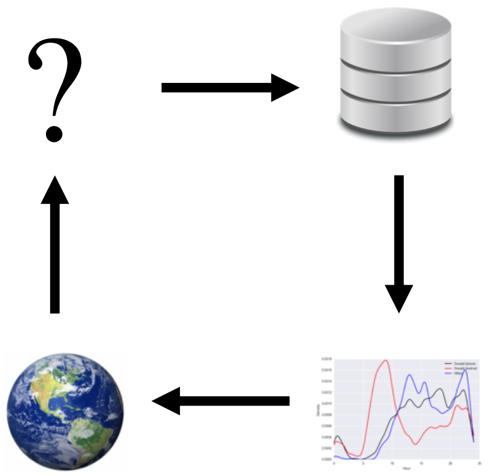
Data Science 100

Lecture 17:

Feature Engineering
Prediction & Cross-validation
Regularization

Data Science Life Cycle



Context
Question
Refine Question to an one answerable with data

Design
Data Collection
Data Cleaning

Modeling
Test-train split
Loss function choice
1. Feature engineering
    Transformations,
    Dummy Variables
    Word vectors

Model evaluation
2. Prediction error

3. Model selection
    Best subset regression
    Cross-Validation
    Regularization

State of the Union Addresses

The first State of the Union Address

***

State of the Union Address
George Washington
December 8, 1790

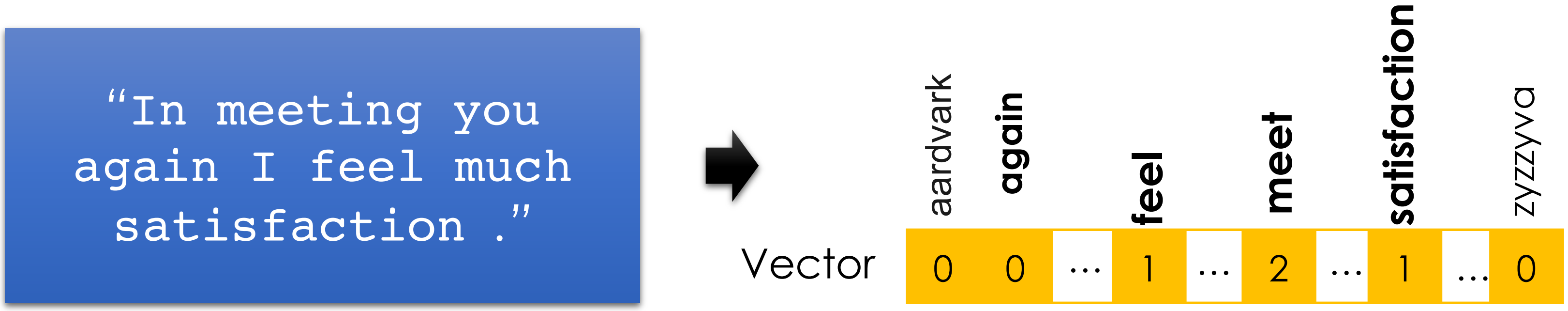
Fellow-Citizens of the Senate and House of Representatives:
In meeting you again I feel much satisfaction in being able to repeat my congratulations on the favorable prospects which continue to distinguish our public affairs. The abundant fruits of another year have blessed our country with plenty and with the means of a flourishing commerce. …

How do we Analyze/Visualize Text?

- **Derived variables** encoding the presence or absence of particular patterns
  - **Example:** food safety violation descriptions → hasUnclean, hasVermin
- **Word frequencies** which words are more common …
  - *An approach for comparing speeches based on word usage*

Text Encoding

- Generalization of one-hot-encoding for a string of text
  - Often remove **stop words** (e.g., the, is, a…) that don't contain significant information
  - Reduce similar words (e.g. meet, meeting, meets, met) to their **stem**
  - Pool all of the words in all speeches into a **bag of words**



"In meeting you again I feel much satisfaction ."

	aardvark	**again**		**feel**		**meet**		**satisfaction**		zyzzyva
Vector	0	0	…	1	…	2	…	1	…	0

Word Frequency Vectors

- Encode text as a long vector of word counts
  - Typically high dimensional and very sparse
  - Word order information is lost… (is this an issue?)
- We have 226 speeches –
  - each speech is turned into a word vector
  - Row in a matrix with:
    - 226 rows
    - 23,127 columns – corresponding to the unique words in all speeches

State of the Union Addresses

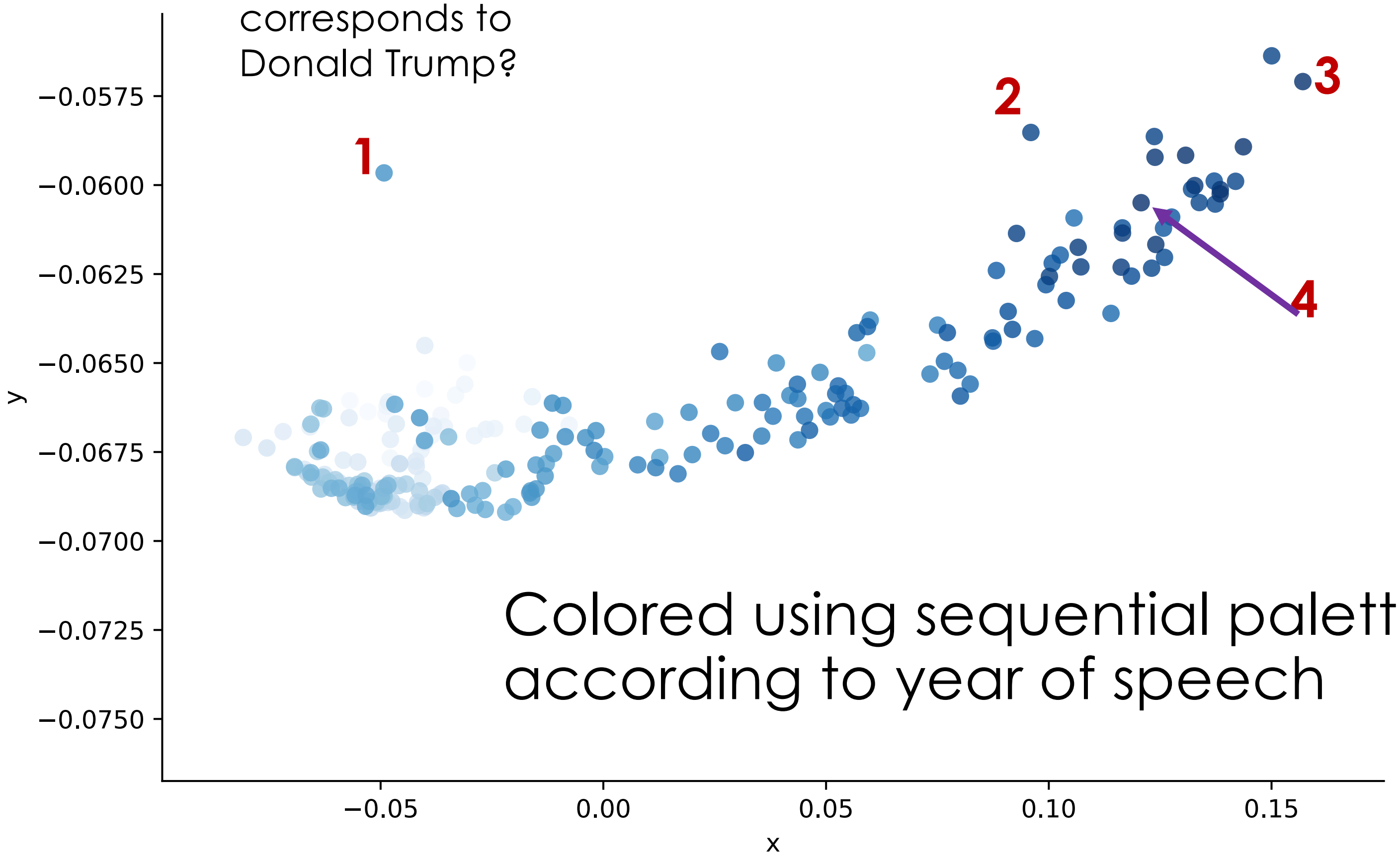
- Dimensionality reduction: ***23,127 columns → 2 columns***
  - Rather than look at projections and the Euclidean distance between points, we define a special distance useful for word vectors. It normalizes by the rarity of a word

term frequency/document frequency =

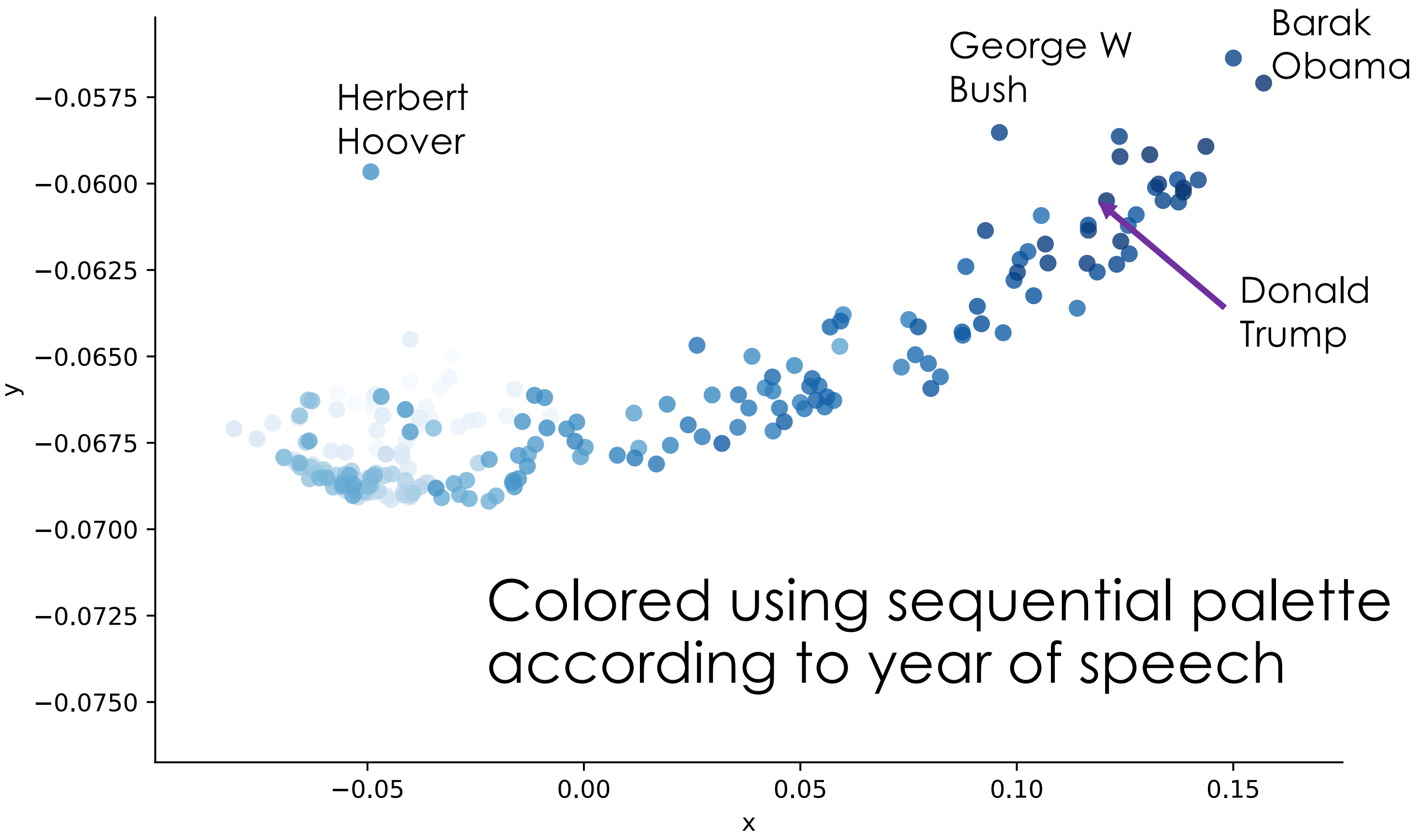
\# times a word appears in doc / #docs contain word

Use this quantity and an approach similar to PCA to reduce each speech to point in 2^d

Which point corresponds to Donald Trump?



