Welcome to CME 250 Introduction to Machine Learning!

Spring 2020 – Online version

April 23th 2020

Stanford | Institute for Computational & Mathematical Engineering

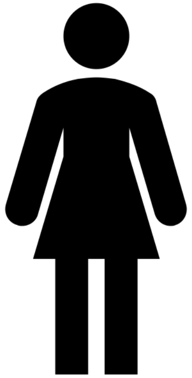
Today’s schedule

“ML magic”

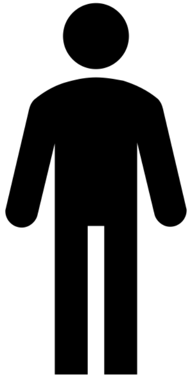
- Evaluating Performance: Model Selection
  - Randomness in our data
  - Model Complexity
  - Sample size
  - Bias - Variance Tradeoff
- Reducing Model complexity: Regularization
- Computing expected error
  - Training – Validation - Test set
  - K-fold cross validation

Let's get to know each other…

Breakout room



You



Another student

Name

Location

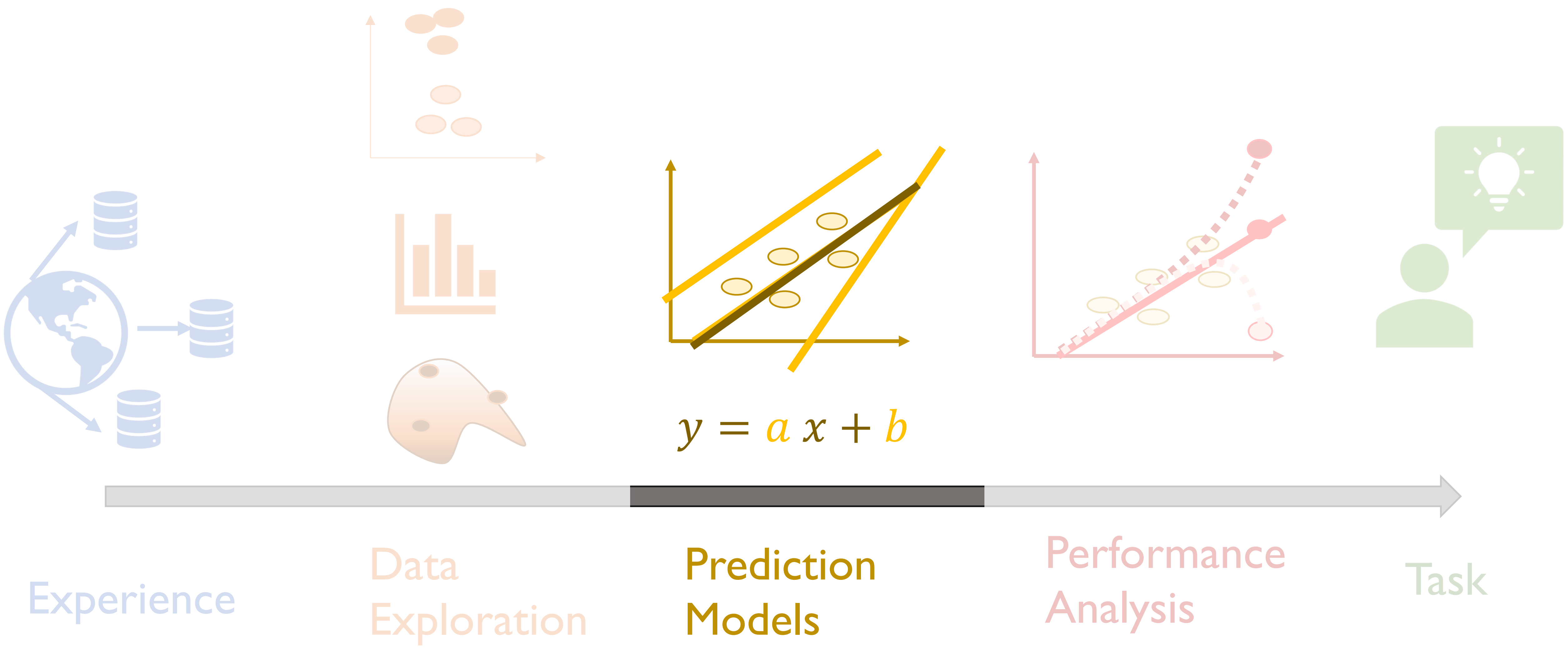
Department

Year

Have you discovered a new recipe / delivery restaurant?

3 mins

Chat/Audio/Video



$y = a\,x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Recap last class: Supervised Learning

"Learn by example"

Features

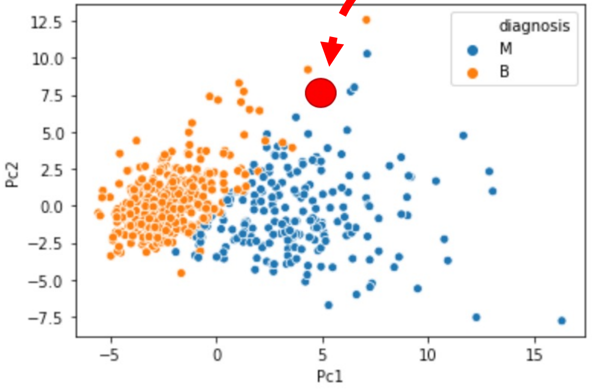
Sample

Given	Inputs	Outputs
	$\left(x_1^{(1)}, x_2^{(1)}, \dots, x_{p-1}^{(1)}, x_p^{(1)}\right) \xrightarrow{\hat{f}} y^{(1)}$	$y^{(1)}$
	$\left(x_1^{(2)}, x_2^{(2)}, \dots, x_{p-1}^{(2)}, x_p^{(2)}\right) \longrightarrow y^{(2)}$	$y^{(2)}$
	…	
	$\left(x_1^{(n)}, x_2^{(n)}, \dots, x_{p-1}^{(n)}, x_p^{(n)}\right) \longrightarrow y^{(n)}$	$y^{(n)}$

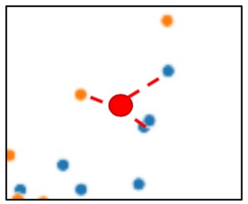
Categorical **Classification**

Quantitative **Regression**

We want to predict $\left(x_1, x_2, \dots, x_{p-1}, x_p\right) \xrightarrow{\hat{f}}$?



K-Nearest Neighbors



Non-parametric

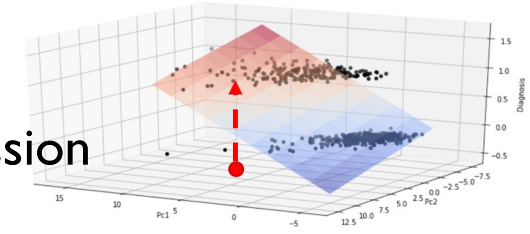
Fit local model for each data

How many neighbors?

Inference

Prediction

Linear Regression



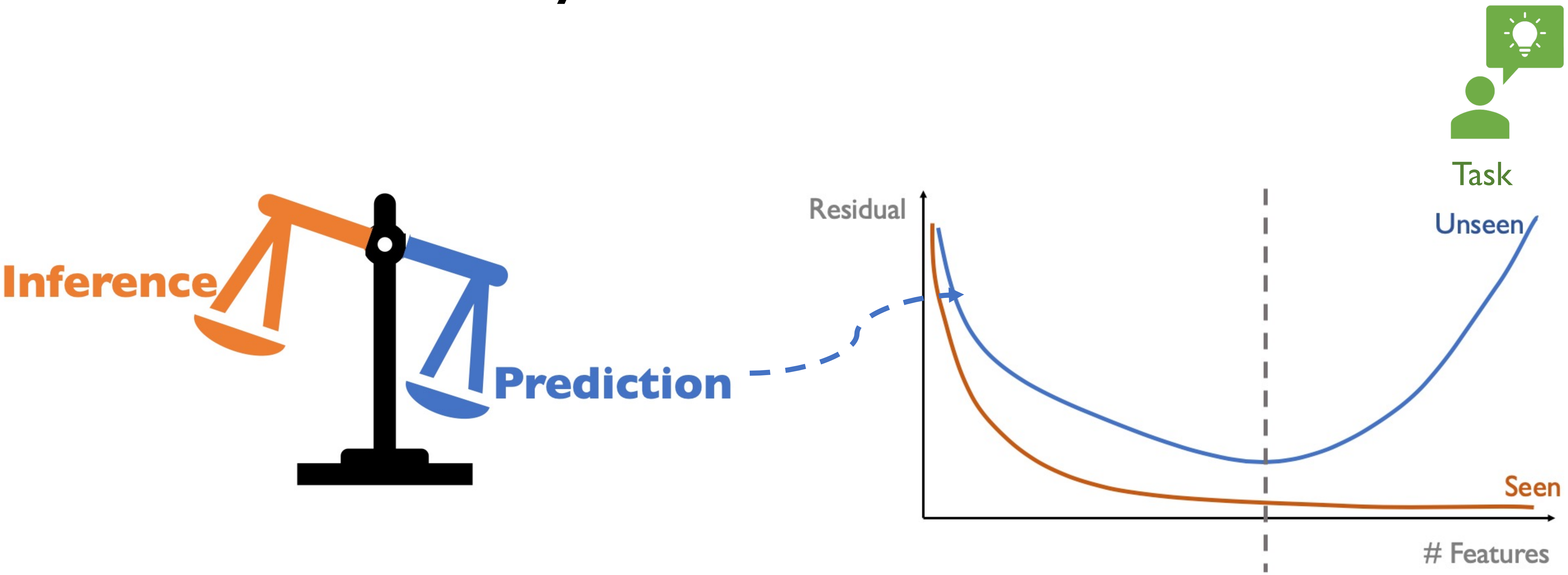
Parametric

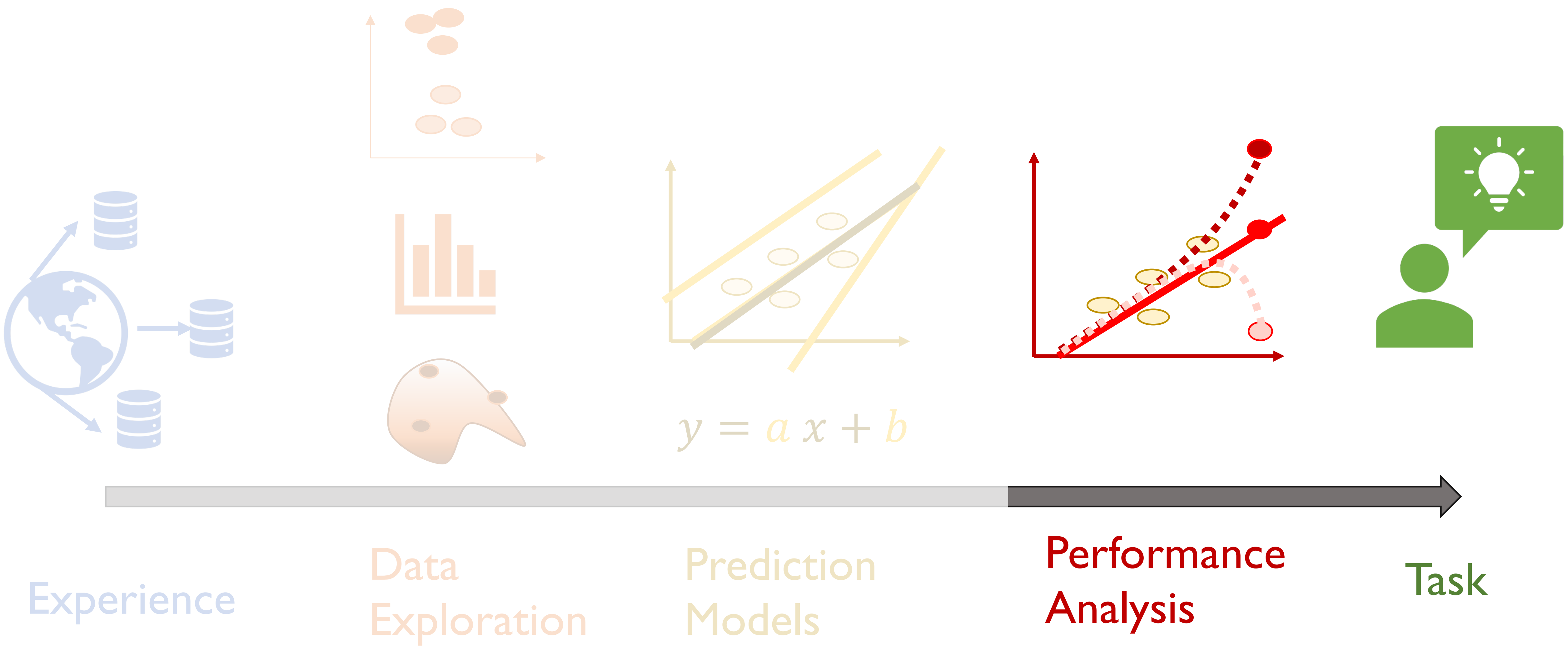
Fit parameters of model for all the data

$$E[Y|X] = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$$

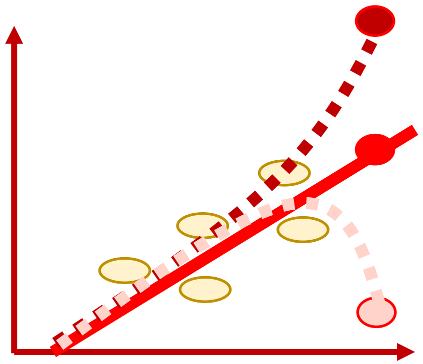
How many features?