Colored using sequential palette according to year of speech



Colored using sequential palette according to year of speech

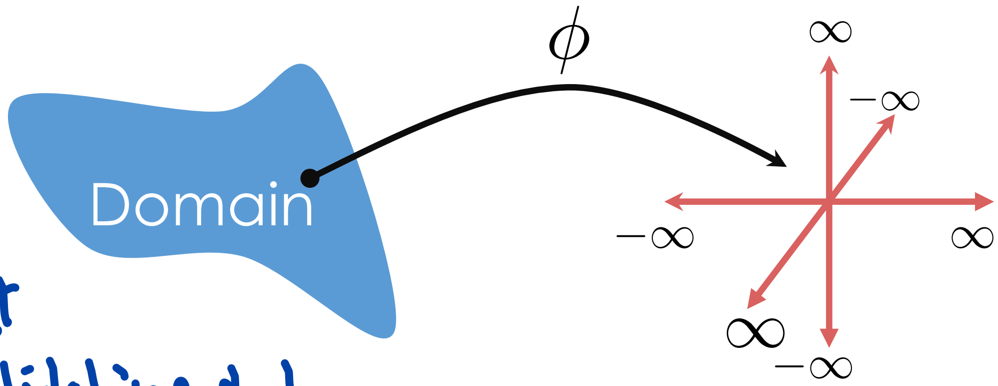
Feature Engineering

Keeping it *Real*

Feature Engineering

Feature Functions:

$$\phi : \mathcal{X} \to \mathbb{R}^p$$



text
qualitative data
quantitative data

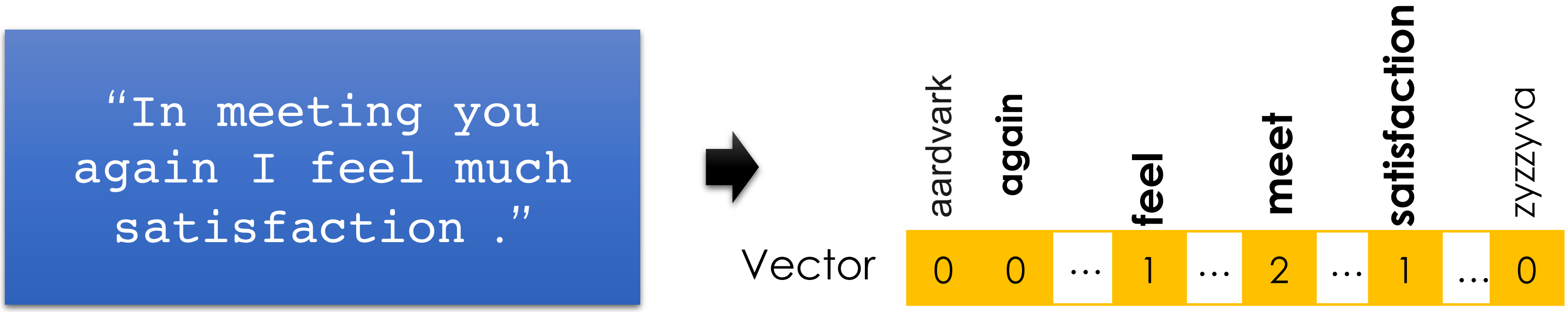
one-hot encoding
word vectors
transformations
eg. polynomials
log

- **One-hot encoding:** *Categorical Data*

state
NY
WA
CA

AL	…	CA	…	NY	…	WA	…	WY
0	…	0	…	1	…	0	…	0
0	…	0	…	0	…	1	…	0
0	…	1	…	0	…	0	…	0

- **Bag-of-words & N-gram:** *Text Data*



“In meeting you again I feel much satisfaction .”

	aardvark	again		feel		meet		satisfaction		zyzzyva
Vector	0	0	…	1	…	2	…	1	…	0

- **Custom Features:** *Domain Knowledge & EDA*

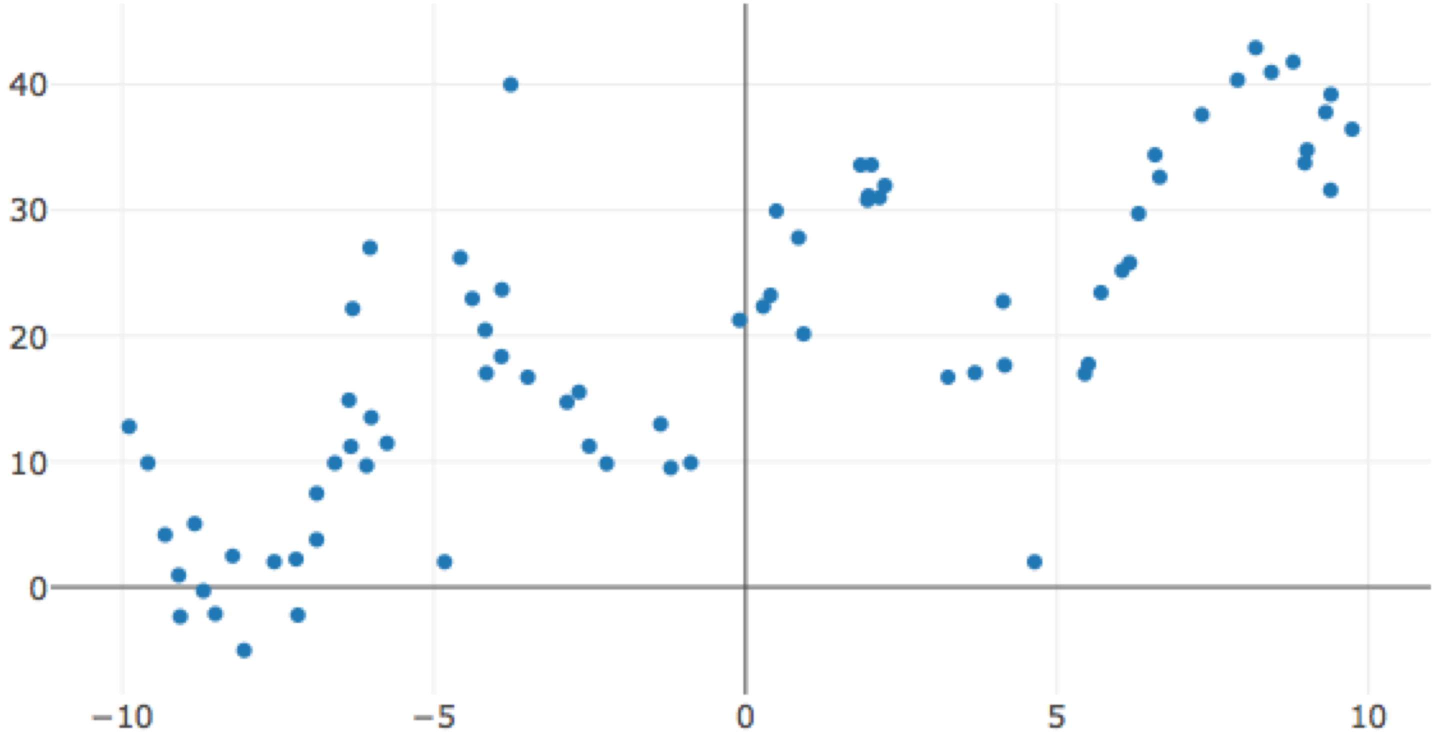
$$\log(x) \qquad\qquad \phi(\text{lat}, \text{lon}, \text{amount}) = \frac{\text{amount}}{\mathbf{Stores}[\mathbf{ZipCode}[\text{lat}, \text{lon}]]}$$

- ***Generic Features:*** *polynomials, orthogonal polynomials, cubic splines, basis functions:*

$$\phi_1(x), \ldots, \phi_k(x)$$

Is this data Linear?

What does it mean to be linear?



What does it mean to be a linear model?

$$f_\beta(\phi(x)) = \phi(x)^t\beta = \sum_{j=1}^{p} \phi_j(x)\beta_j$$

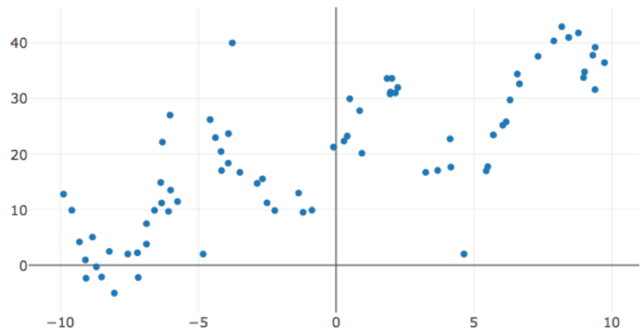
In what sense is the above **model linear**?

We have a linear combination of ϕ_j's

Examples of Non-linear Feature Functions

- In our toy dataset there appears to be cyclic patterns
- One reasonable collection of feature functions might be:

$$\phi(x) = [x, \sin(x), 1]$$

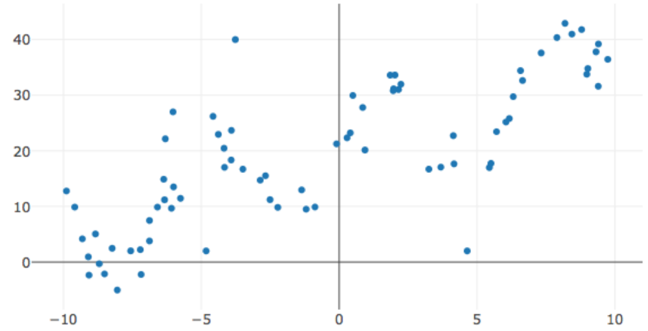


$$f_\beta(\phi(x)) = \vec{1}\beta_0 + \vec{x}\beta_1 + \sin(x)\beta_2$$

Examples Non-linear Feature Functions

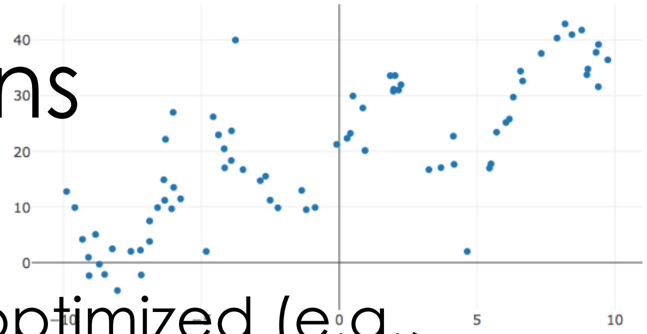
- Linear models don't include model parameters in non-linear transformations!

$$f_\beta(\phi(x)) = \beta_0 + x\beta_1 + \sin(\overbrace{\beta_3 x + \beta_4}^{\text{problematic}})\beta_2$$



- This **is not a linear model**!

Non-linear Feature Functions



- **hyper-parameters** that are externally optimized (e.g., using a grid search and not the normal equations…)
  - Often by trying a range of values.
  - This **is a linear model:** we minimize over the betas

$$f_\beta(\phi(x)) = \beta_0 + x\beta_1 + \sin(\gamma x + \alpha)\beta_2$$

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 \sin(5x)$$

Model Selection

Kenya Variables

- Girth, Length, Height, BCS, Age, Sex
- Not including transformations, how many possible models could we have examined?

$2^6 = 64$

- What if we used two-way interaction terms, e.g., Girth x Age?

$2^6 + (6 \text{ choose } 2) = 64 + 15 = 79$

here we include
only some of the possible models –
For each two-variable model, we
also include the product of the 2 variables

How we chose the model

- We use a physical model as a starting point
- We considered model complexity
- We made residual plots
- We examined MSE (Mean Square Error, AKA Empirical Risk)
- BUT, we didn't look at all 64 models
- What are other ways to choose a model?

Best Subset Regression

- Fit all 64 models:
  - 6 one-variable models
  - 15 two-variable models
  - 20 three-variable models
  - 15 four-variable models
  - 6 five variable models
  - 1 6-variable model

(Note that I am counting each qualitative variable as 1, which isn't quite right)

For each degree, (one-variable, two-variable, etc.),

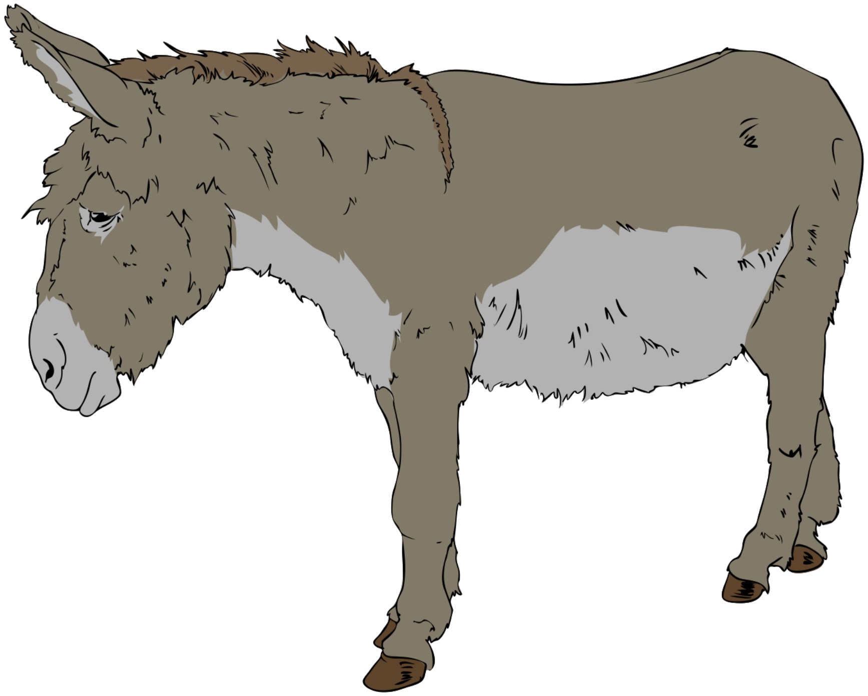
find the model that minimizes the empirical risk

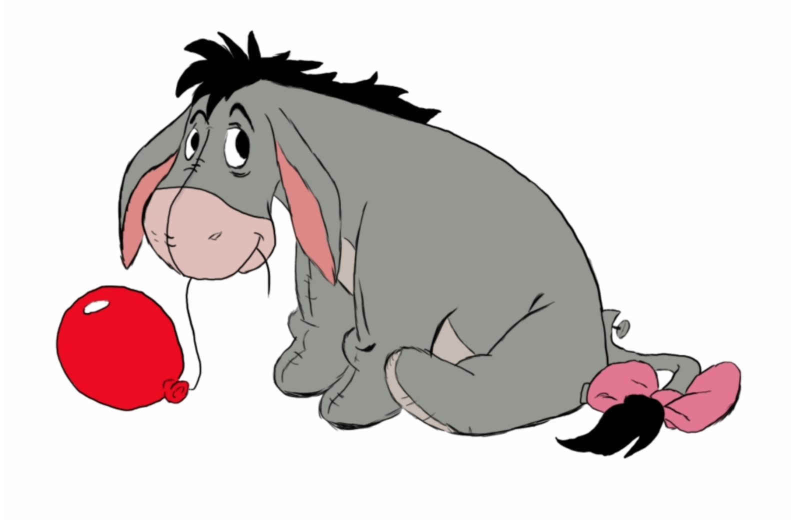
We still have a problem: how many variables do we settle on for the model?

How to choose the best size model?

- Ideally we want to do well in predicting a donkey in the future
- We used the 500+ donkeys to fit the models
- We want the model to do well at predicting the weight of a donkey that we have not seen/measured.

Along comes a new donkey…





How to choose the best size model?

The unseen donkey will be from the same distribution of donkeys as the ones that we have seen already.

We want the expected loss for this new donkey to be small

$$\mathbb{E}(Y_0 - \hat{Y}_0)^2$$

We obtain the x_0 values for this donkey, and estimate weight with

$x_0 \cdot \hat{\beta}$

Recall, that to estimate the prediction error, we set aside 20% of the data to assess our model after we chose it

But we need help choosing the model!

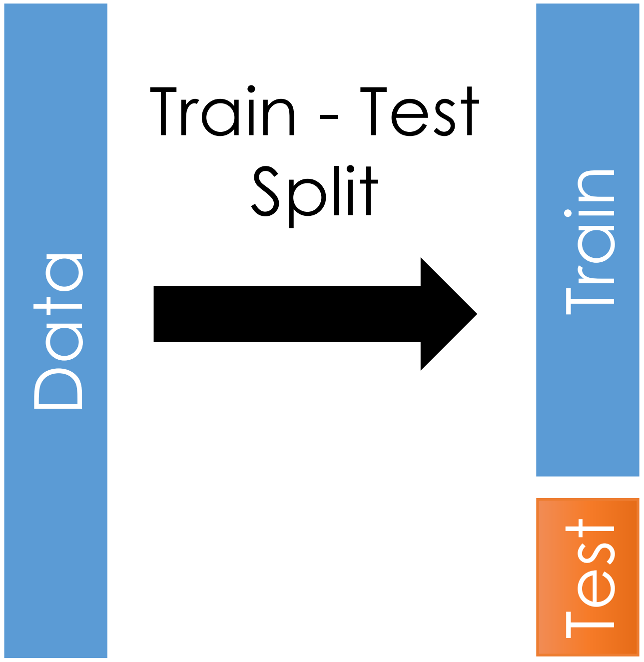
The Train-Test Split

- **Training Data:** used to fit model
- **Test Data:** check generalization error

$$\mathbb{E}(Y_0 - \hat{Y}_0)^2 \approx \frac{1}{m}\sum_{j=1}^{m}(Y_j - x_j^t\hat{\beta})^2$$

Tested on
m = 0.2n
observations

Trained on 0.8xn
observations



Data

Train - Test
Split

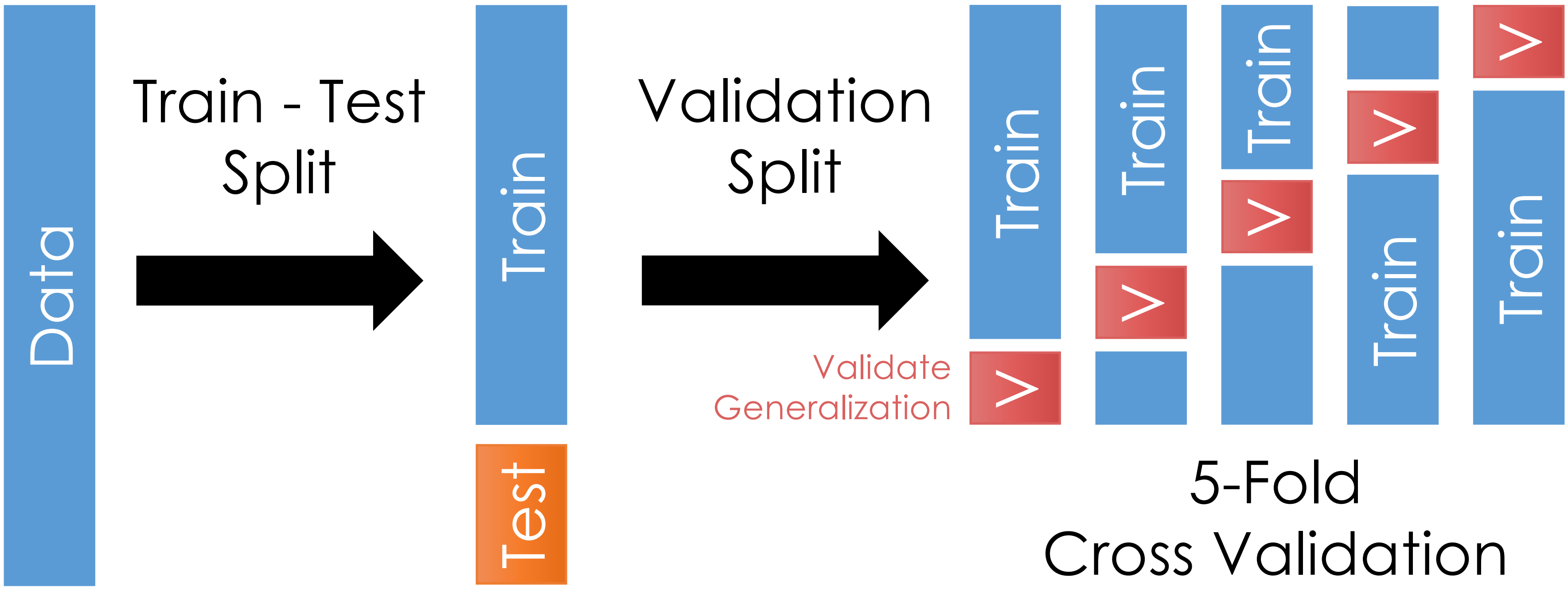
Train

Test

You can only use the test dataset once after deciding on the model.

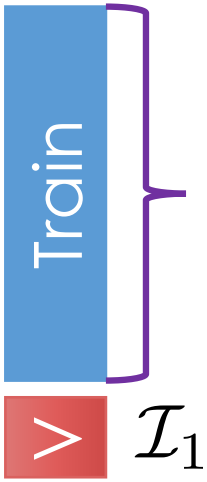
Imitate the test-train split: Cross-validation

Generalization: *Validation Split*



Cross validation simulates multiple train test-splits on the training data.

Generalization: *Validation Split*



$$\mathcal{I}_1^c = \{i \notin \text{fold } 1\}$$

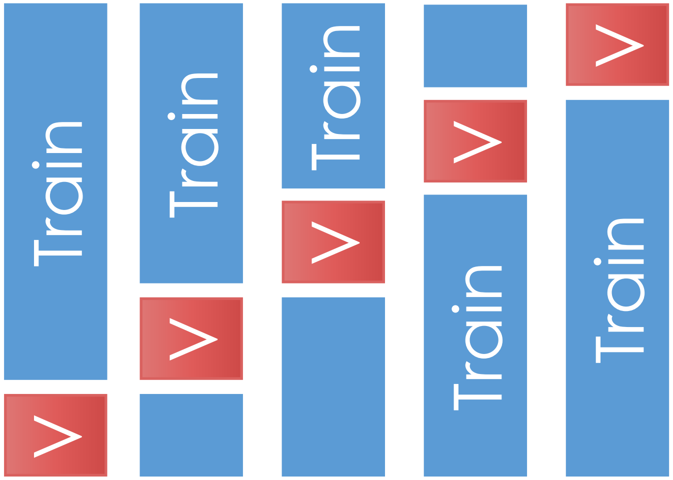
$$\mathcal{I}_1 = \{i \in \text{fold } 1\}$$

$$\sum_{\mathcal{I}_1} \left(y_i - f_{\hat{\beta}_{\mathcal{I}_1^c}}\left(\phi(x_i)\right)\right)^2$$

Compute
square error
for data in
the first fold

Fit model
with data not
in first fold

Generalization: *Validation Split*



How many times does y_1 get used?
however

REPEAT

$$\sum_{k=1}^{5}\sum_{\mathcal{I}_k}(y_i - f_{\hat{\beta}_{\mathcal{I}_k^c}}(\phi(x_i)))^2$$

These 5 sums are not independent of one another, but each summand has independence between the data used to fit the model and the data used to assess prediction error