Motivation of today's class




$y = a x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Model Assessment and Selection



Performance Analysis

Introduction to Statistical Learning

Chapter 5.1: Cross Validation

Chapter 6: Regularization

Elements Statistical Learning

Chapter 7.1-7.3: Bias vs Variance

Chapter 7.10: Cross Validation

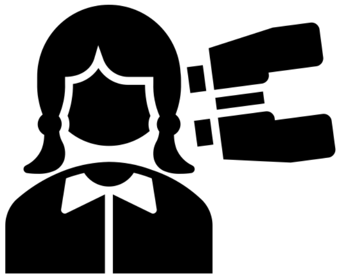
Chapter 3.4: Regularization

Randomness: Seen and unseen data are different

1) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict

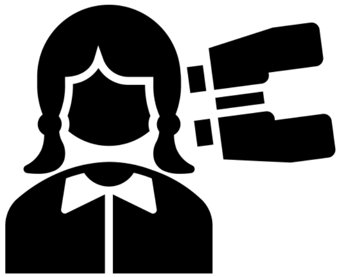


Randomness: Seen and unseen data are different

1) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict

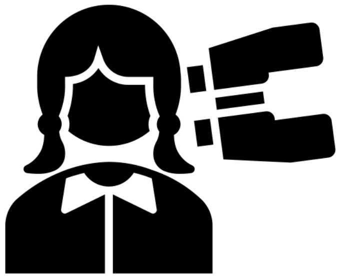


Randomness: Seen and unseen data are different

1) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict

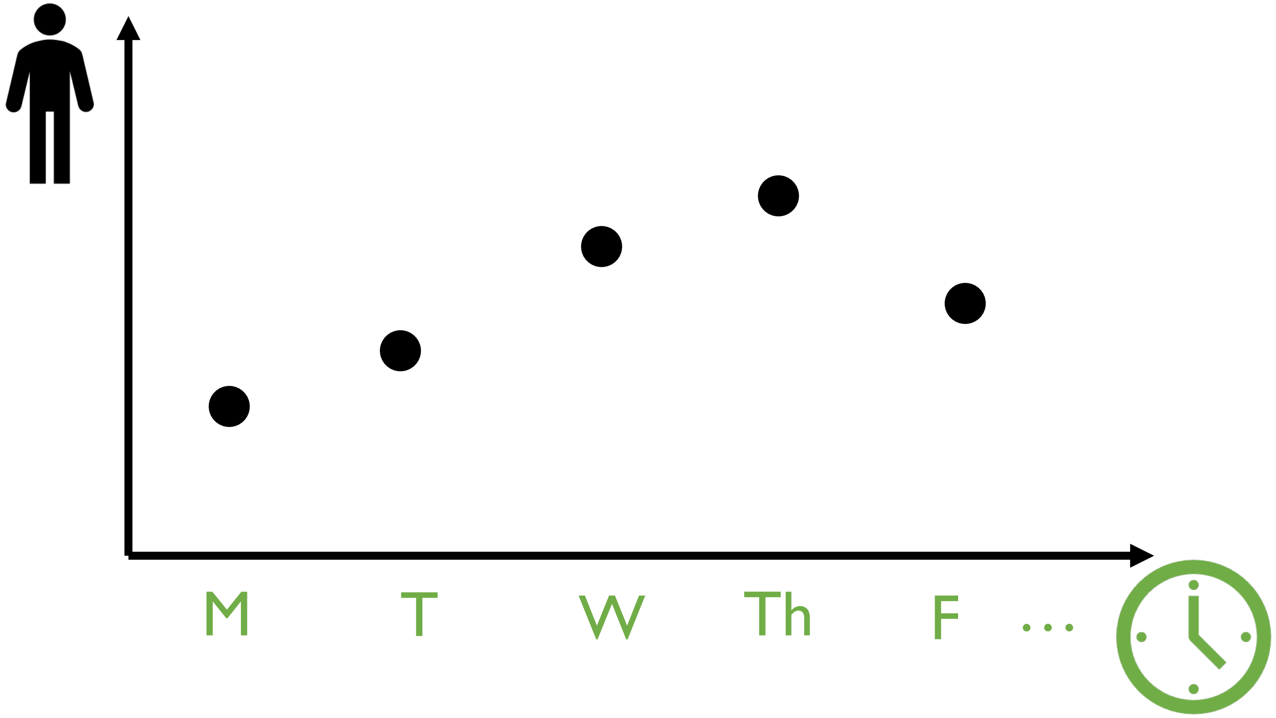


Collecting the samples

1) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict

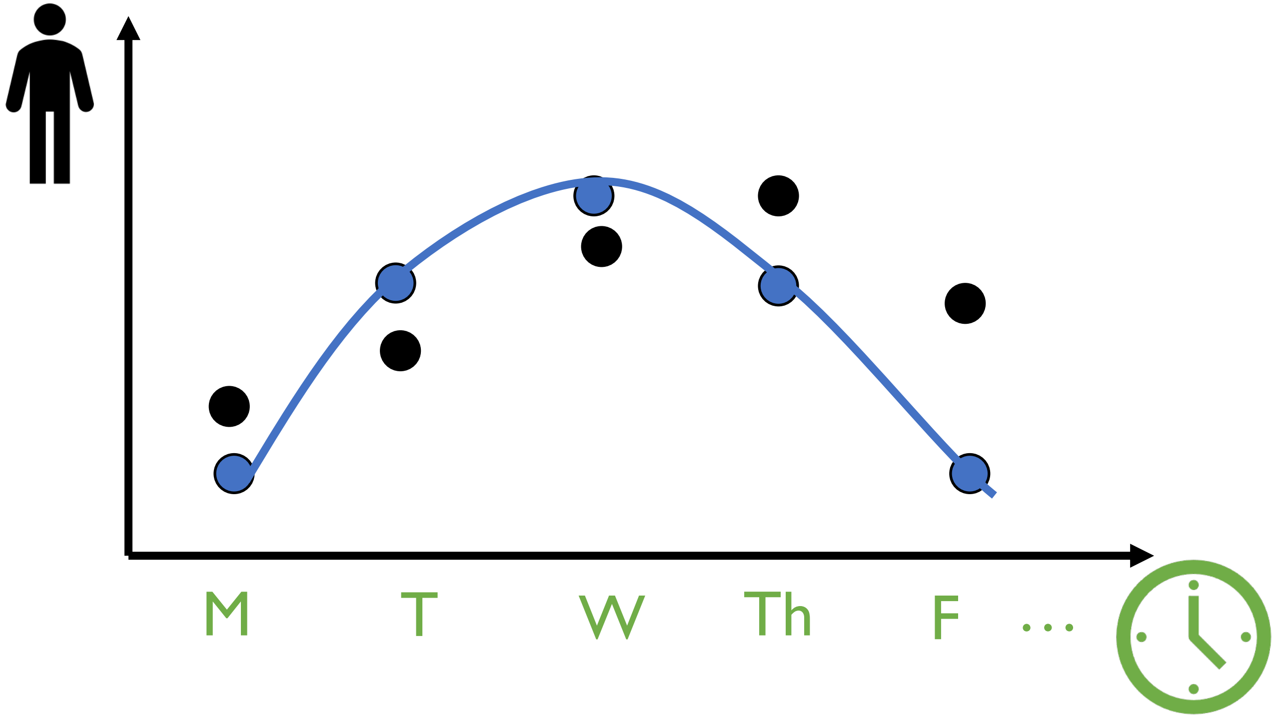


Collecting the samples

I) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict

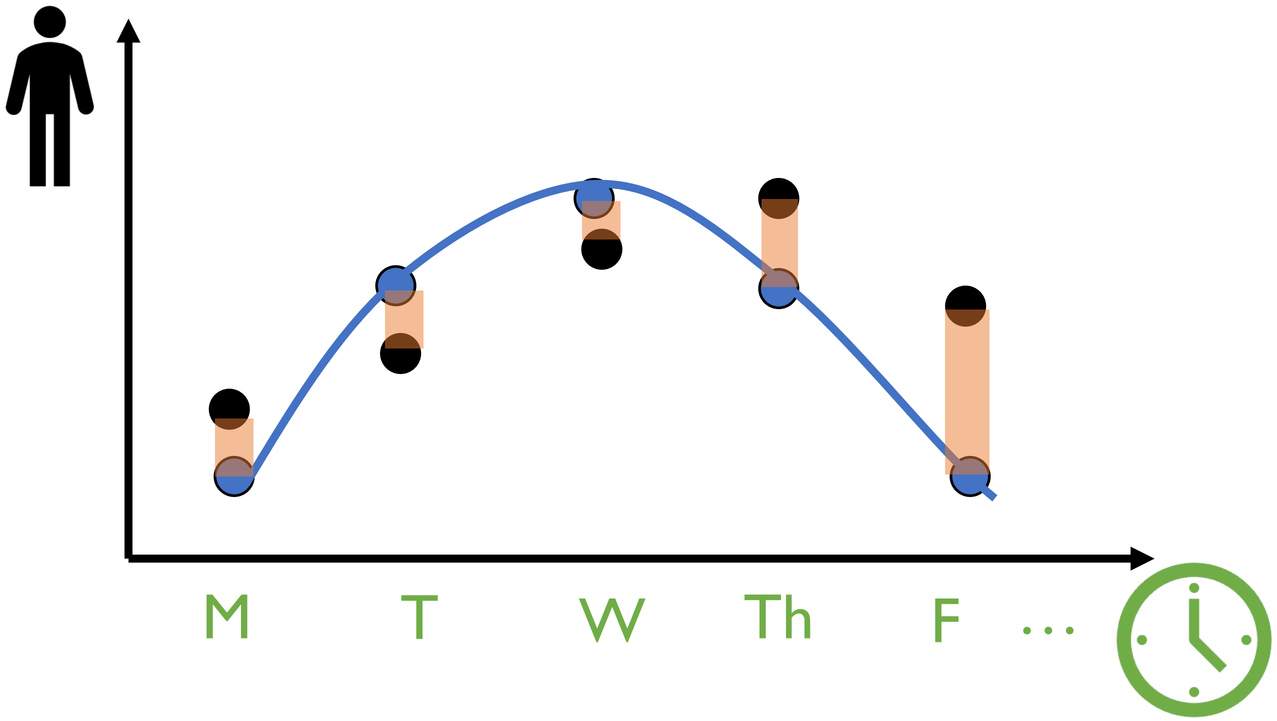


Collecting the samples

1) How the data is: $Y = f(X) + \varepsilon$

We intend to predict ← deterministic

random → We cannot predict



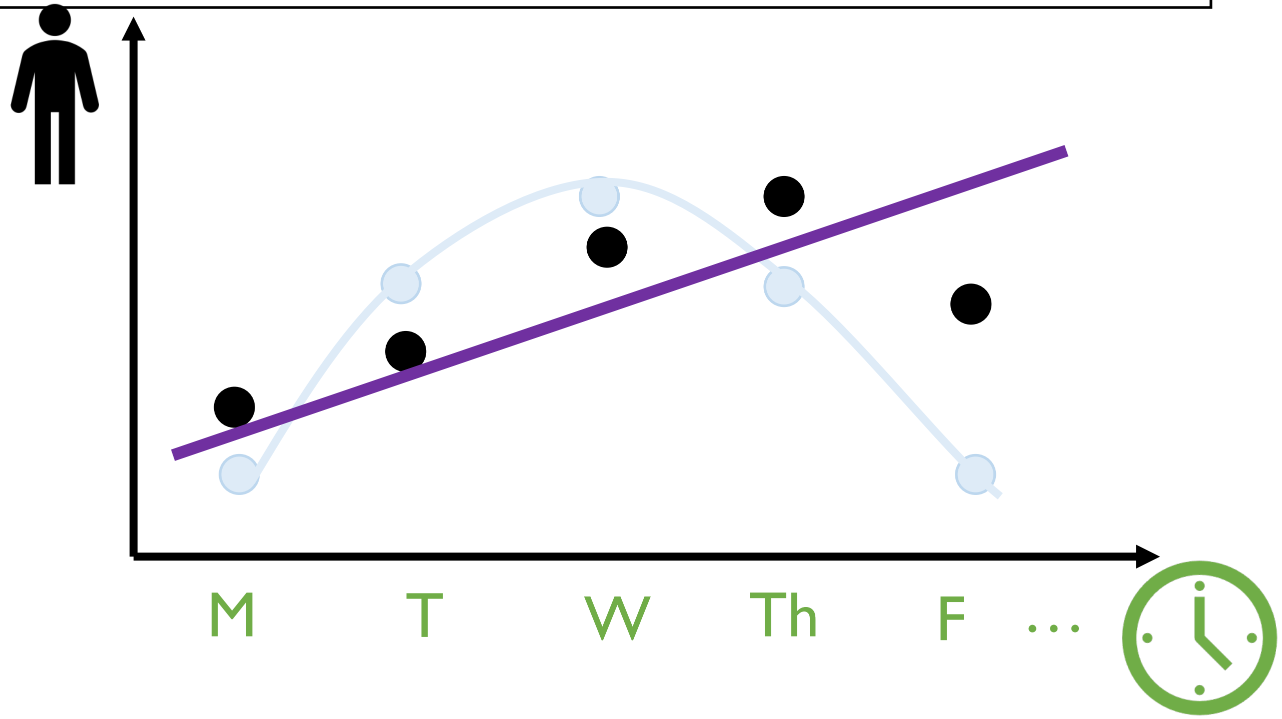
We can only see $y^{(i)}$

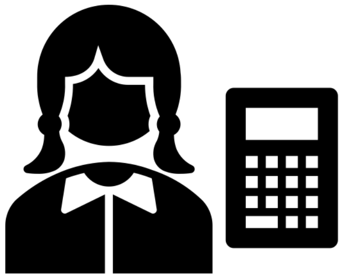
We don't know $f(x^{(i)})$ nor $\varepsilon^{(i)}$

Choosing a model

1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose a model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$



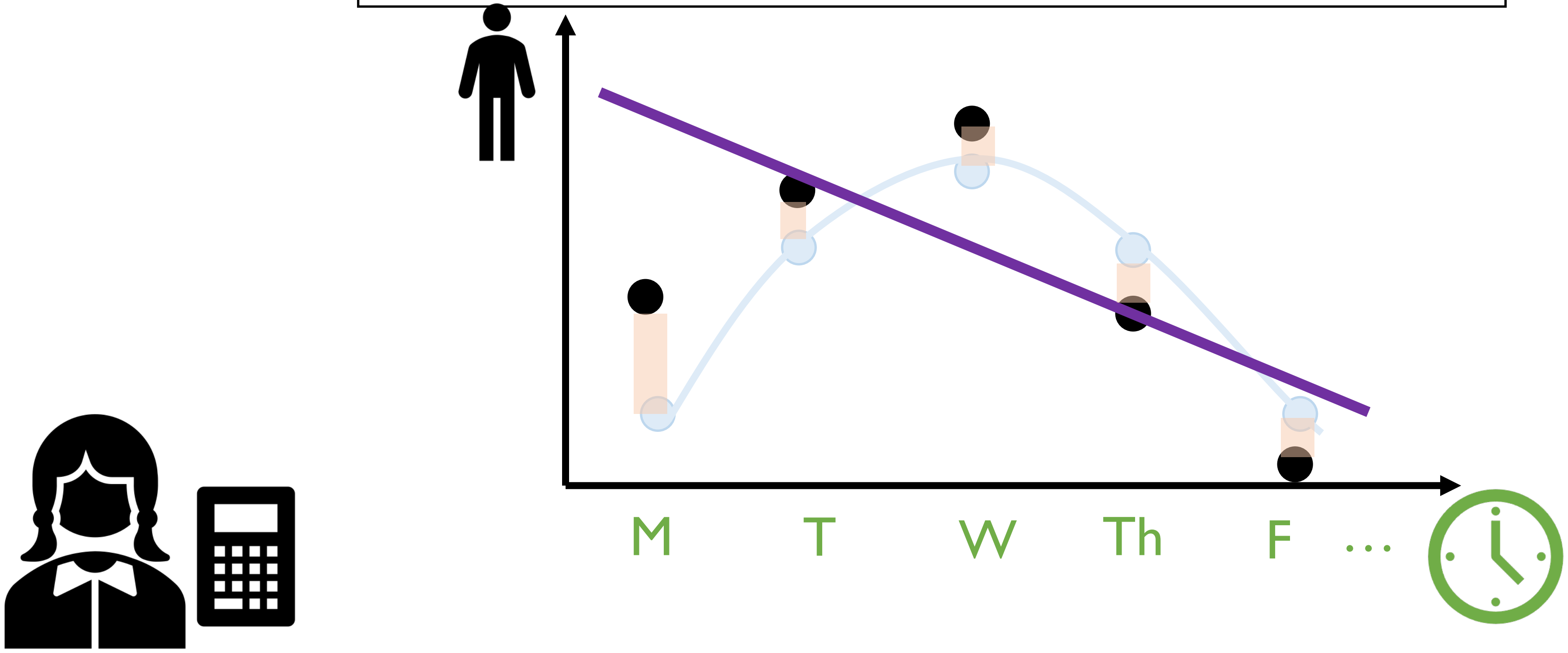


We compute the model using our observations

The model is random!