How to Implement with Best Subset Regression

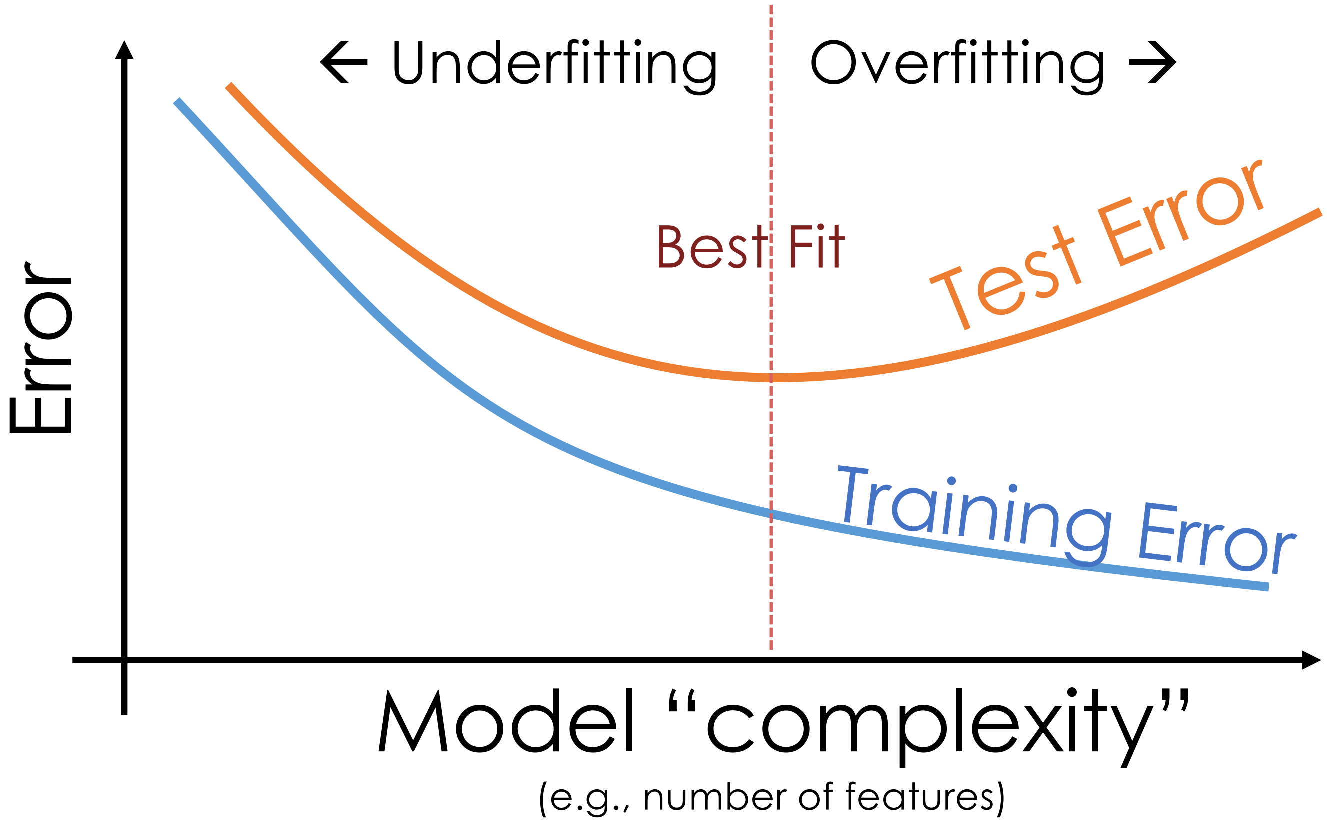
1. Find the best one-variable model for data not in fold 1.
2. Obtain the loss for that one-variable model using fold 1.
3. Repeat for folds 2, 3, 4, 5
4. Combine into one assessment for the best one-variable model

Repeat for each size model.

Select the model size according to the minimum cross-validated error.

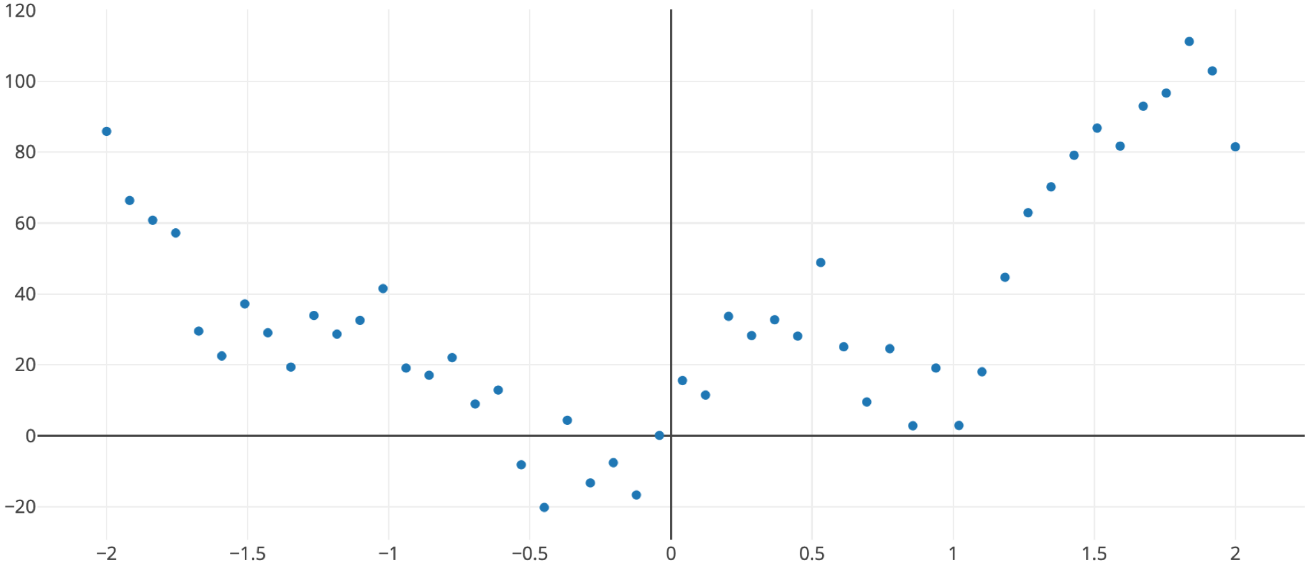
Find the best model for that size using all of the training data.

Training vs **Test** Error



Fitting Polynomials: Cross-validation

$$\overbrace{\beta_0 \sin(5x) + \beta_1 x + \beta_2 x^2} + \overset{\text{error}}{\epsilon}$$



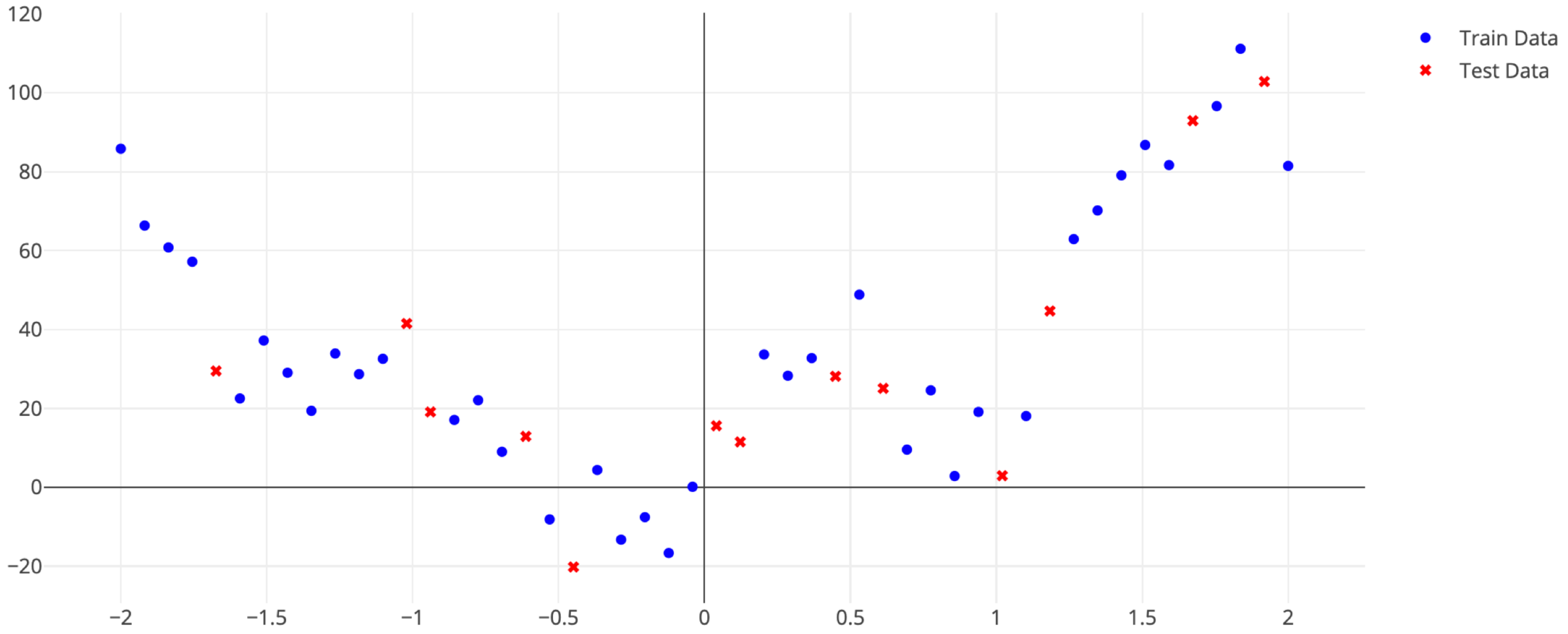
Choose one of 32 models

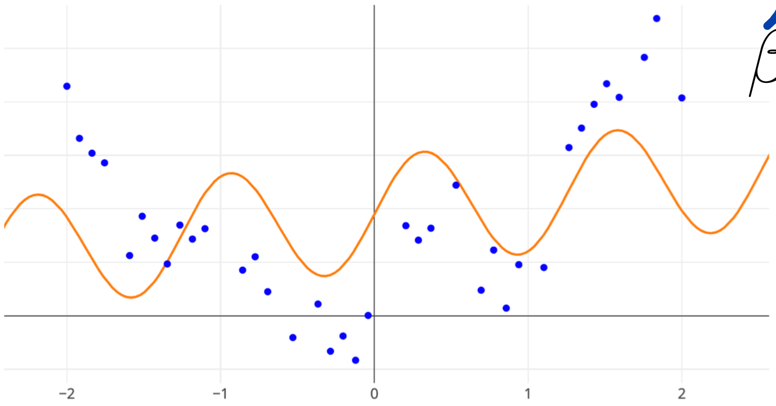
$$\beta_0 \sin(5x) + \beta_1 x$$

$$\beta_0 \sin(5x) + \beta_1 x + \beta_2 x^2$$

$$\beta_0 \sin(5x) + \beta_1 x + \beta_2 x^2 + \cdots + \beta_{32} x^{32}$$

Set aside test set



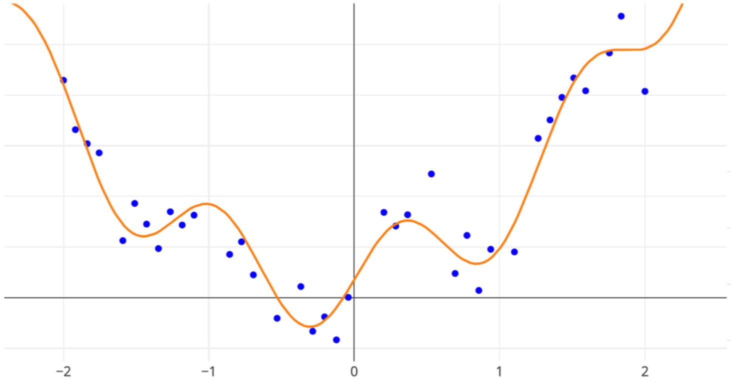


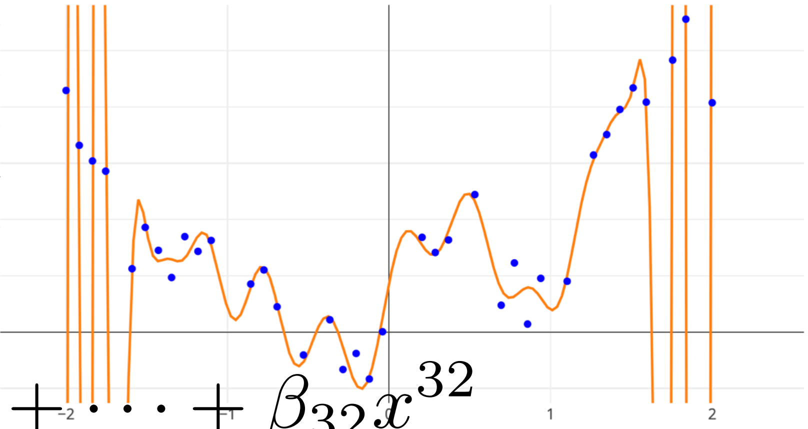
$$\hat{\beta}_0 \sin(5x) + \hat{\beta}_1 x$$

best 1-degree poly

$$\hat{\beta}_0 \sin(5x) + \hat{\beta}_1 x + \hat{\beta}_2 x^2$$

best 2-degree polynom





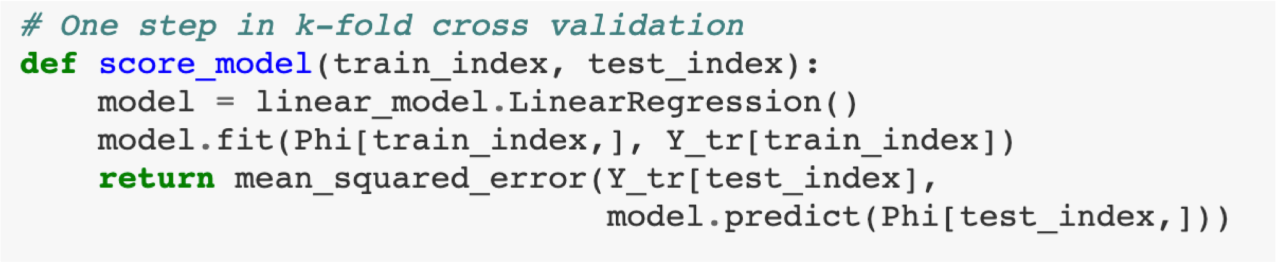
$$\beta_0 \sin(5x) + \beta_1 x + \beta_2 x^2 + \cdots + \beta_{32} x^{32}$$

best 32-degree polynomial

5-fold Cross-Validation

```
from sklearn.model_selection import KFold
kfold_splits = 5
kfold = KFold(kfold_splits, shuffle=True, random_state=42)
```