1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose a model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$

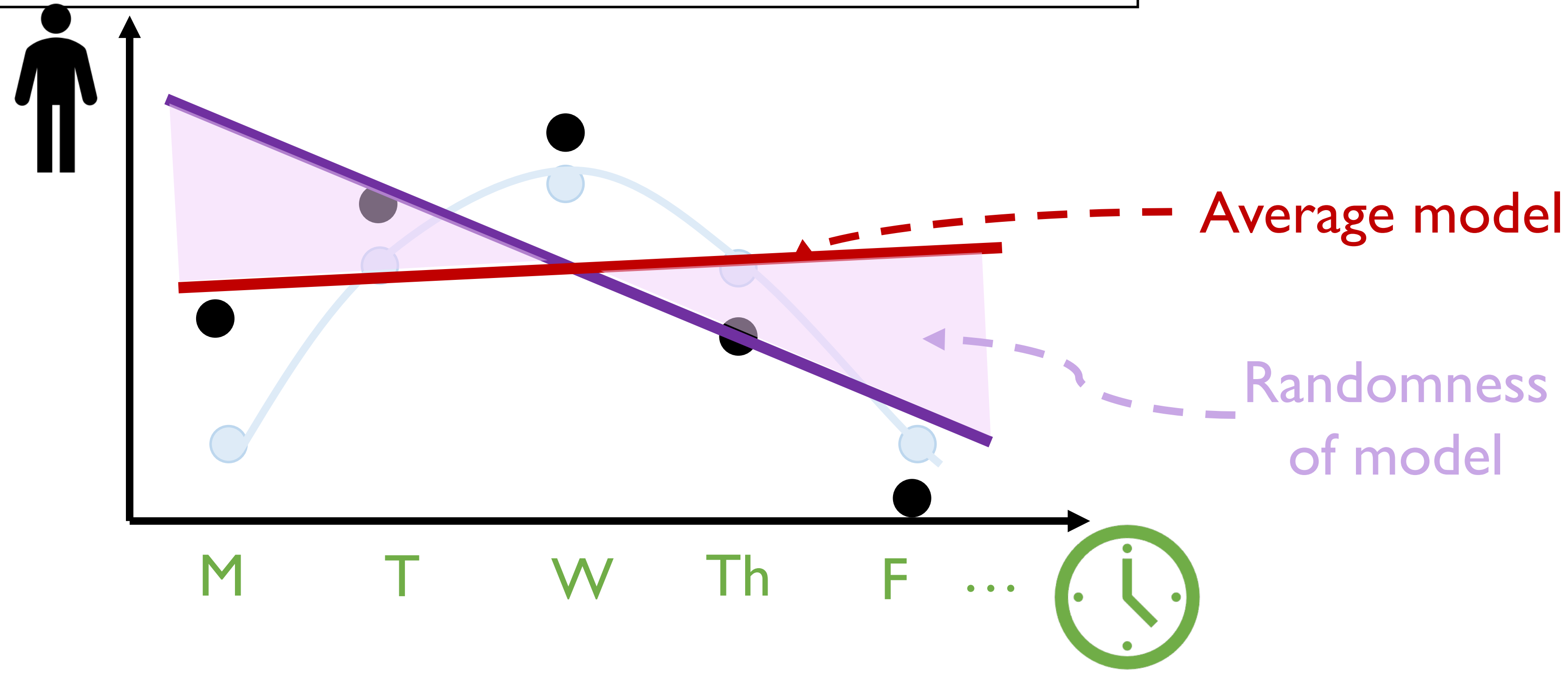


With a different set of observations we get a different model

The model is random!

1) How the data is: $Y = f(X) + \varepsilon$

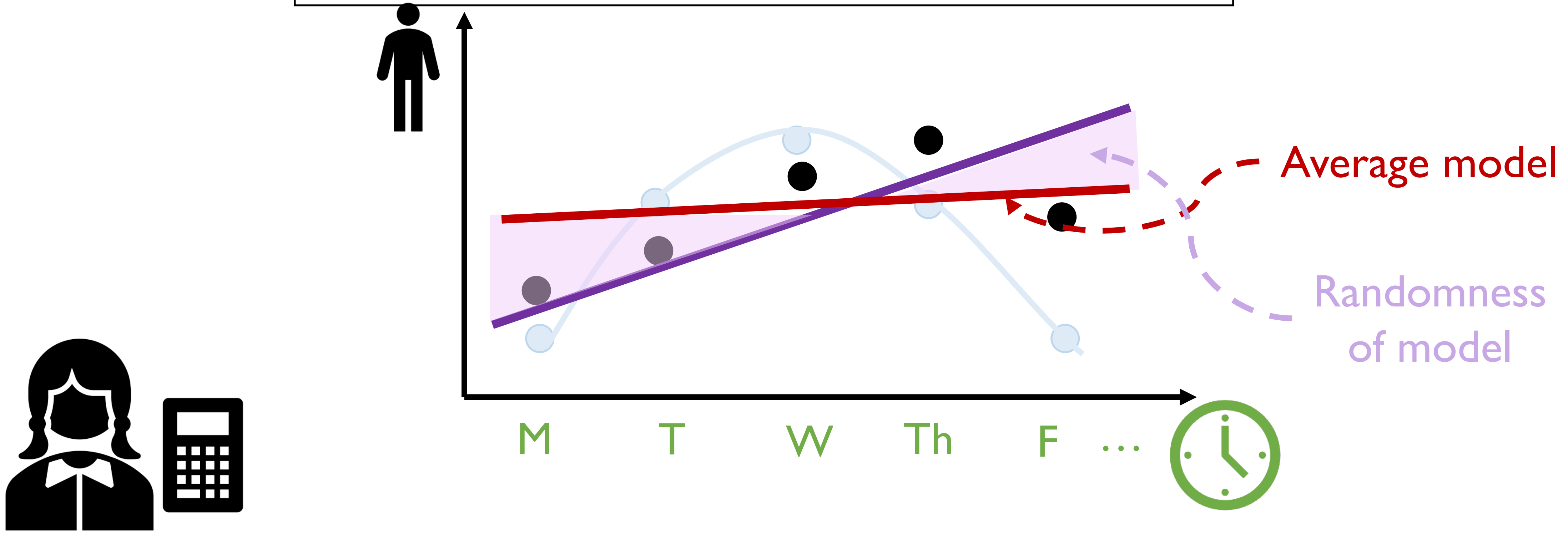
2) We suppose a model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1}X$



The model is random!

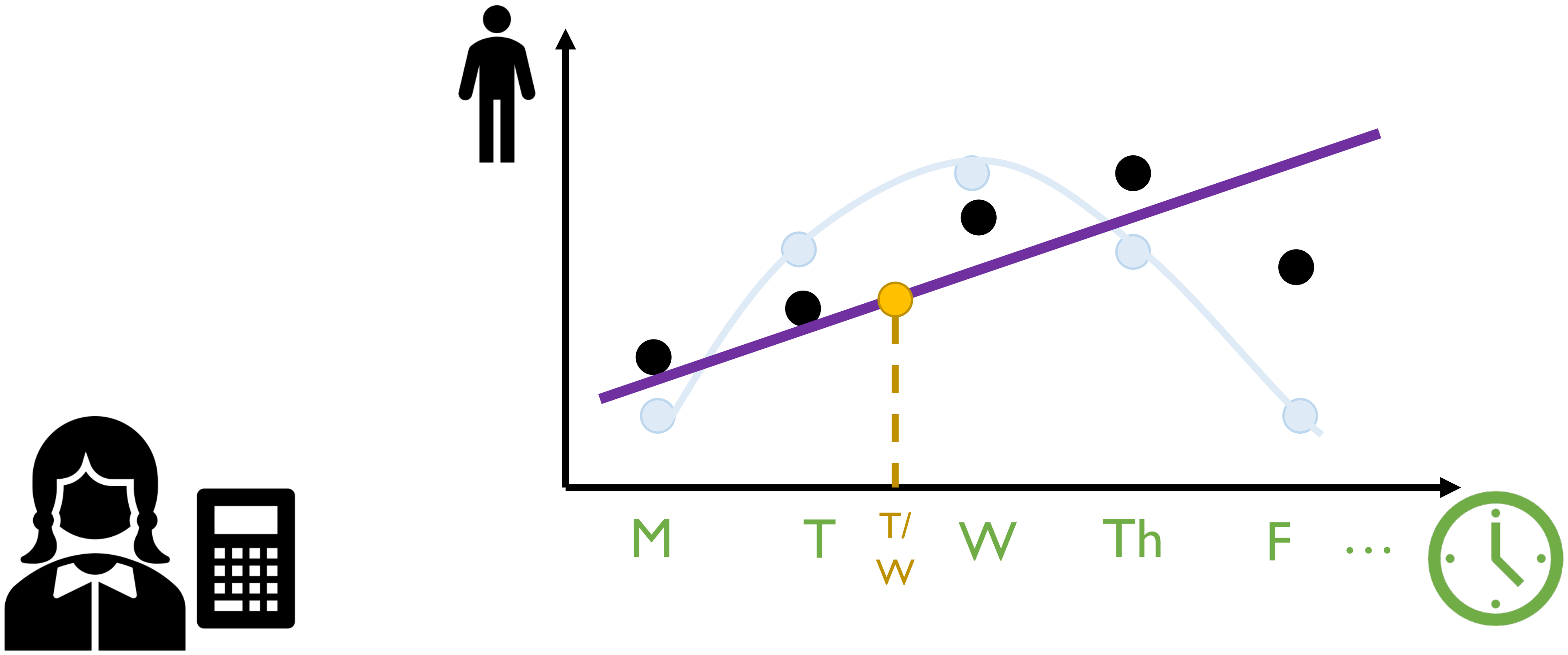
1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose a model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$



Prediction error: Unseen data

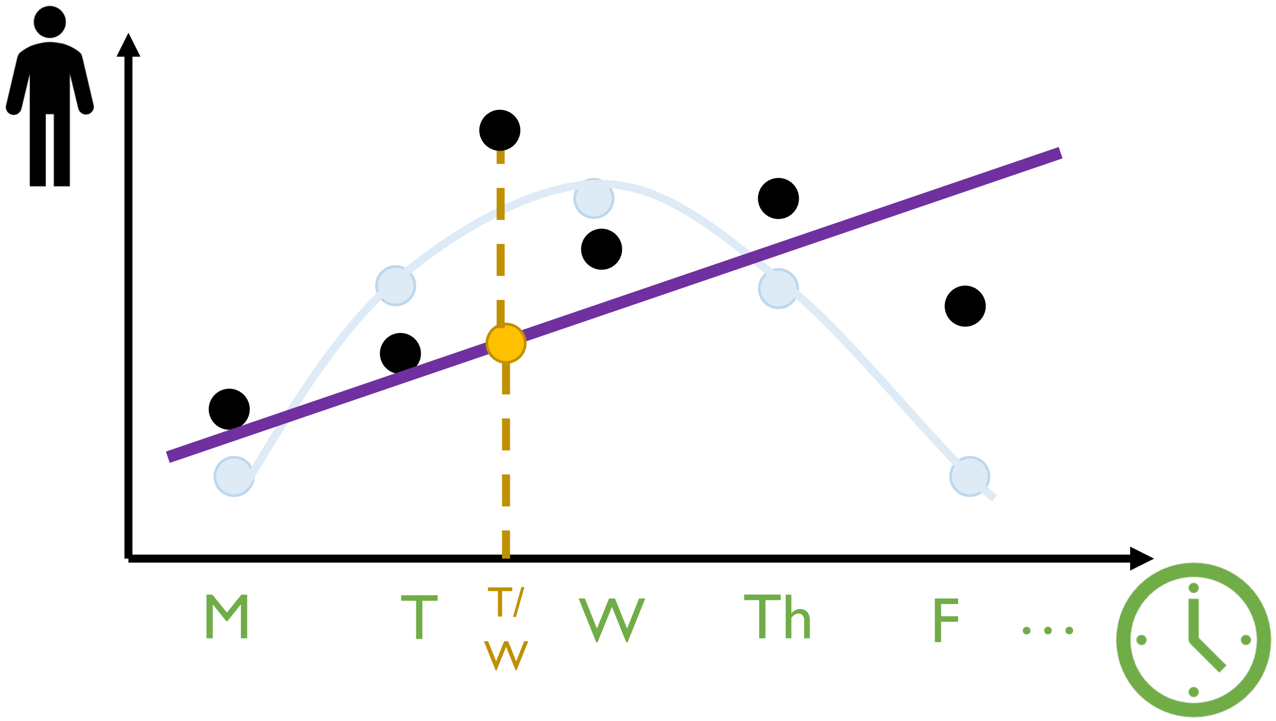
1) How the data is:	$Y = f(X) + \varepsilon$
2) We suppose model:	$\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$
2) We predict new data:	$Y \approx \hat{f}(X)$



How accurate is this prediction?

Prediction error: Unseen data

1) How the data is:	$Y = f(X) + \varepsilon$
2) We suppose model:	$\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$
2) We predict new data:	$Y \approx \hat{f}(X)$



How accurate is this prediction?

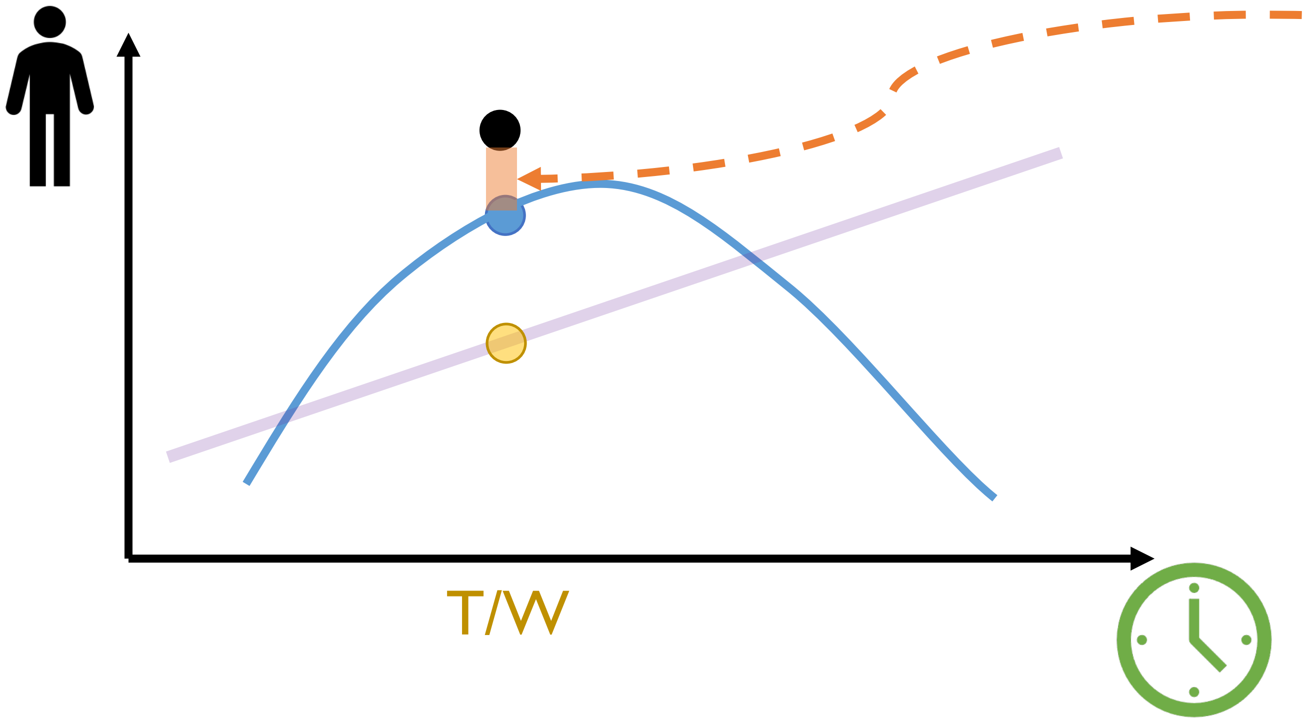
Is it the same as measuring y?

Sources of error: Irreducible error

1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$

2) We predict new data: $Y \approx \hat{f}(X)$



We cannot know ε
randomness of new observation

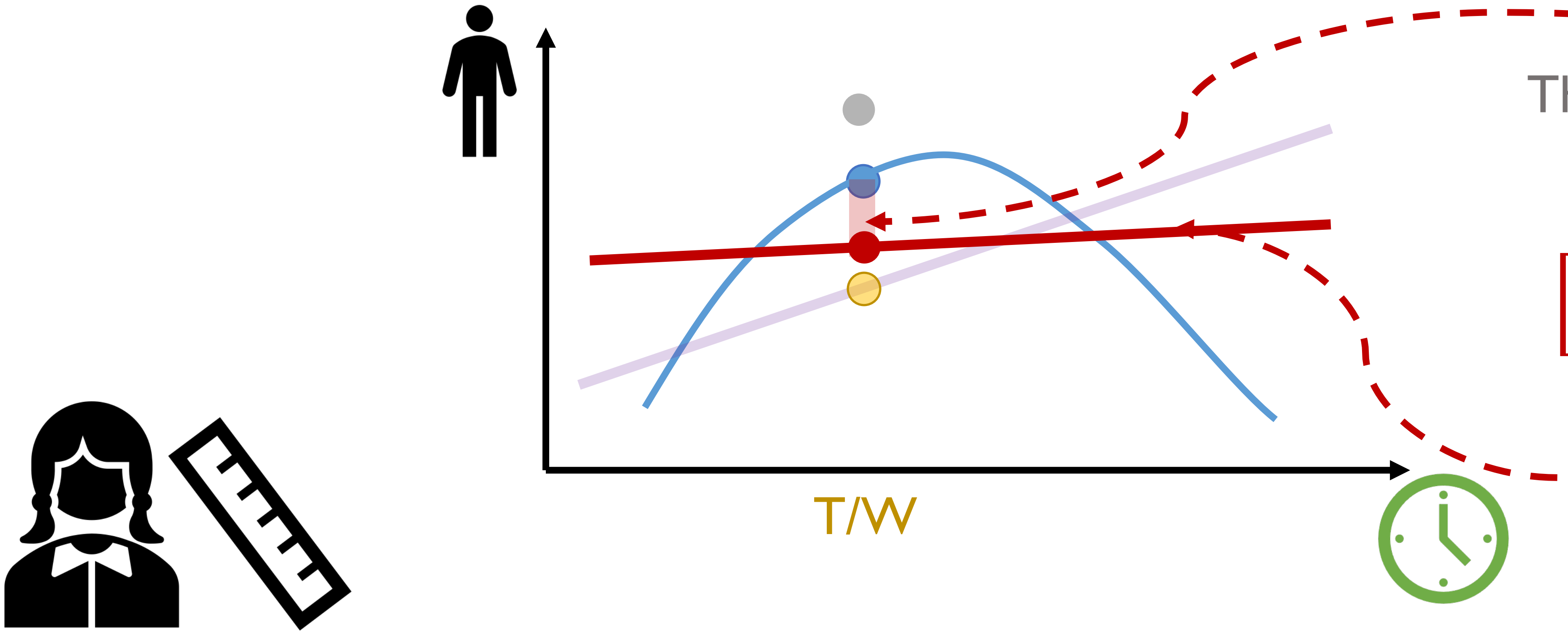
$$E[Y - f(x)]^2$$

Sources of error: Bias

1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$

2) We predict new data: $Y \approx \hat{f}(X)$



The error in assuming the wrong model

$$\left[f(x) - E\left[\hat{f}(x)\right]\right]^2$$

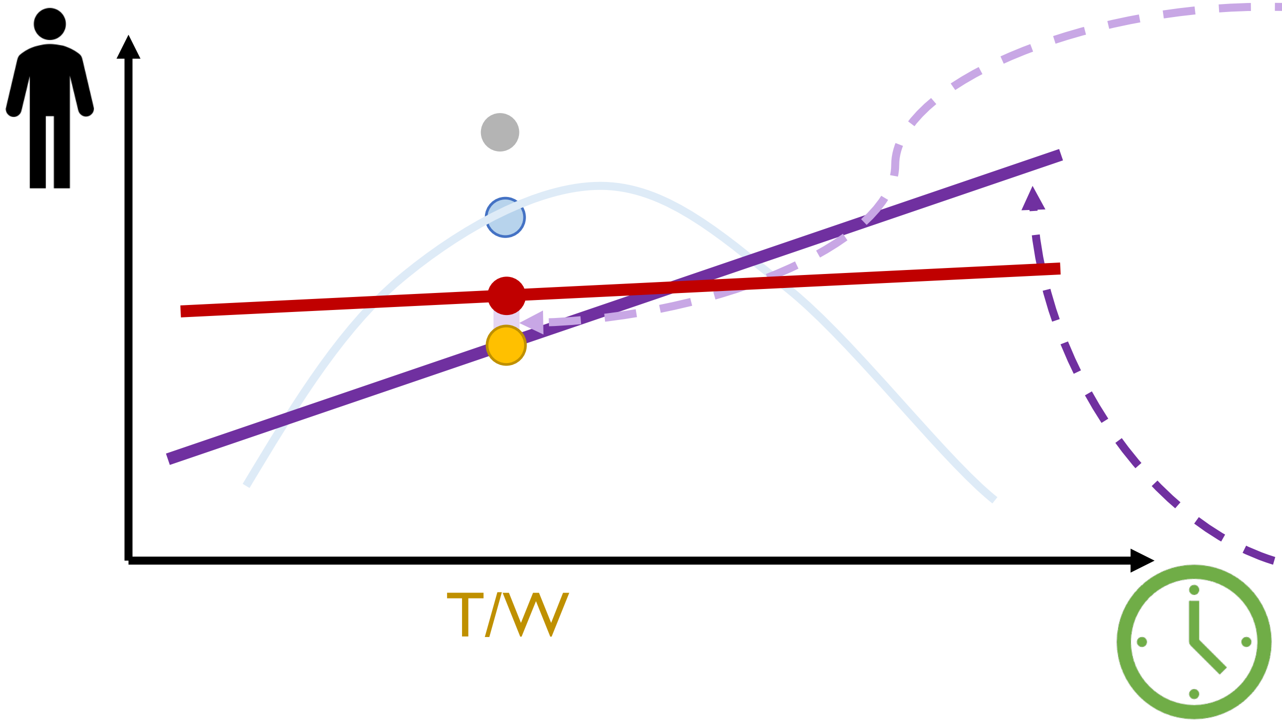
Average model

Sources of error: Variance

1) How the data is: $Y = f(X) + \varepsilon$

2) We suppose model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$

2) We predict new data: $Y \approx \hat{f}(X)$



Randomness of the model depending on samples

$$E\left[\hat{f}(x) - E\left[\hat{f}(x)\right]\right]^2$$

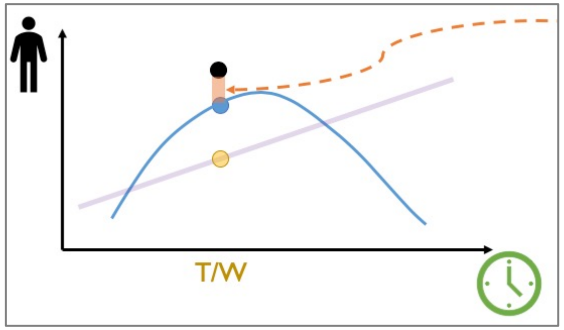
Predicted model

Prediction error explained

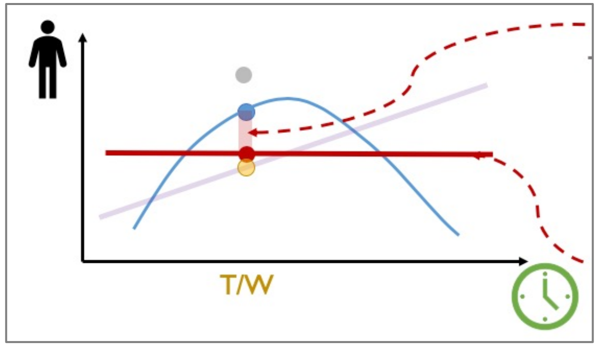
1) How the data is: $Y = f(X) + \varepsilon$
2) We suppose model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$
2) We predict new data: $Y \approx \hat{f}(X)$

$$\mathrm{E}\left[\left(\mathrm{Y} - \hat{f}(x)\right)^2\right] =$$

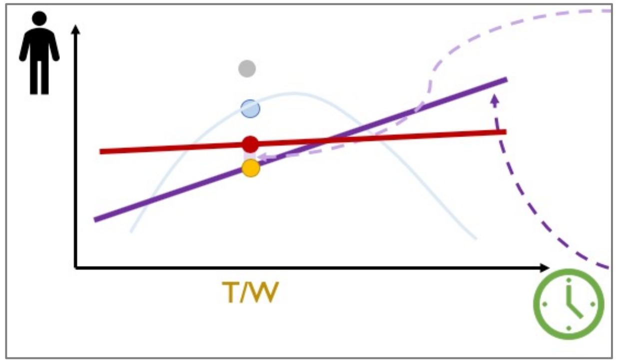
$$E[Y - f(x)]^2 + \left(f(x) - E[\hat{f}(x)]\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]$$



Irreducible error



Bias^2



Variance

Prediction error explained: Linear Regression

1) How the data is: $Y = f(X) + \varepsilon$
2) We suppose model: $\hat{f}(X) = \widehat{\beta_0} + \widehat{\beta_1} X$
2) We predict new data: $Y \approx \hat{f}(X)$

$$E\left[\left(Y - \hat{f}(x)\right)^2\right] =$$

$$E[Y - f(x)]^2 + \left(f(x) - E[\hat{f}(x)]\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]$$

$$E[Y - f(x)]^2 \rightarrow E[\varepsilon]^2$$

$$E[\hat{f}(x)] \rightarrow (f(x) - x^T \beta_*)^2$$

Best linear approximation of f

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] \rightarrow \frac{p}{N} E[\varepsilon]^2$$

Irreducible error

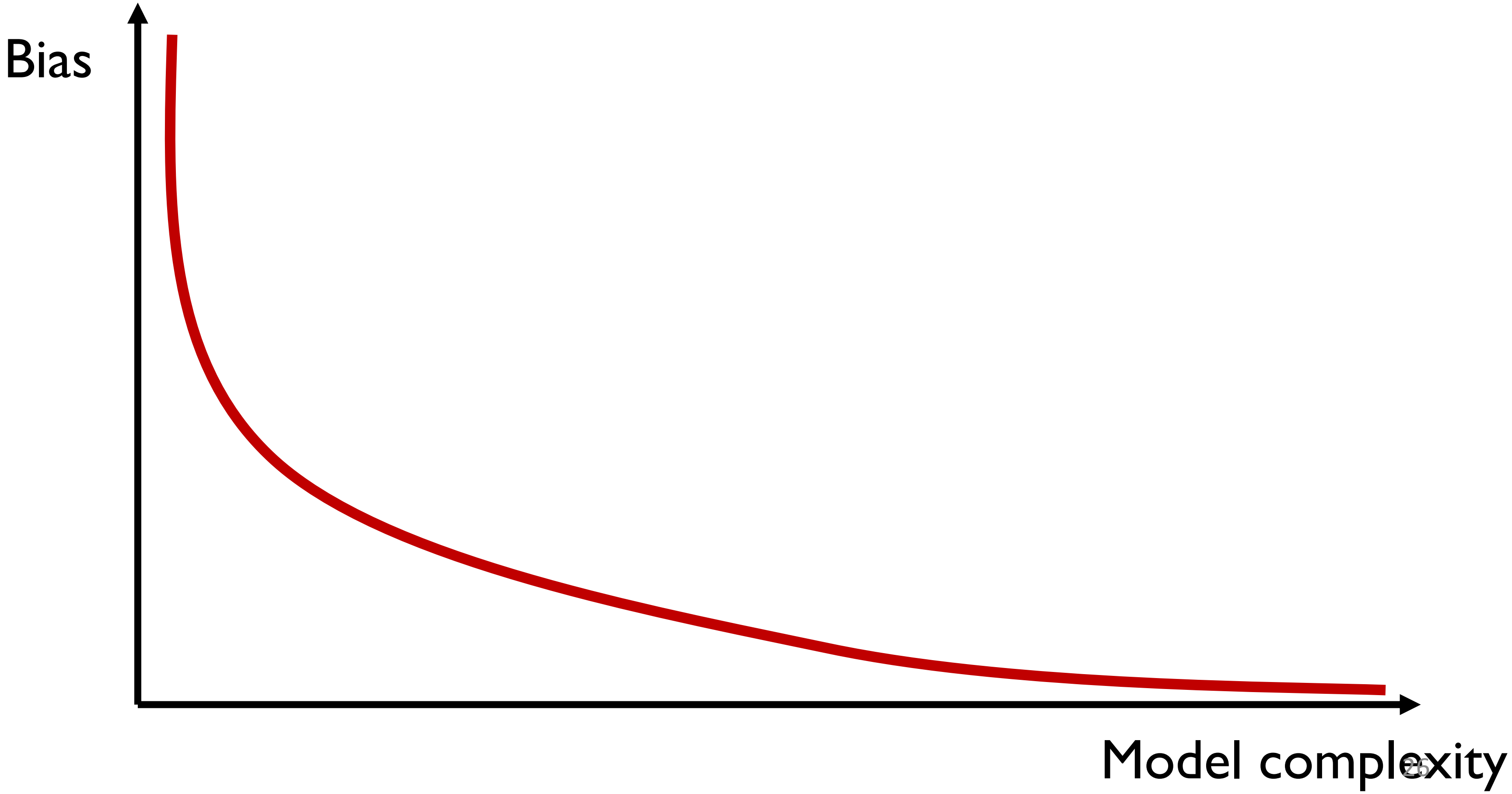
Bias^2

Variance

Bias and model complexity

$$\left(f(x) - E\left[\hat{f}(x)\right]\right)^2 \longrightarrow (f(x) - x^T\beta_*)^2$$ For linear regression