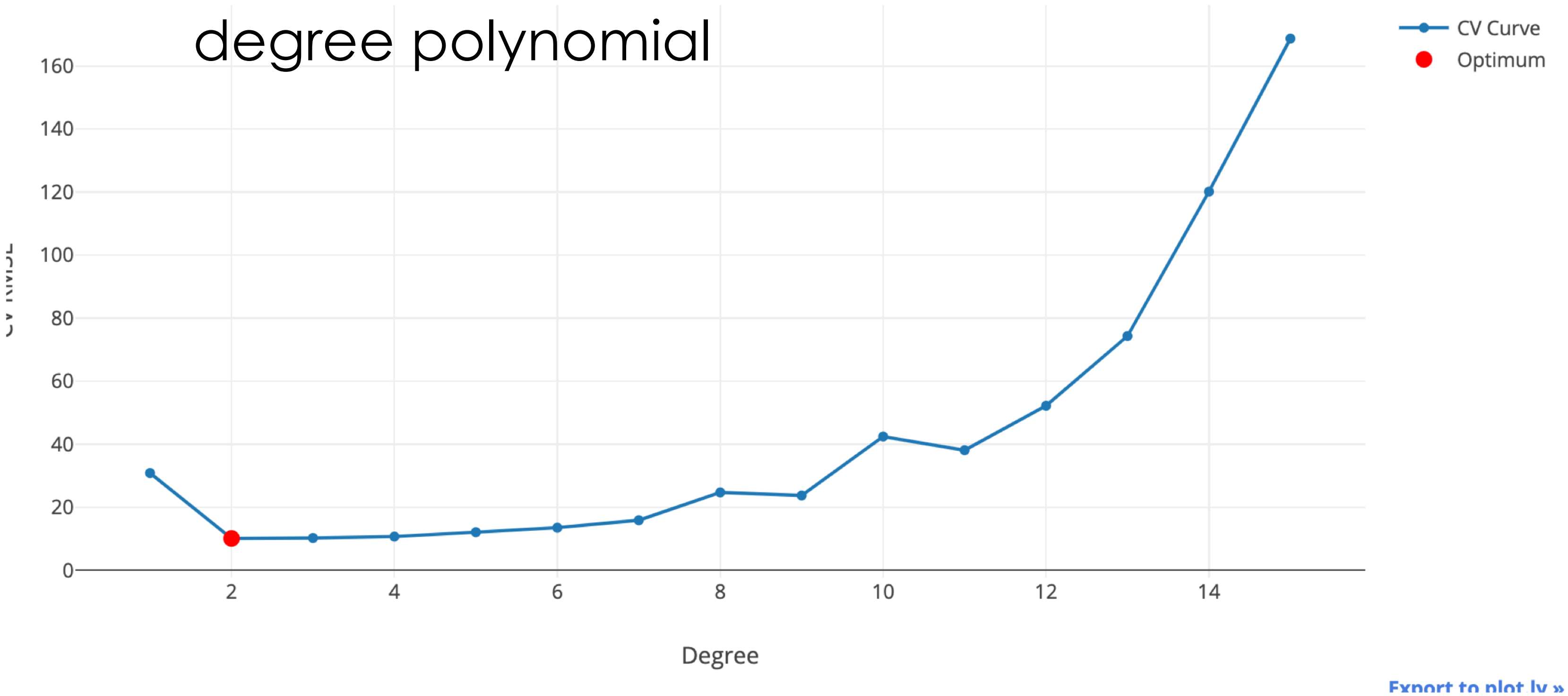
Create 5 random folds



```
# One step in k-fold cross validation
def score_model(train_index, test_index):
    model = linear_model.LinearRegression()
    model.fit(Phi[train_index,], Y_tr[train_index])
    return mean_squared_error(Y_tr[test_index],
                              model.predict(Phi[test_index,]))
```

For each fold, use the fold’s complement to train and the fold to test

Cross-validated
RMSE for each
degree polynomial



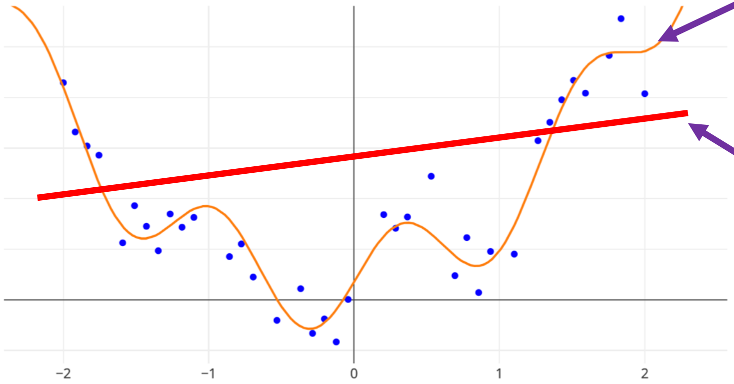
A fundamental challenge in modeling and learning

Fundamental Challenge

- **Bias:** the expected deviation between the predicted value and the true value
- **Variance:** two sources
  - **Observation Variance:** the variability of the random noise in the process we are trying to model.
  - **Estimated Model Variance:** the variability in the predicted value across different training datasets.

Bias

The expected deviation between the predicted function and the true function



True signal

Expected simple linear model

Under-fitting – tends to a large bias

If you could generate the n observations over and over. Each time you fit a simple linear model, The average of these fits is the Expected Simple Linear Model

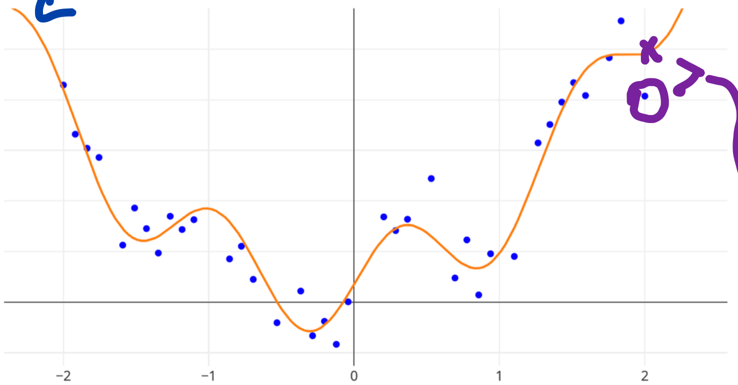
Observation Variance

the variability of the random noise in the process we are trying to model

- ➢ measurement variability
- ➢ stochasticity
- ➢ missing information

Beyond our control (usually)

true curve



The data do not fall on the curve.

$y_i - g(x_i)$

true curve

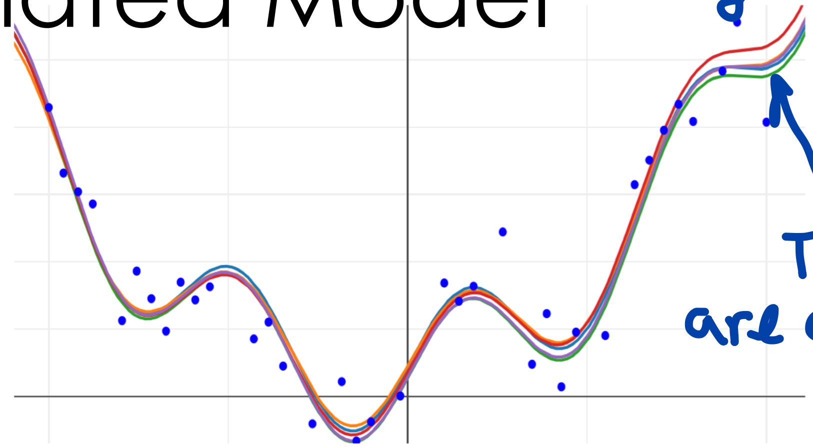
$= \varepsilon_i$

We don't observe ε_i or $g(x_i)$

Variance in the Estimated Model

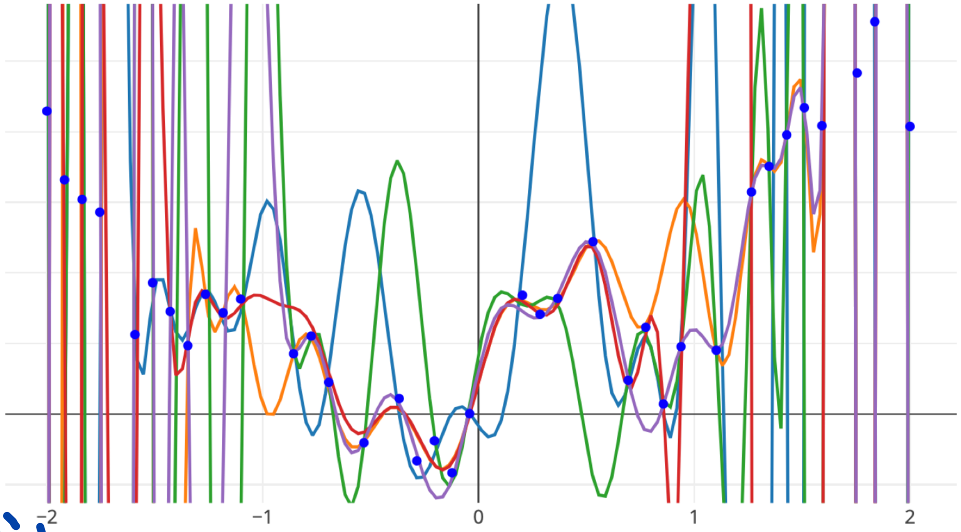
variability in the predicted function across different training datasets

- ➢ Sensitivity to variation in the training data
- ➢ Poor generalization
- ➢ **Overfitting – tends to large variance**



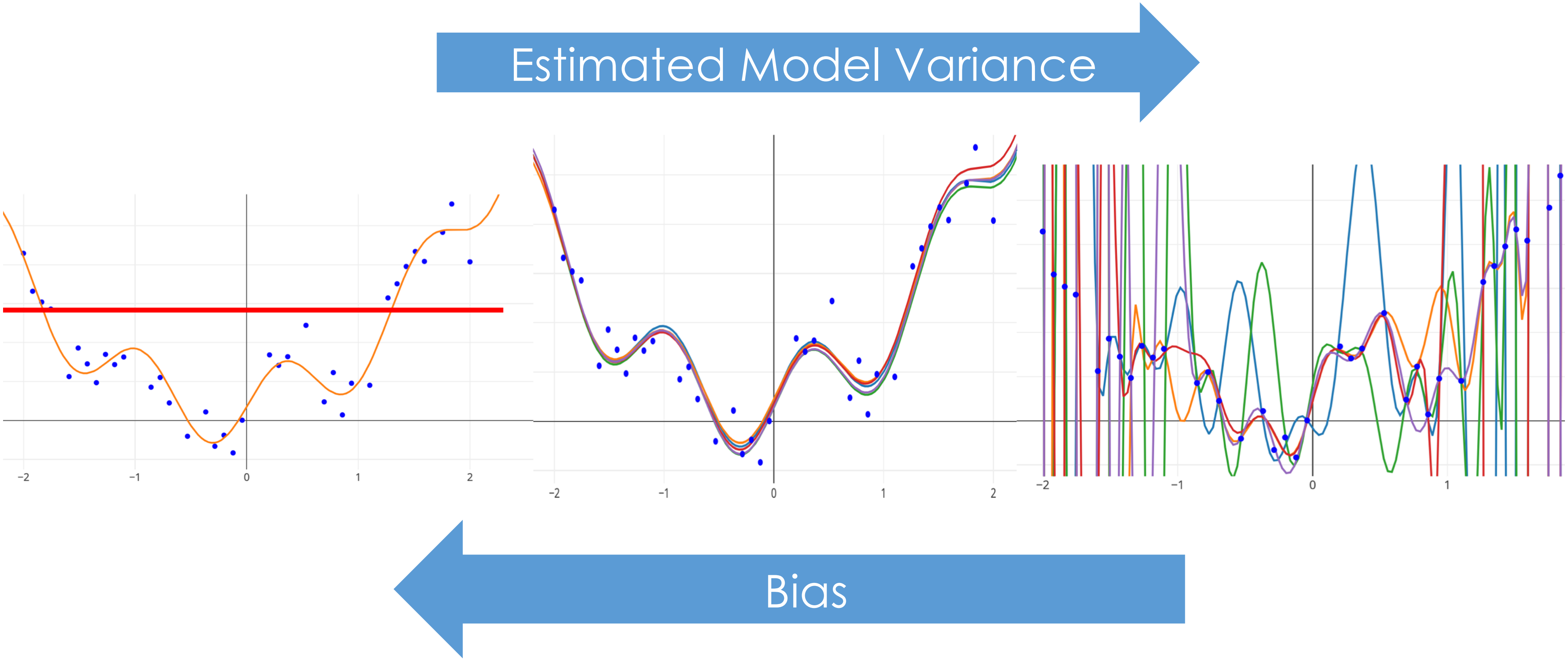
If generate data over and over, each set will give a diff $f_{\hat{\beta}}$

These are all close



these $f_{\hat{\beta}}$ have a lot of variability because they overfit the data so they change a lot from one data set to next

The Bias-Variance Tradeoff



Estimated Model Variance

Bias

Analysis of the Bias-Variance Trade-off

Analysis of Prediction Error

- For the test point ***x*** the expected error:
  - *Random variables are* ***red***

Assume noisy observations
→ Y is a random variable

True Function

$$Y = g(x) + \epsilon$$

Noise term:

$$\mathbf{E}\left[\epsilon\right] = 0$$

$$\mathbf{Var}\left[\epsilon\right] = \sigma^2$$

$$\mathbb{E}(Y - f_{\hat{\beta}}(x))^2$$

Assume ***training data*** is random
→ beta hat is a random
variable

Analysis of Squared Error

Goal: Expected Loss

$$\text{Risk} = \mathbb{E}(Y - f_{\hat{\beta}}(x))^2 =$$

Obs. Var. + (**Bias**)^2 + **Mod. Var.**

Other terminology:

σ^2 + (**Bias**)^2 + **Variance**

$$\mathbb{E}\left[\left(Y - f_{\hat{\beta}}(x)\right)^2\right] = \mathbb{E}\left[\left(\underbrace{g(x) + \varepsilon}_{\text{def of } Y} - f_{\hat{\beta}}(x)\right)^2\right]$$

Rearrange terms

$$= \mathbb{E}\left[\left(\varepsilon + g(x) - f_{\hat{\beta}}(x)\right)^2\right]$$

noise
in observations

true
signal
model

optimized
model for our
data

Add & Subtract $\mathbb{E}\left[f_{\hat{\beta}}(x)\right]$

What is the expected value

The average when we fit the model to new data sets over and over

Expected Loss

$$= \mathbb{E}\Big[\big(\varepsilon + g(x) - \mathbb{E}(f_{\hat{\beta}}(x)) + \mathbb{E}(f_{\hat{\beta}}(x)) - f_{\hat{\beta}}(x)\big)^2\Big]$$

Consider these 3 terms separately

ε - noise $\quad \mathbb{E}(\varepsilon)=0 \quad \mathrm{Var}(\varepsilon)=\sigma^2$

$g(x)-\mathbb{E}[f_{\hat\beta}(x)]$ = bias nothing random here

These are two curves

$f_{\hat\beta}(x)-\mathbb{E}[f_{\hat\beta}(x)]$ = how far is the least squares fit for my data from the expected curve

Expected Loss =

$$\mathbb{E}[\varepsilon^2] + \mathbb{E}\left[\left(g(x) - \mathbb{E}[f_{\hat{\beta}}(x)]\right)^2\right]$$

$$+ \mathbb{E}\left[\left(f_{\hat{\beta}}(x) - \mathbb{E}[f_{\hat{\beta}}(x)]\right)^2\right] + \text{cross product terms}$$

$$= \sigma^2 + \text{Bias}^2 + \text{Var}(f_{\hat{\beta}}) + \text{cross product terms}$$

Note $\text{Var}(\varepsilon) = \mathbb{E}[\varepsilon - \mathbb{E}(\varepsilon)]^2 = \mathbb{E}(\varepsilon^2) = \sigma^2$

Briefly Look at the cross-product terms

$$\mathbb{E}\left[\varepsilon\,\underbrace{\left(g(x)-\mathbb{E}\left[f_{\hat{\beta}}(x)\right]\right)}_{\text{constant}}\right]$$

Recall

$\mathbb{E}(cZ)=c\,\mathbb{E}(Z)$

$= 0$

$$\mathbb{E}\left[\underbrace{\left(g(x)-\mathbb{E}\left[f_{\hat{\beta}}(x)\right]\right)}_{\text{constant}}\left(f_{\hat{\beta}}(x)-\overbrace{\mathbb{E}\left[f_{\hat{\beta}}(x)\right]}^{\text{constant}}\right)\right]$$

$= 0$

Recall $\mathbb{E}[Z-c]=\mathbb{E}(Z)-c$

$$\mathbb{E}\left[\varepsilon\left(f_{\hat{\beta}}(x)-\mathbb{E}\left[f_{\hat{\beta}}(x)\right]\right)\right]$$

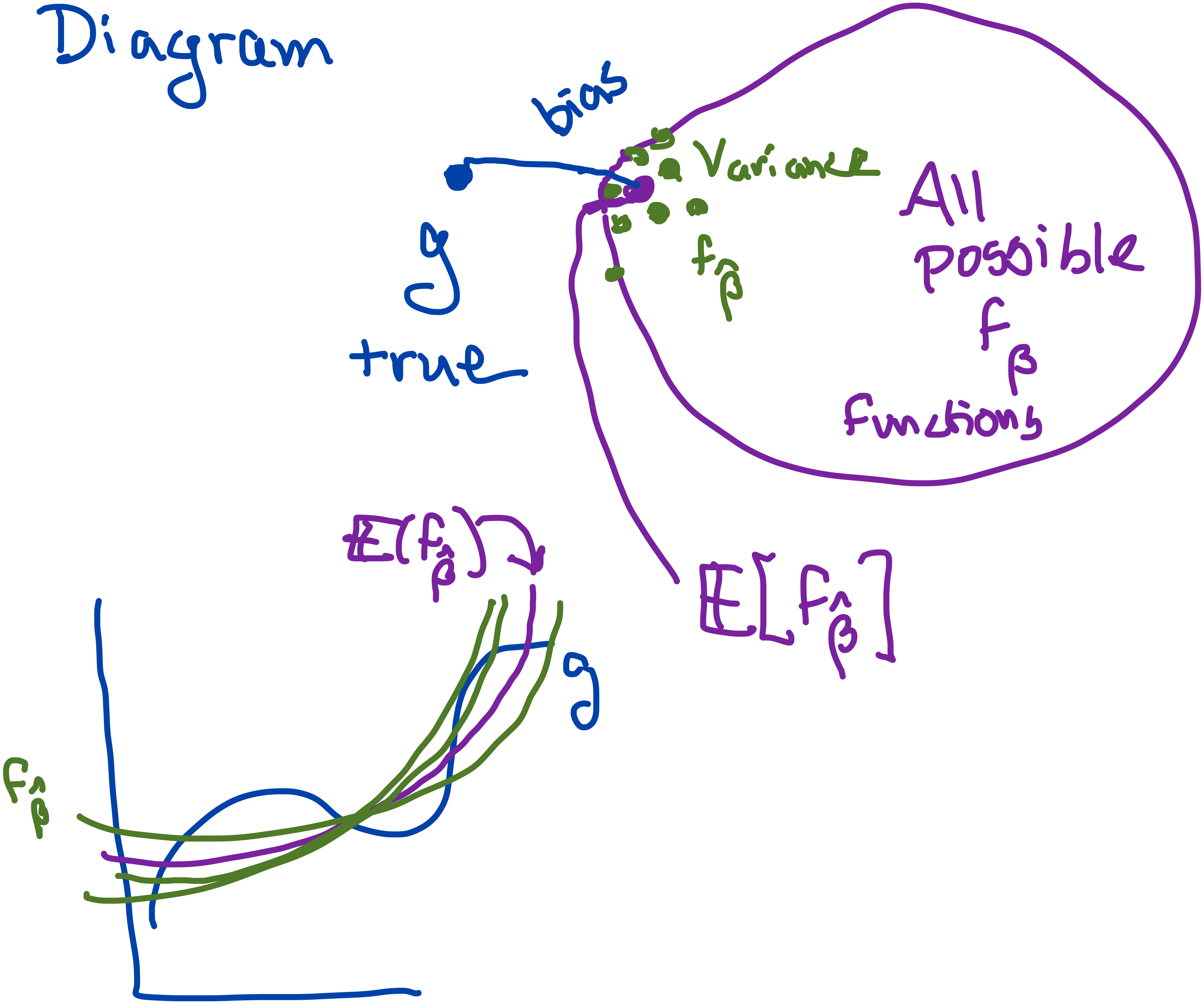
New propert: U and V independent random variables

then $\mathbb{E}[UV] = \mathbb{E}[U]\,\mathbb{E}[V]$

$$= \mathbb{E}[\varepsilon]\,\mathbb{E}\left[f_{\hat{\beta}}(x)-\mathbb{E}\left[f_{\hat{\beta}}(x)\right]\right]$$

$$= 0$$

Diagram


bias
g
true
Variance
$f_{\hat\beta}$
All possible f_β functions
$\mathbb{E}[f_{\hat\beta}]$
$\mathbb{E}(f_{\hat\beta})$
g
$f_{\hat\beta}$

$$\mathbb{E}(Y - f_{\hat{\beta}}(x))^2 =$$

$$\sigma^2$$

Obs. Variance

“Noise”

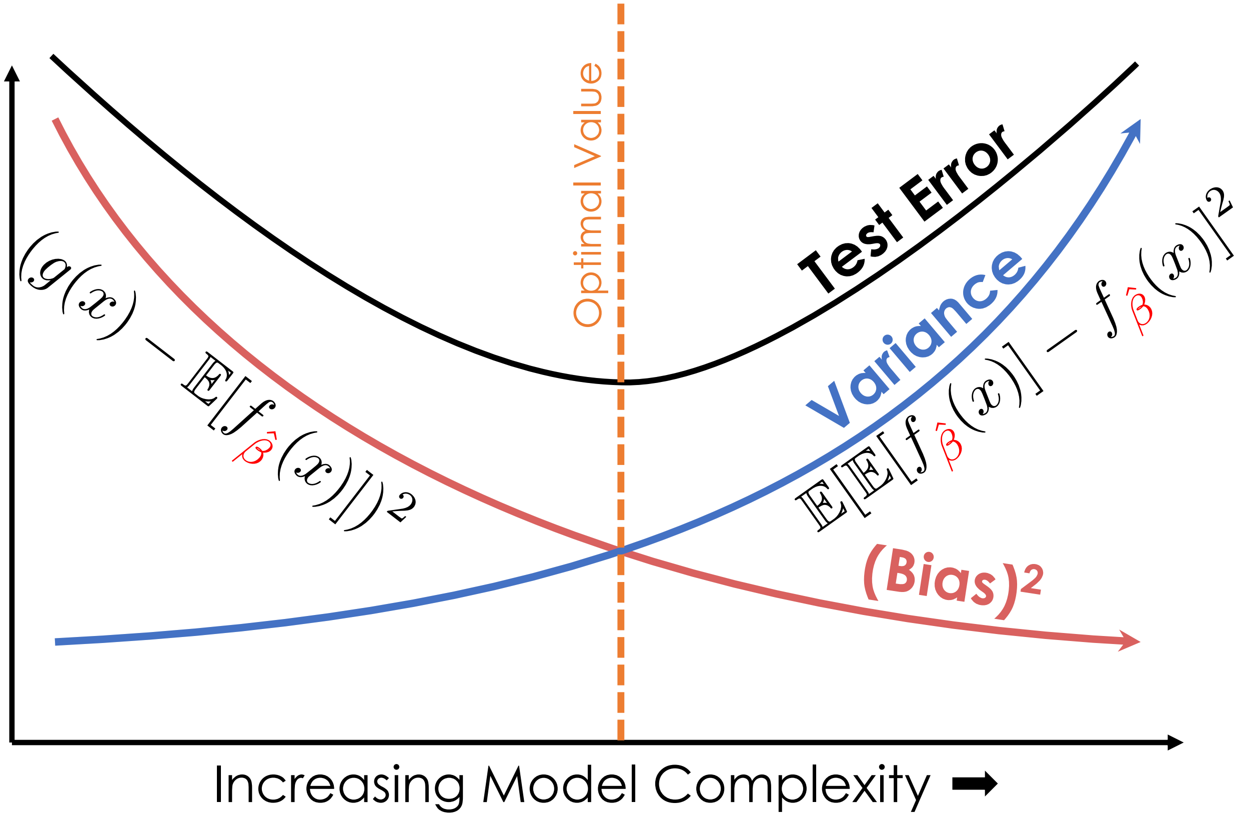
$$+ \quad (g(x) - \mathbb{E}[f_{\hat{\beta}}(x)])^2$$