The more parameters, the better we can fit f

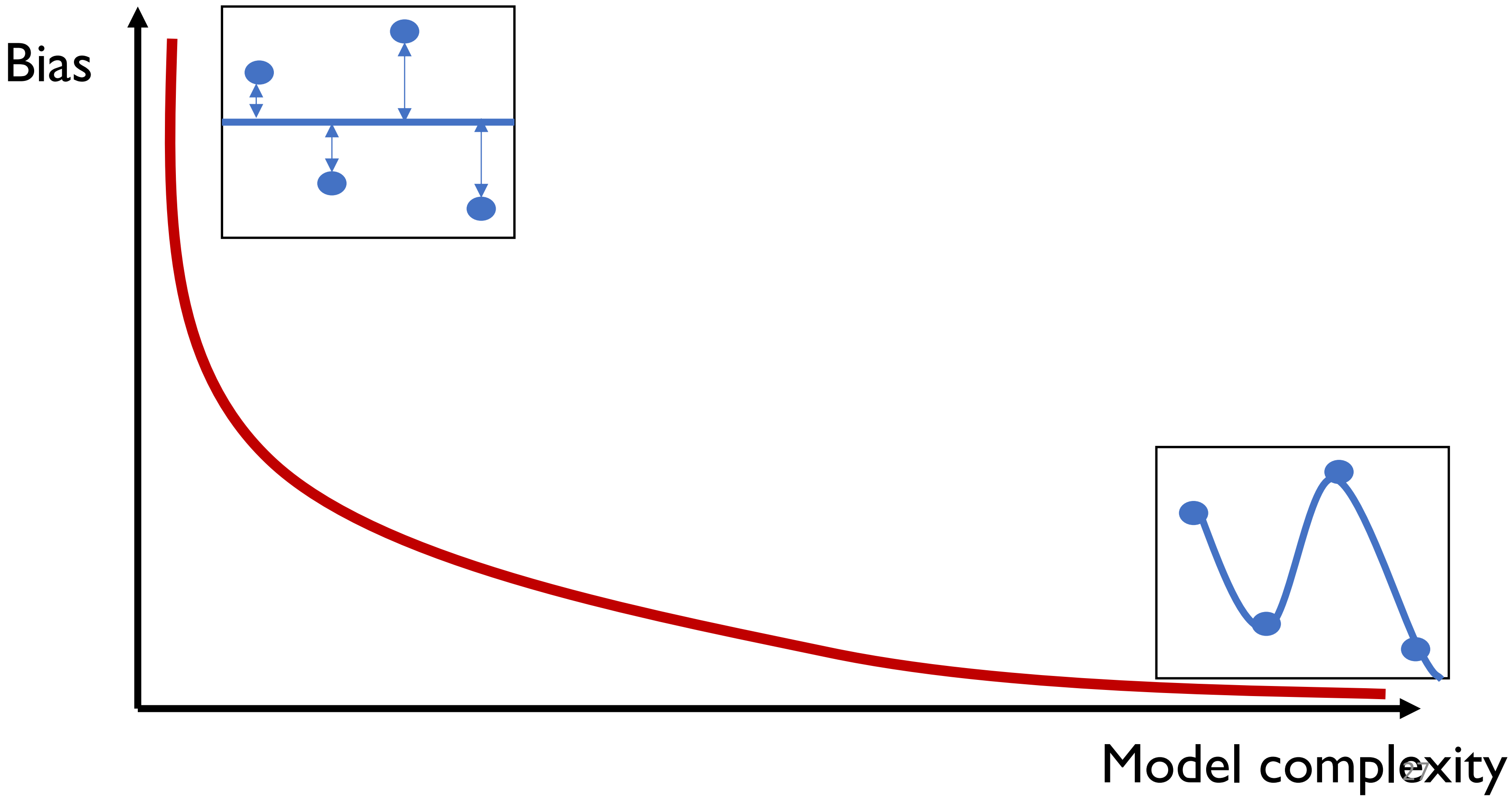


Bias and model complexity

$$(f(x) - E[\hat{f}(x)])^2 \longrightarrow (f(x) - x^T \beta_*)^2$$

For linear regression

The more parameters, the better we can fit f

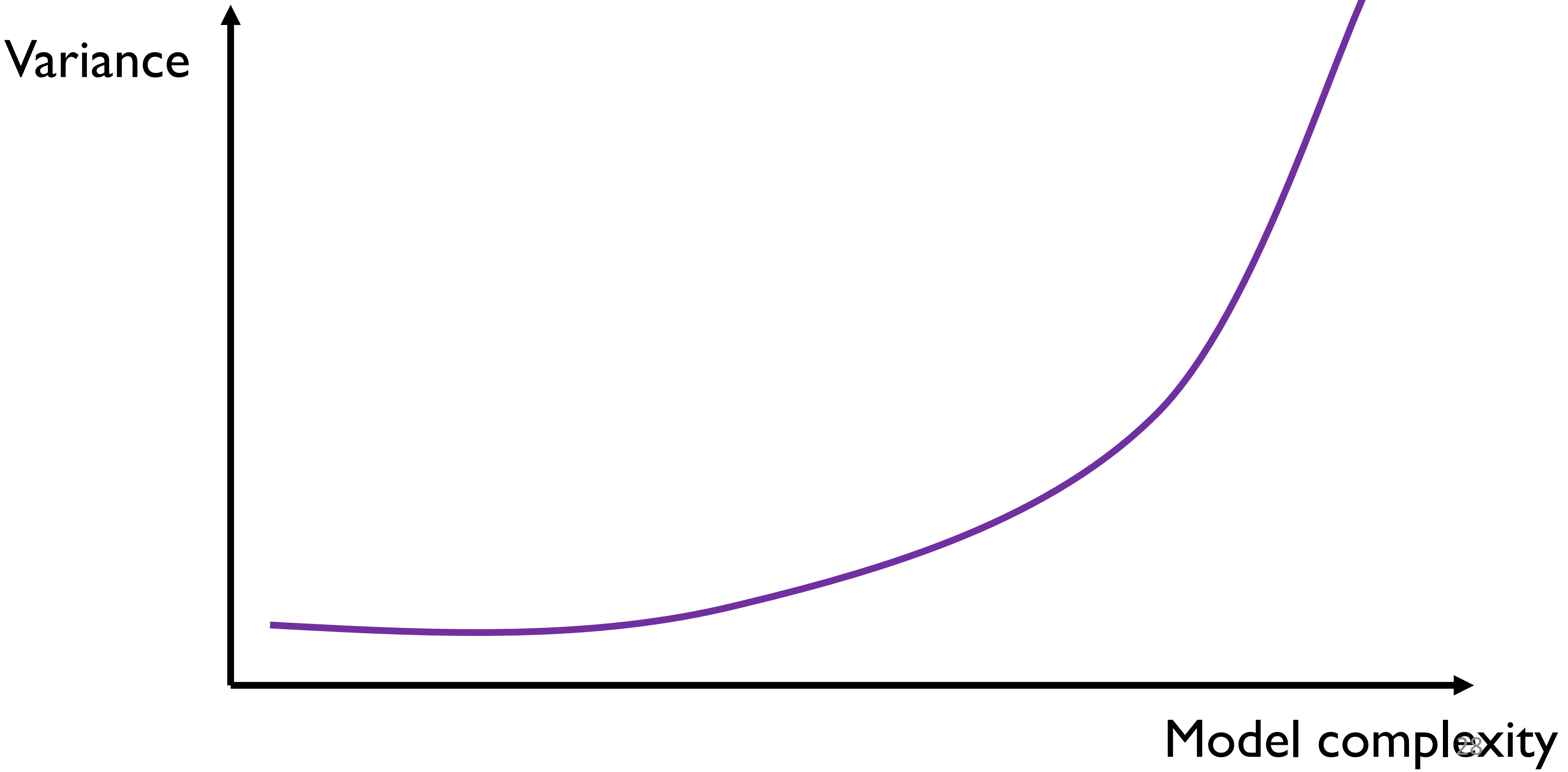


Variance and model complexity

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] \longrightarrow \frac{p}{N} E[\varepsilon]^2$$

For linear regression

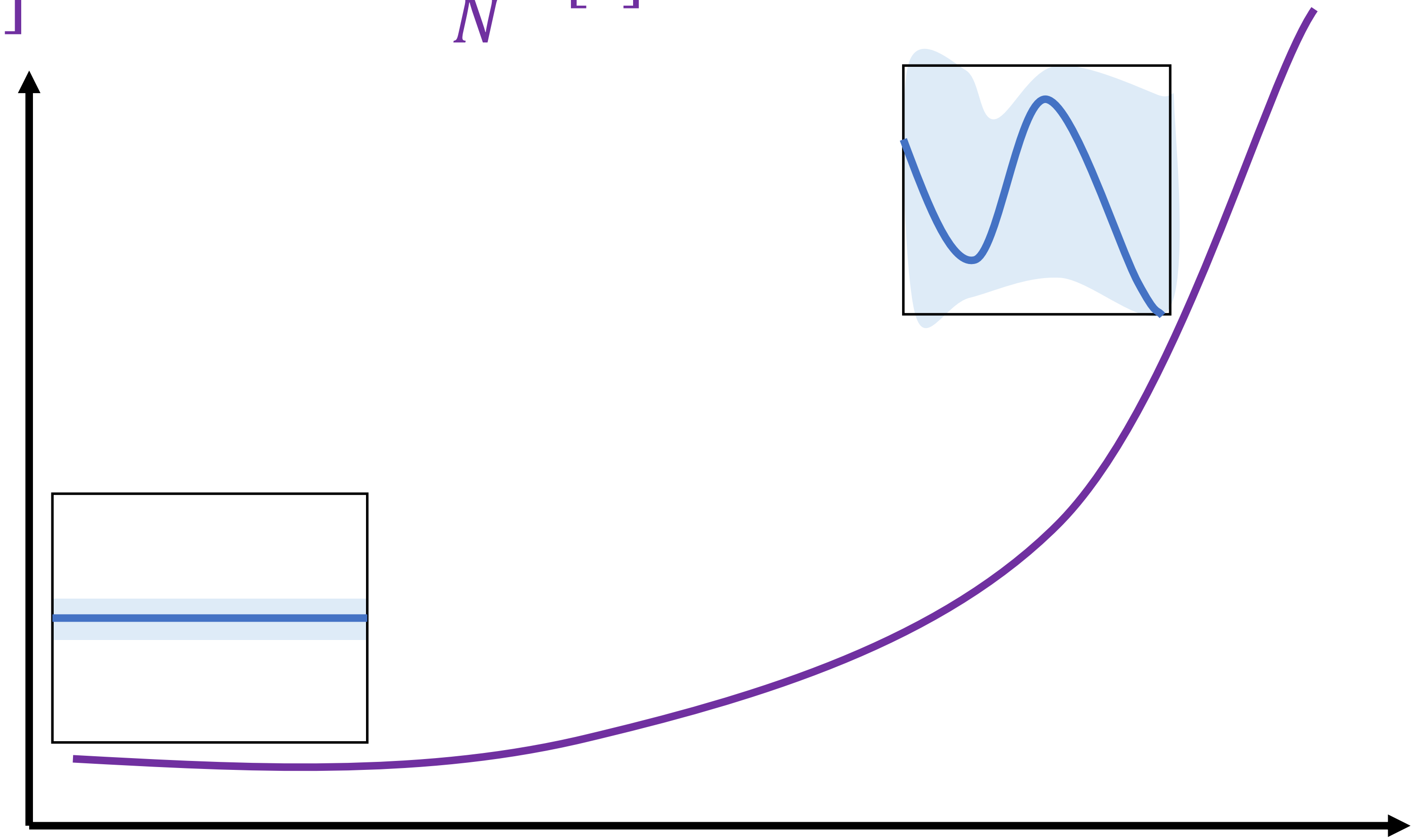
The more parameters, the more variability in prediction



Variance and model complexity

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] \longrightarrow \frac{p}{N} E[\varepsilon]^2$$