(Bias)²

$$+ \ \mathbb{E}[\mathbb{E}[f_{\hat{\beta}}(x)] - f_{\hat{\beta}}(x)]^2$$

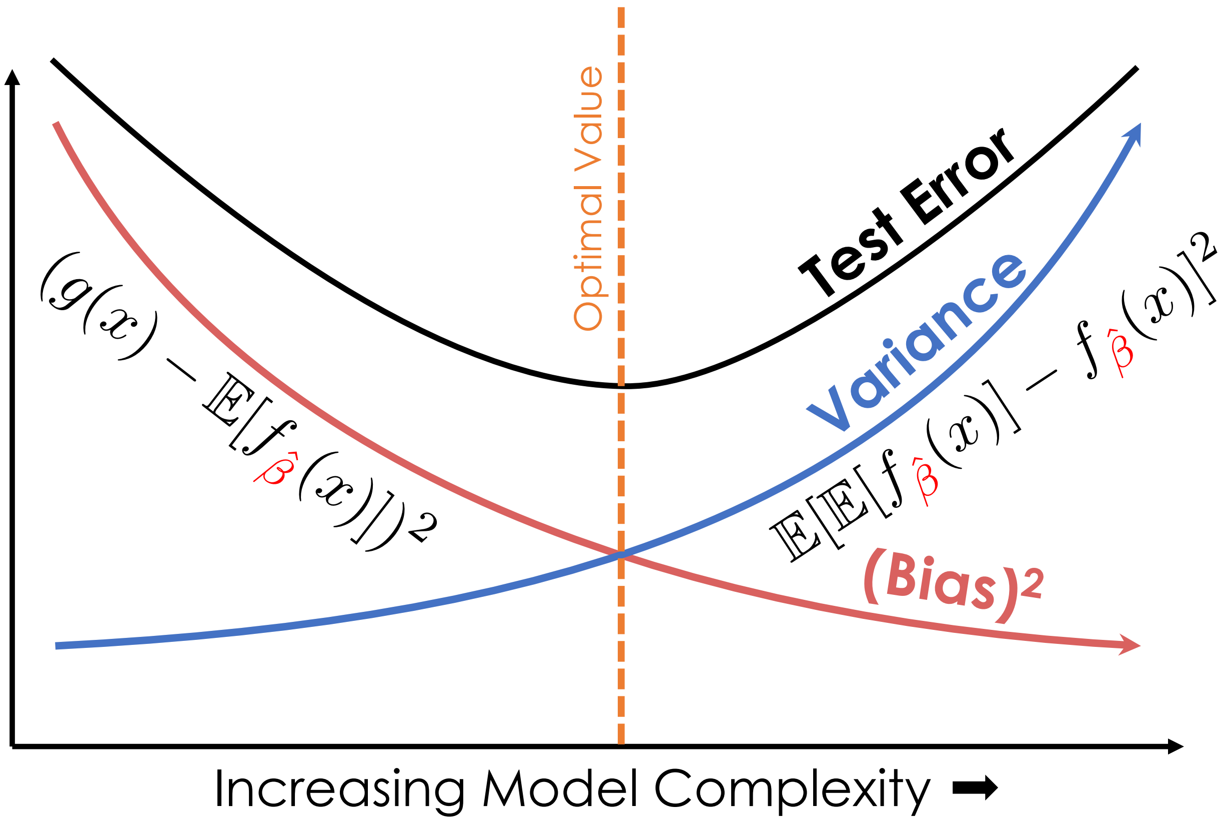
Model Variance

Bias Variance Plot



How do we control **model complexity**?

- So far:
  - Number of features
  - Choices of features
- **Next: Regularization**



Bias Variance Derivation Quiz

- Match each of the following:

(1) $\mathbb{E}(Y)$

(2) $\mathbb{E}(\epsilon^2)$

(3) $(g(x) - \mathbb{E}[f_{\hat{\beta}}(x)])^2$

(4) $\mathbb{E}(\epsilon(g(x) - f_{\hat{\beta}}(x))$

A. 0

B. Bias^2

C. Model Variance

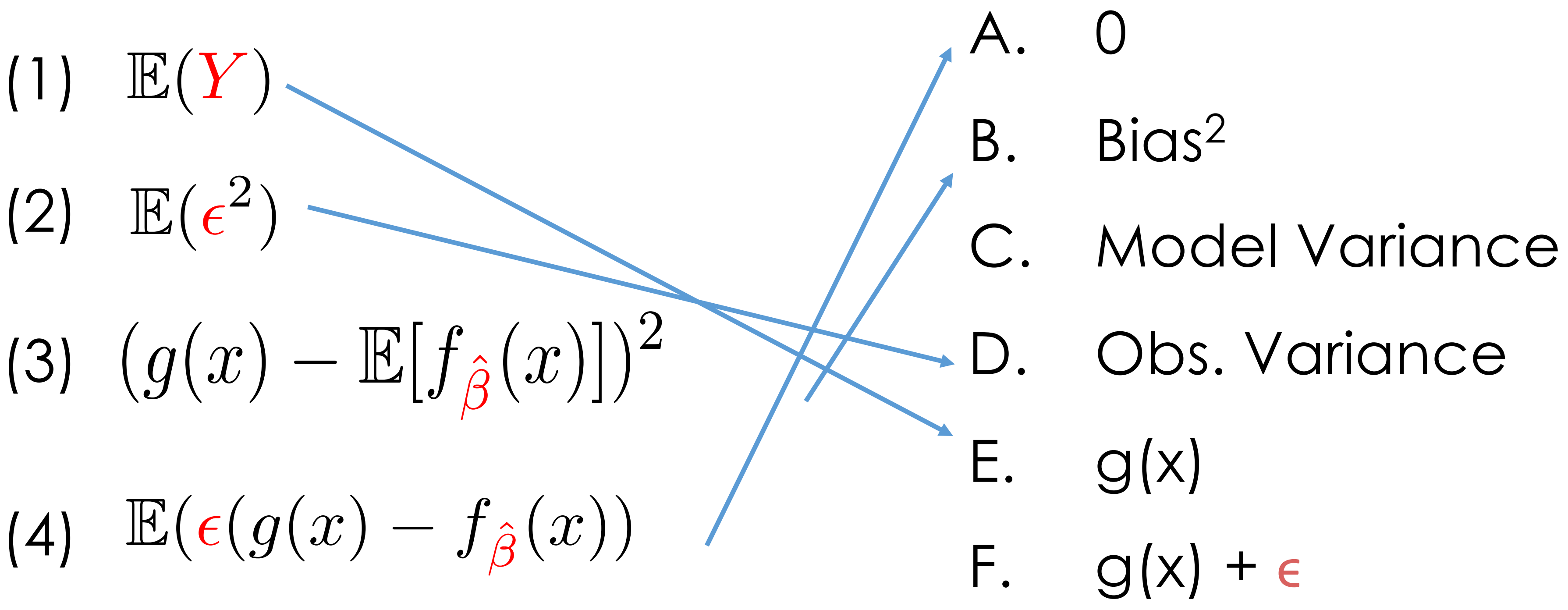
D. Obs. Variance

E. g(x)

F. g(x) + ϵ

Bias Variance Derivation Quiz

- Match each of the following:



(1) $\mathbb{E}(Y)$

(2) $\mathbb{E}(\epsilon^2)$

(3) $(g(x) - \mathbb{E}[f_{\hat{\beta}}(x)])^2$

(4) $\mathbb{E}(\epsilon(g(x) - f_{\hat{\beta}}(x))$

A. 0

B. Bias^2

C. Model Variance

D. Obs. Variance

E. g(x)

F. g(x) + ϵ

Regularization

Parametrically Controlling the *Model Complexity*

- Tradeoff:
  - **Increase bias**
  - **Decrease variance**

Basic Idea of Regularization

Adjust all of the $\hat{\beta}_j$

Make them closer to 0

Regularization is
AKA
Shrinkage

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \mathrm{LOSS}(y_i, f_{\beta}(x_i))$$

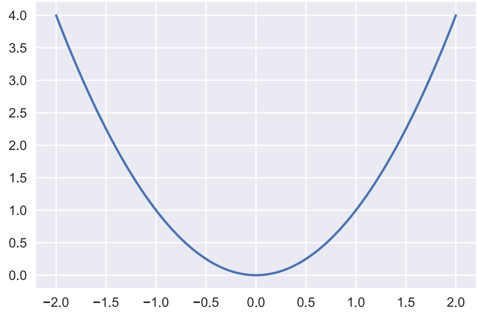
$$\beta : \mathcal{S}(\beta) \leq c$$

$\mathcal{S}(\beta)$ measures the size of the coefficients

Common Regularization Functions

Ridge Regression (L2-Reg)

$$\mathcal{S}(\beta) = \sum_{j=1}^{p} \beta_j^2$$



- Distributes weight across related features (robust)
- Analytic solution (easy to compute)
- Does not encourage sparsity → small but non-zero weights.

LASSO (L1-Reg)

$$\mathcal{S}(\beta) = \sum_{j=1}^{p} |\beta_j|$$

- **Encourages sparsity** by setting weights = 0
  - Used to select informative features
- Does not have an analytic solution → numerical methods

Standardization and the **Intercept** Term

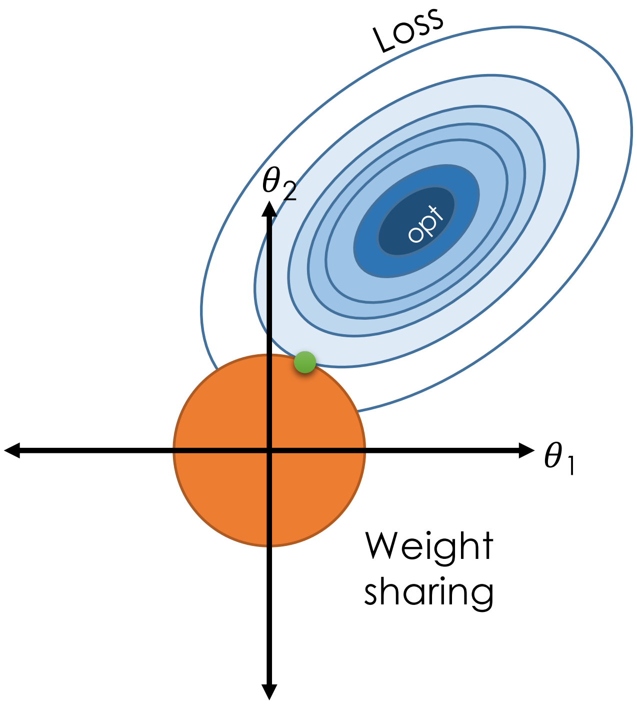
- Regularization penalized dimensions equally
- **Standardize features**
  - Ensure that each dimensions has the same scale
  - centered around zero
- **Intercept Terms**
  - Don't regularize intercept term ← **Suggested**
  - Center *y* values (e.g., subtract mean)

Standardization

For each dimension *k*:

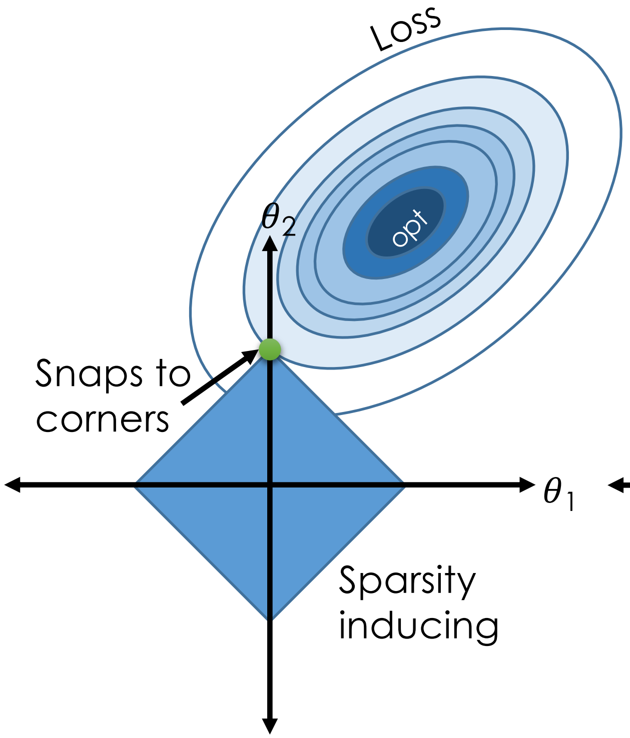
$$z_k = \frac{x_k - \mu_k}{\sigma_k}$$

Regularization and Norm Balls



Weight sharing

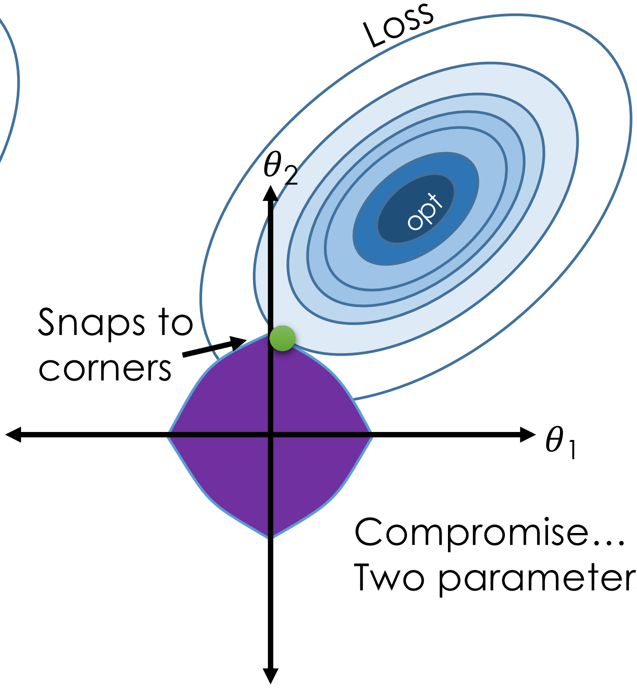
L2 Norm (Ridge)



Snaps to corners

Sparsity inducing

L1 Norm (LASSO)



Snaps to corners

Compromise…
Two parameters …

L1 + L2 Norm
(Elastic Net)

Equivalent Representation

Fit the Data

Penalize Complex Models

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \mathrm{LOSS}(y_i, f_{\beta}(x_i)) + \lambda \mathcal{S}(\beta)$$

Regularization Parameter

- How do we determine λ

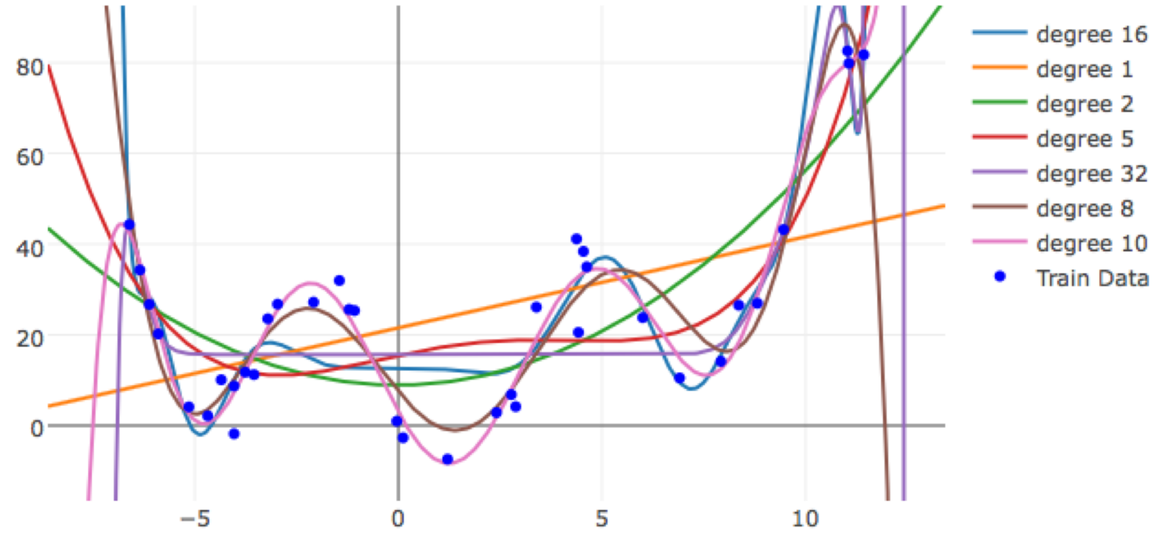
Note we can also minimize w/out the 1/n. This is just a rescaling of λ

The Regularization Function $\mathcal{S}(\beta)$

Goal: *Penalize model complexity*

Recall earlier: $\phi(x) = \left[x, x^2, x^3, \ldots, x^p\right]$

- More features → overfitting …
- How can we control overfitting through β



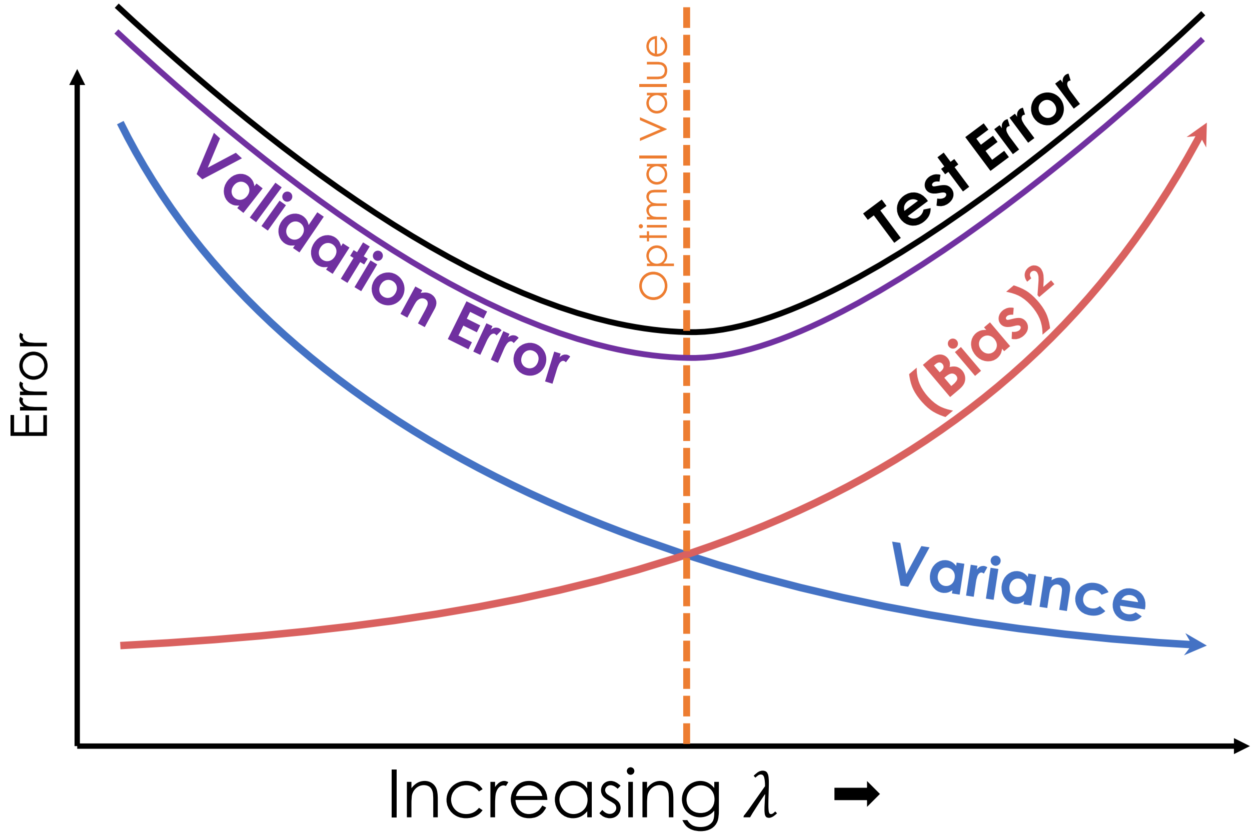
Determining the Optimal λ

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \text{LOSS}(y_i, f_\beta(x_i)) \; + \lambda \mathcal{S}(\beta)$$

- Value of λ determines bias-variance tradeoff
  - Larger values → more regularization → more bias → less variance

Determining the Optimal λ

How do we determine λ?



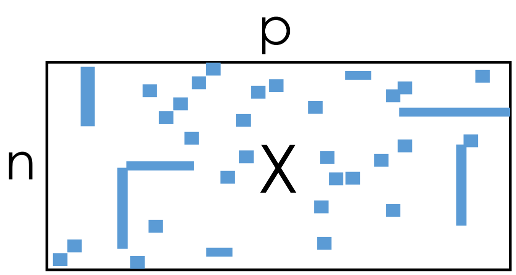
- Value of λ determines bias-variance tradeoff
  - Larger values → more regularization → more bias → less variance
- Determined through cross validation

Regularization and **High-Dimensional Data**

Regularization is often used with high-dimensional data

Tall Skinny Matrix

- n >> p
- typically dense
- Regularization can help with complex feature transformations

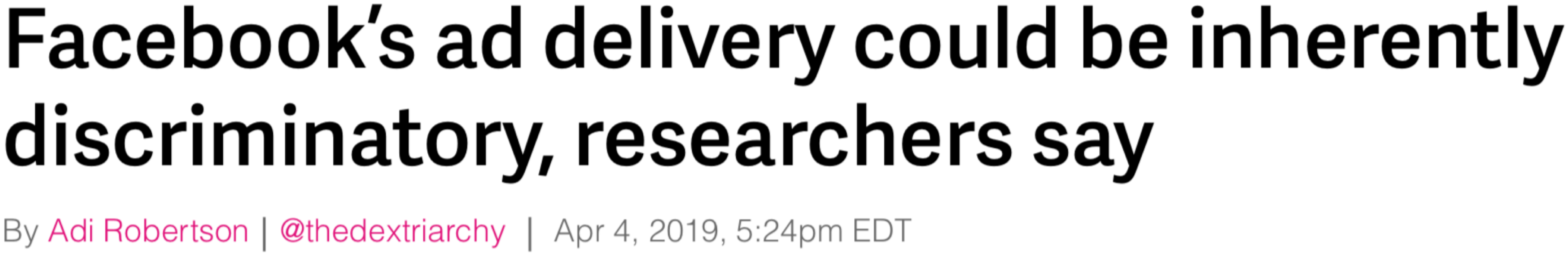


High-dimensional sparse matrix

- p > n
- **requires** regularization
- **Goal:** to determine **informative dimensions**
  - Consider L1 Regularization. Why?
- **Goal:** is to make **robust predictions**
  - Consider L2 (+L1) Regularization

Modeling is hard, especially when you have tons of features

Why Stanford Researchers Tried to Create a 'Gaydar' Machine



Facebook's ad delivery could be inherently discriminatory, researchers say

By Adi Robertson | @thedextriarchy | Apr 4, 2019, 5:24pm EDT