For linear regression



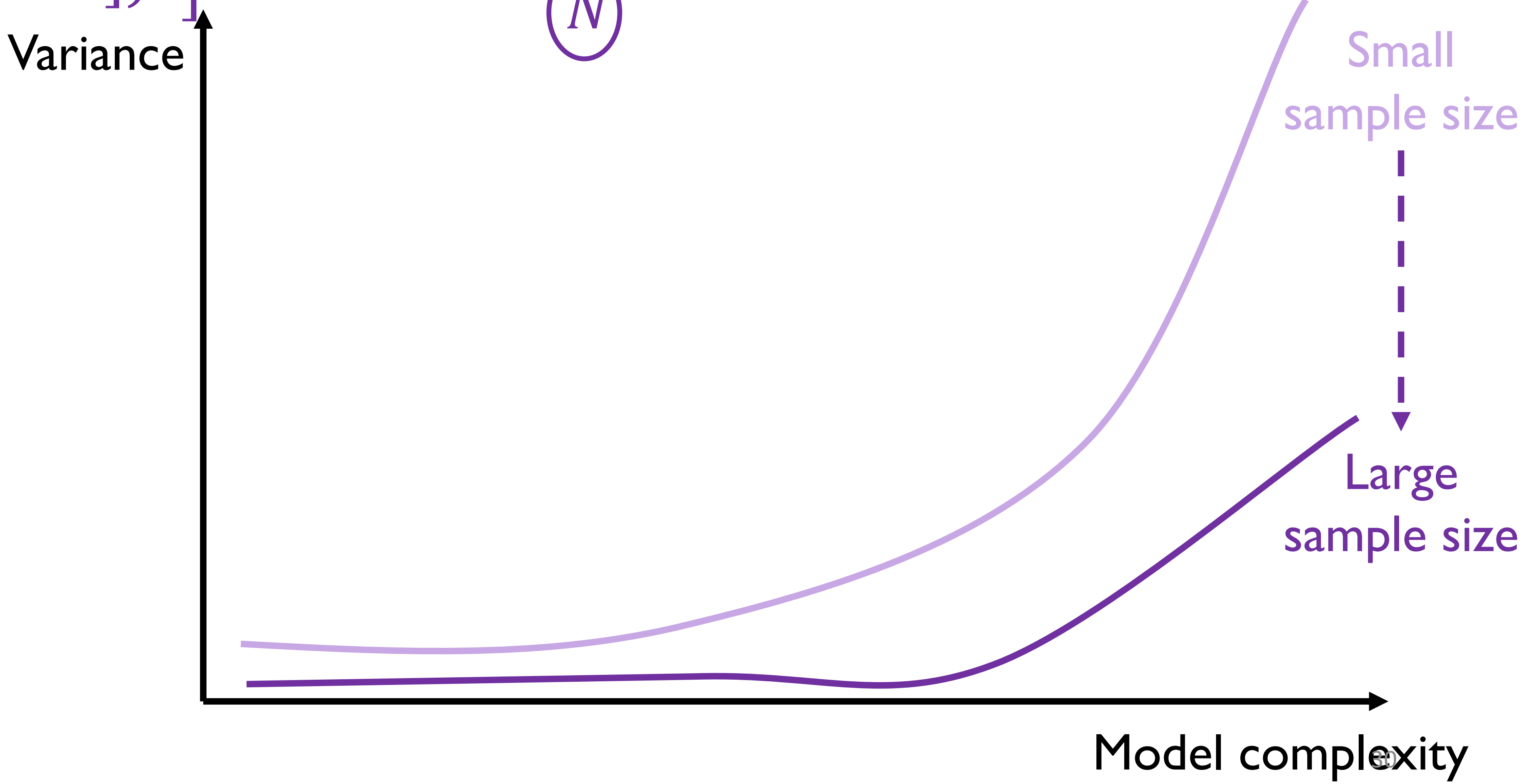
The more parameters, the more variability in prediction

Variance and sample size

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] \longrightarrow \frac{p}{N} E[\varepsilon]^2$$

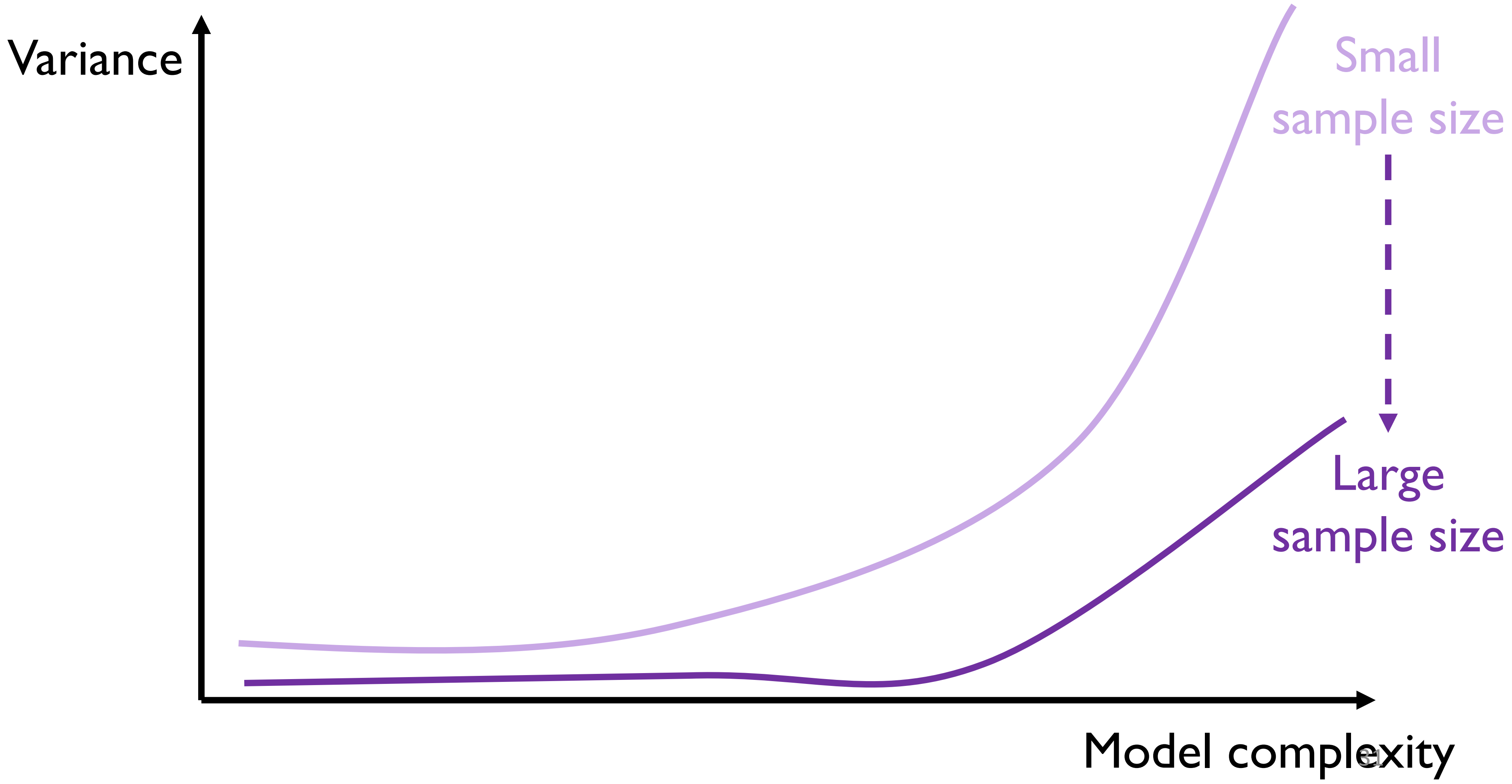
For linear regression

The more samples used, the smaller the variance

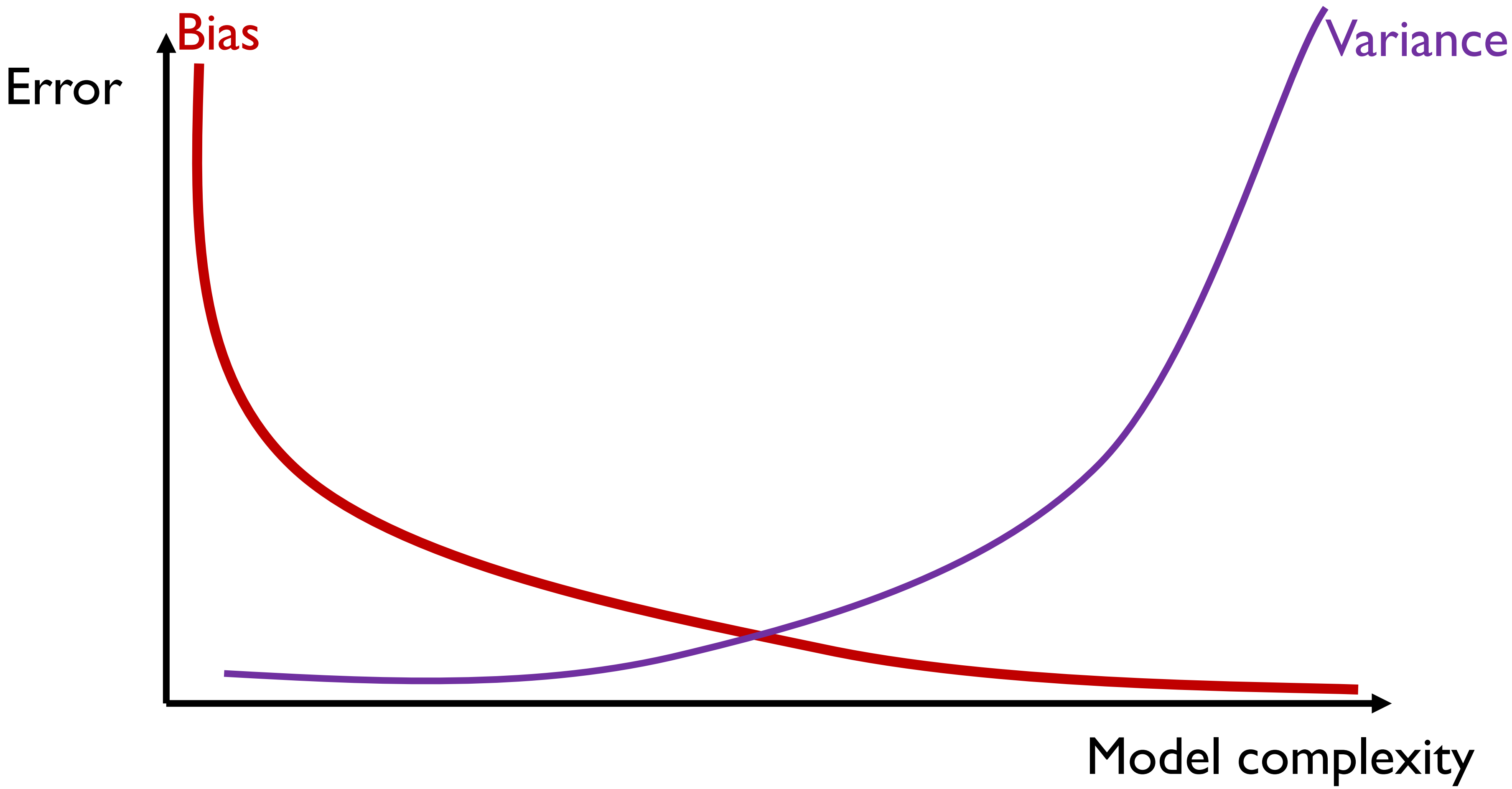


Variance and sample size

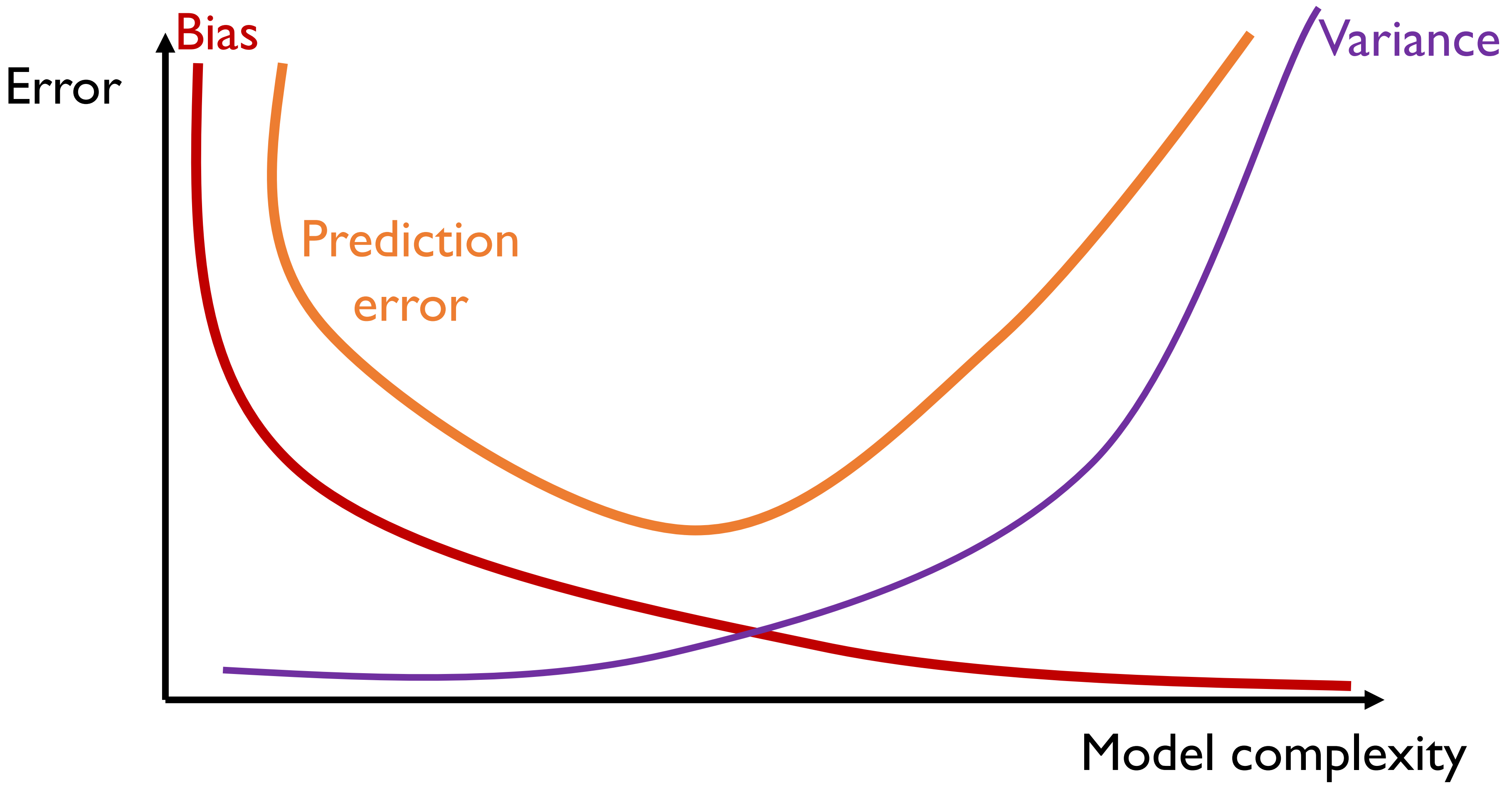
The more samples used, the smaller the variance



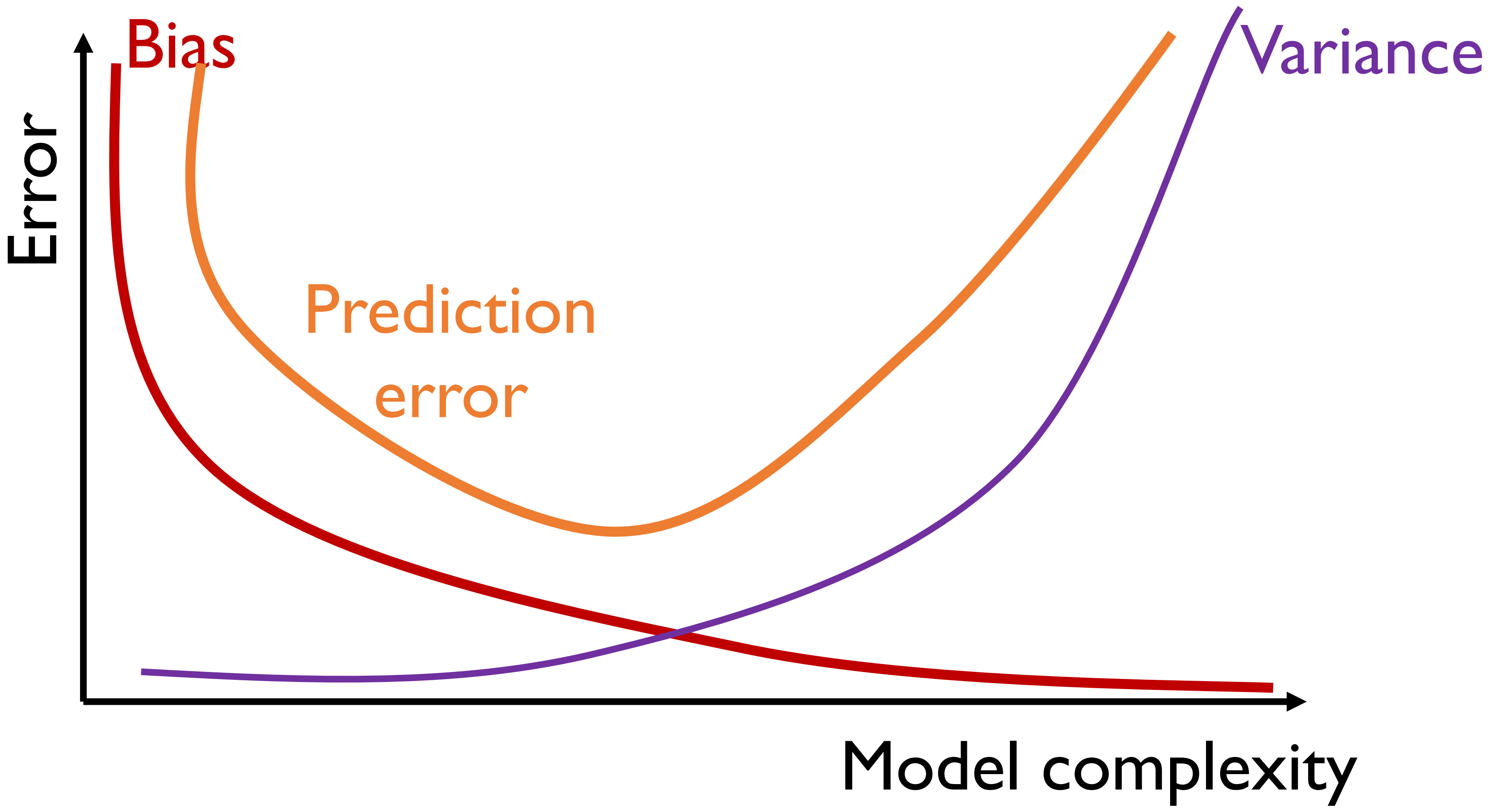
Expected prediction error: Bias and Variance Tradeoff



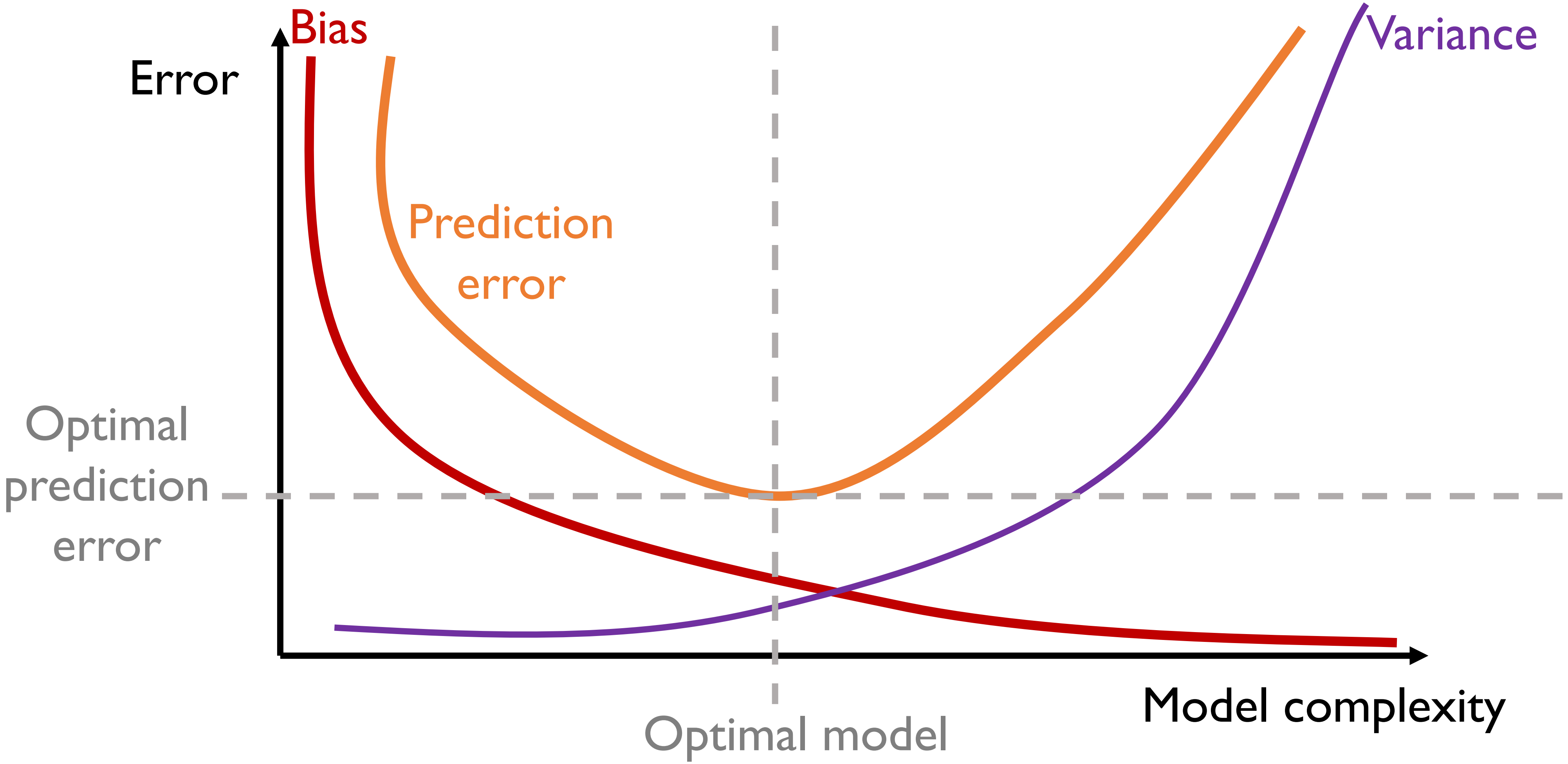
Expected prediction error: Bias and Variance Tradeoff



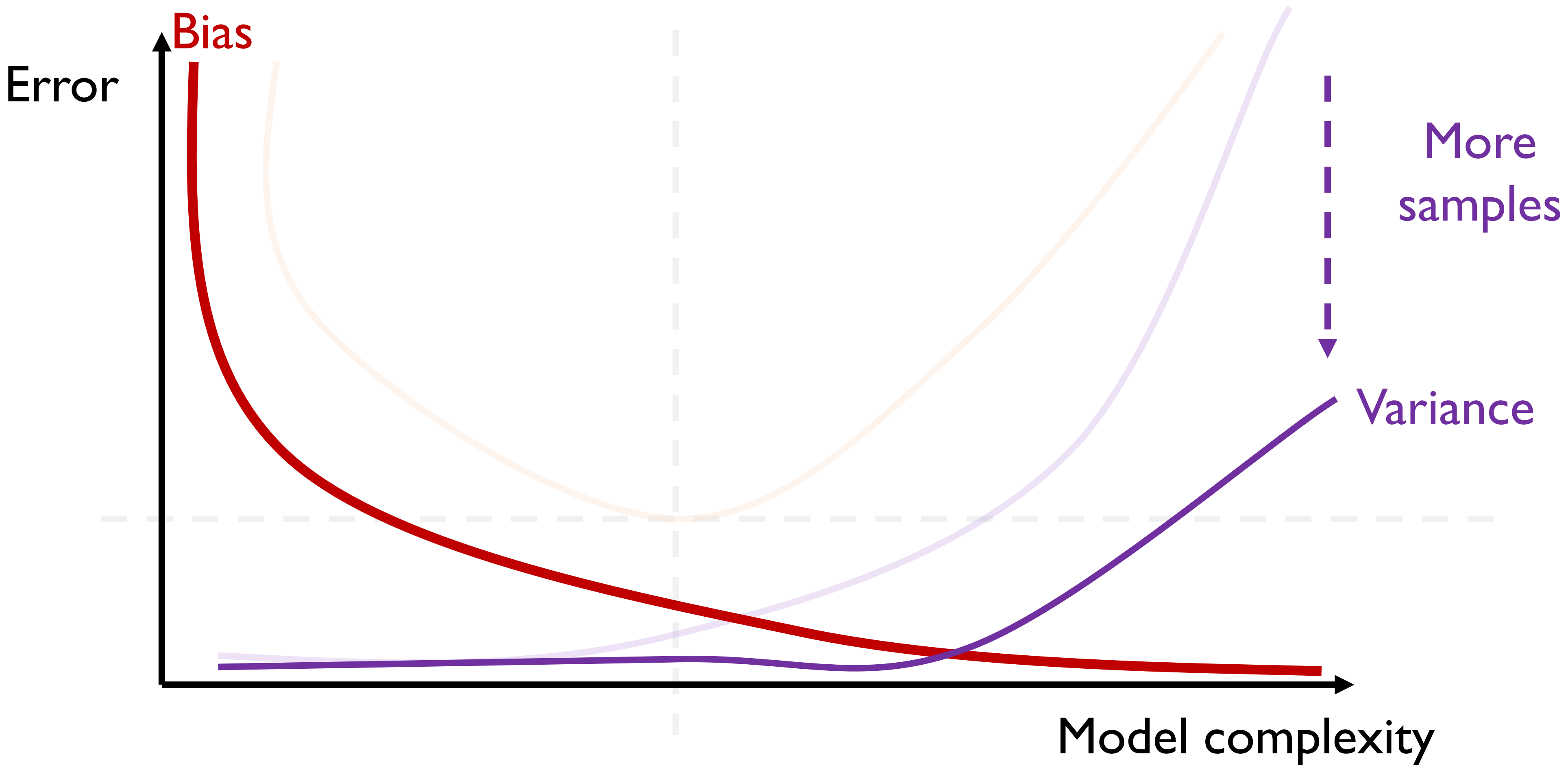
Expected prediction error: Bias and Variance Tradeoff



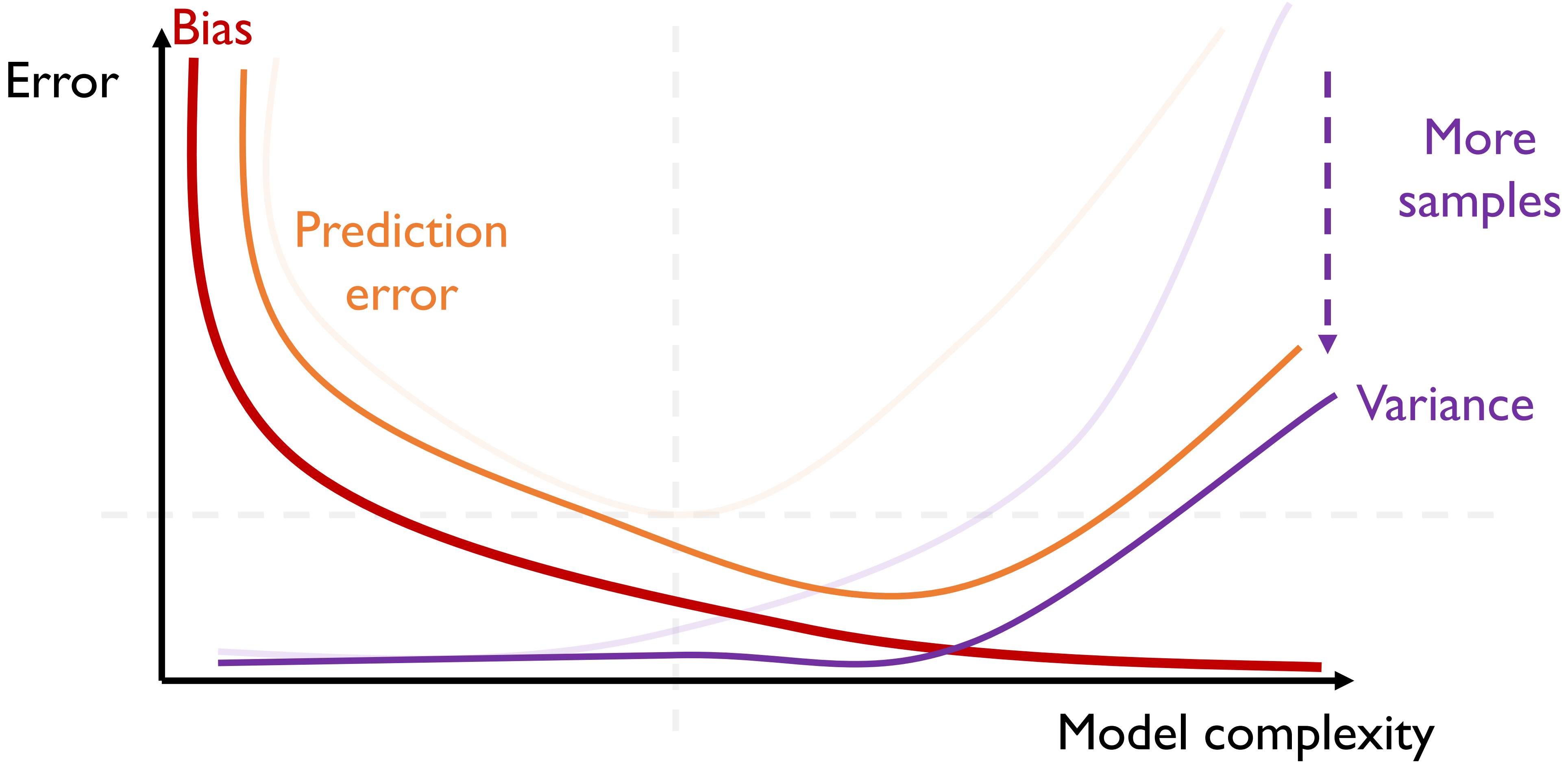
Expected prediction error: Bias and Variance Tradeoff



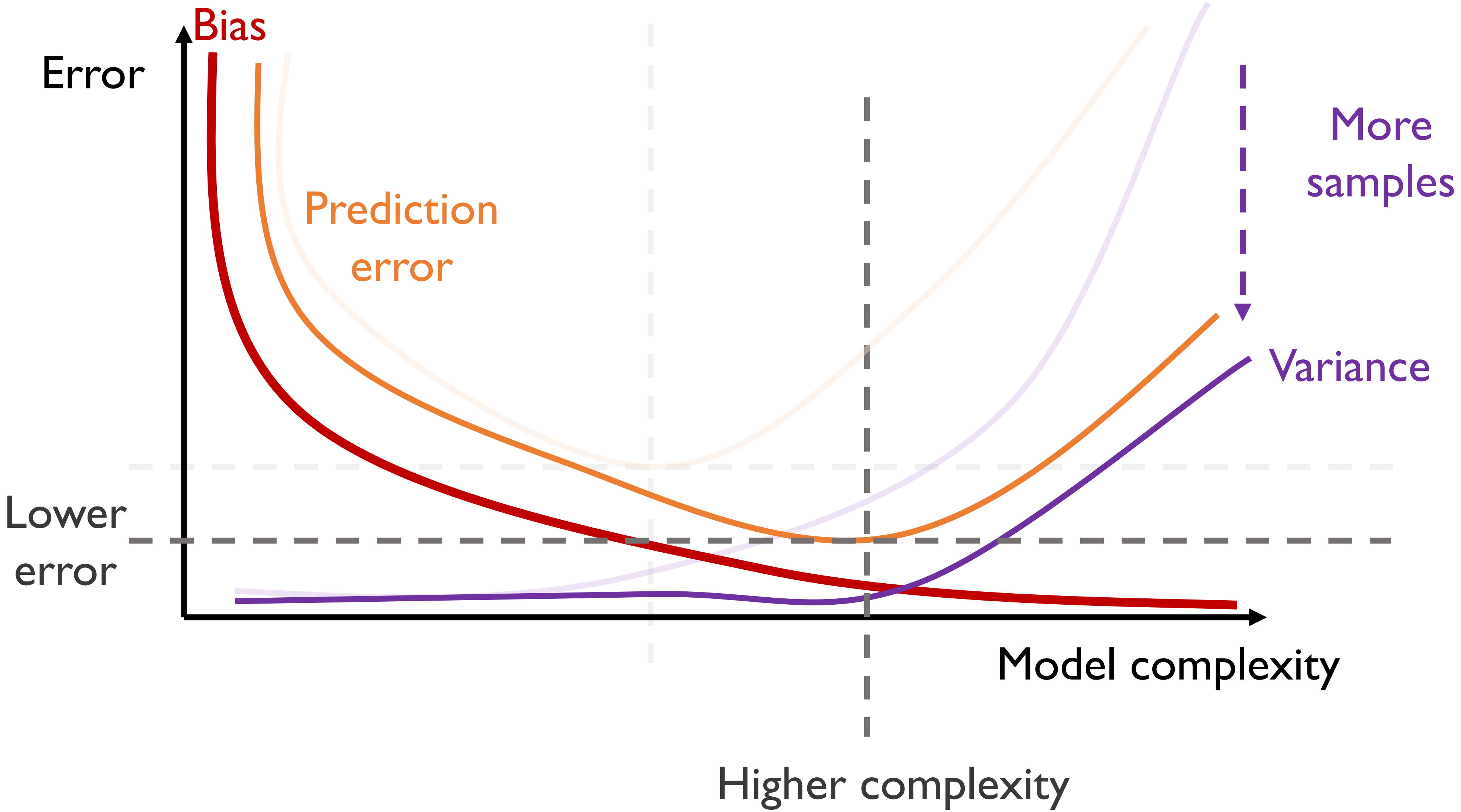
Expected prediction error: More samples



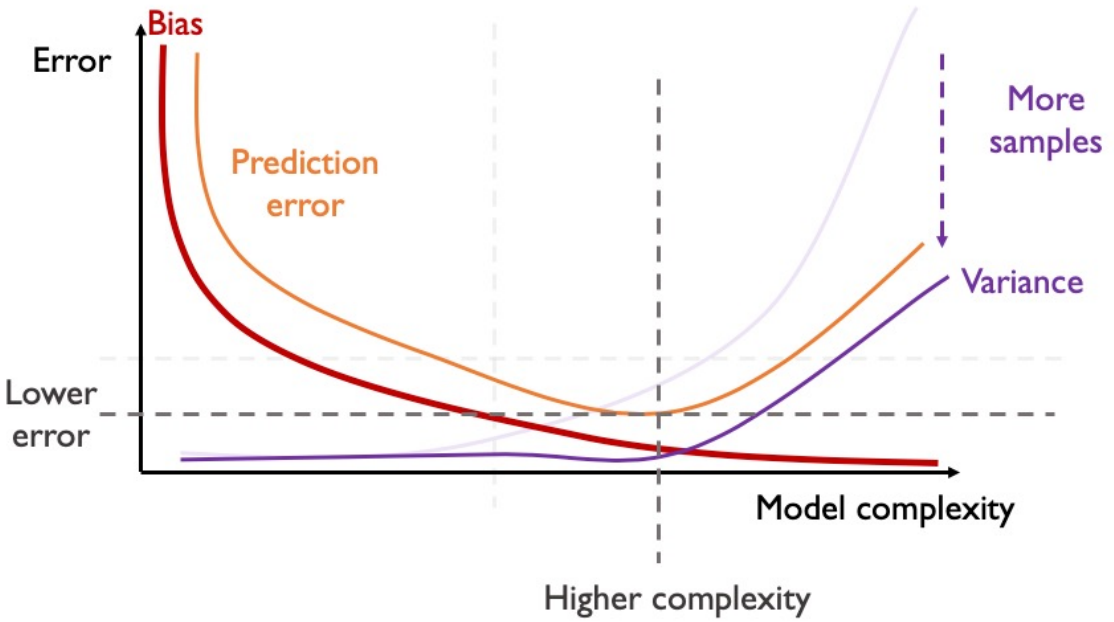
Expected prediction error: More samples



Expected prediction error: More samples



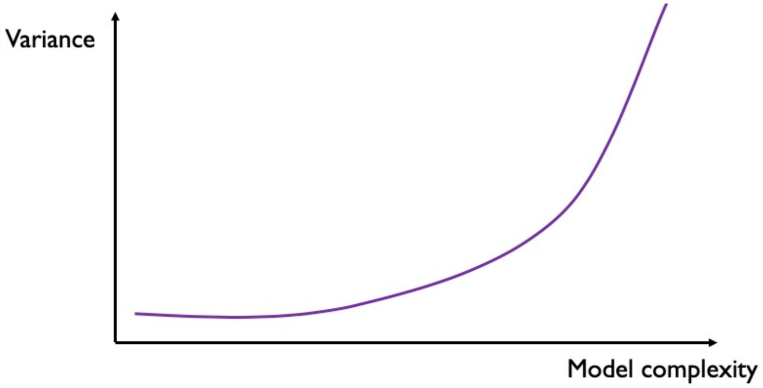
Bias vs Variance: What to keep in mind



1) Error depends on:
   1) Bias: How well we fit the underlined model (decreases with complexity)
   2) Variance: Variability from the sampled data (increases with complexity)
2) More complex models reduce prediction error **only** if there are enough samples

Why is reducing # parameters useful?

1) Reduce Variance



2) Interpretability

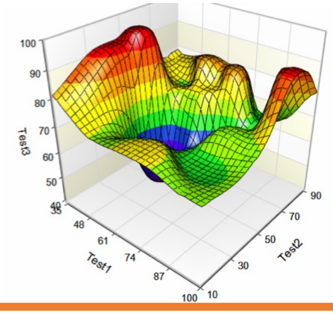
$$y = 3x_1 + 2$$

vs.

$$y = 3x_1^2 + x_2 x_3 + 10\log(x_4)$$

3) Easy optimization

vs.



parameters = Model complexity

How do we reduce # parameters for regression?

1) Subset selection

For each k

Select the *best set* of k features (smallest residual)

2) Principal Components

For each k

Create regression using the *kth principal components*

3) Regularization

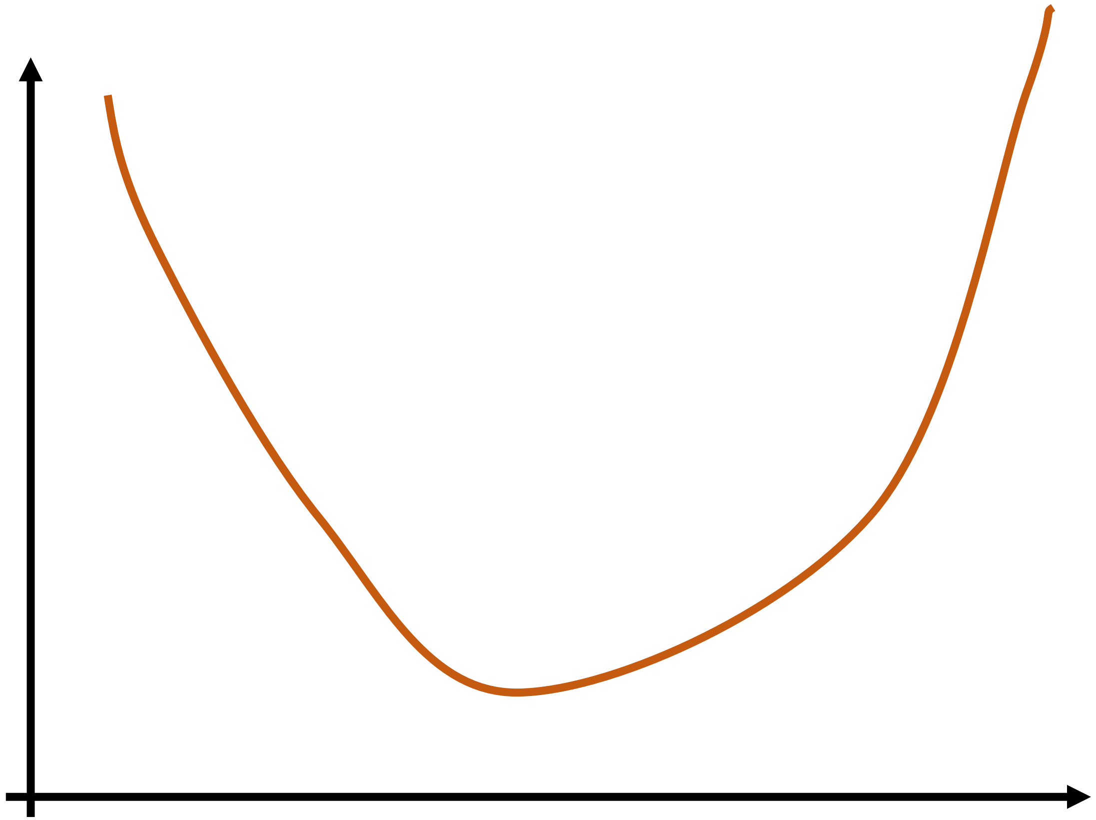
Add constraints

$$\min_{\beta} \|X\beta - Y\|_2^2 + \boldsymbol{\lambda}\, \boldsymbol{L(\beta)}$$

\# parameters = Model complexity

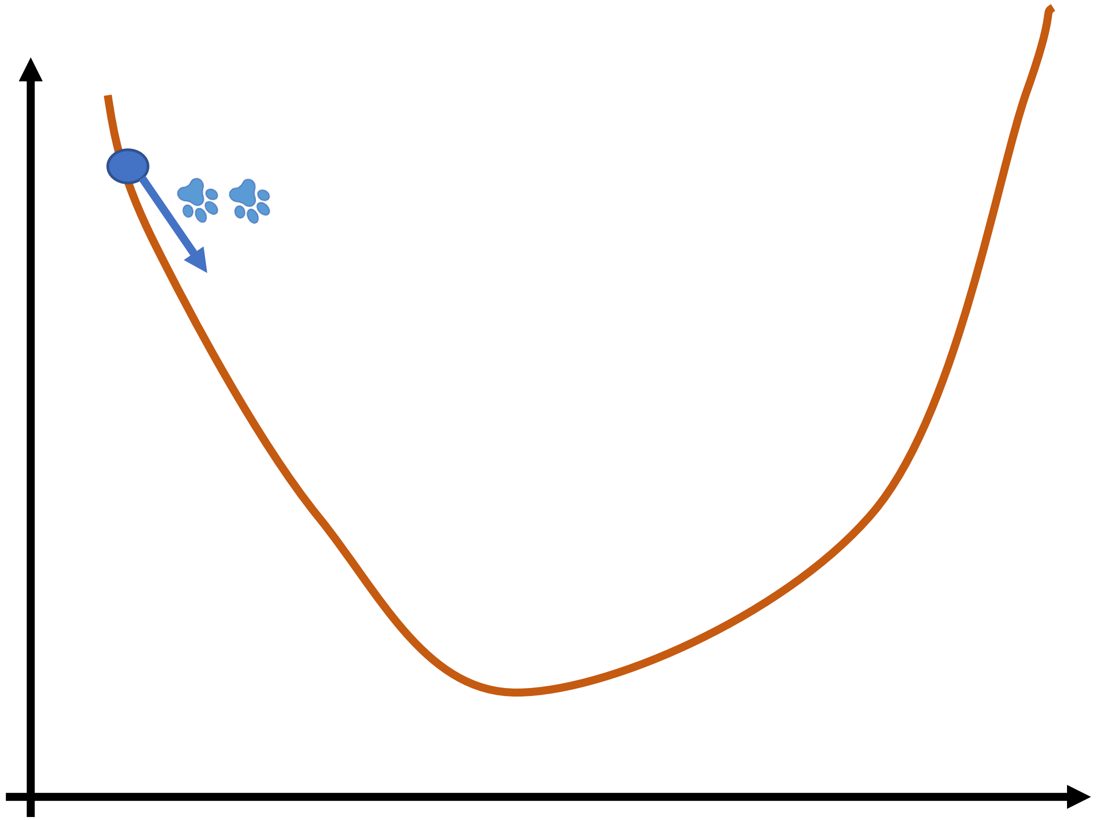
Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$



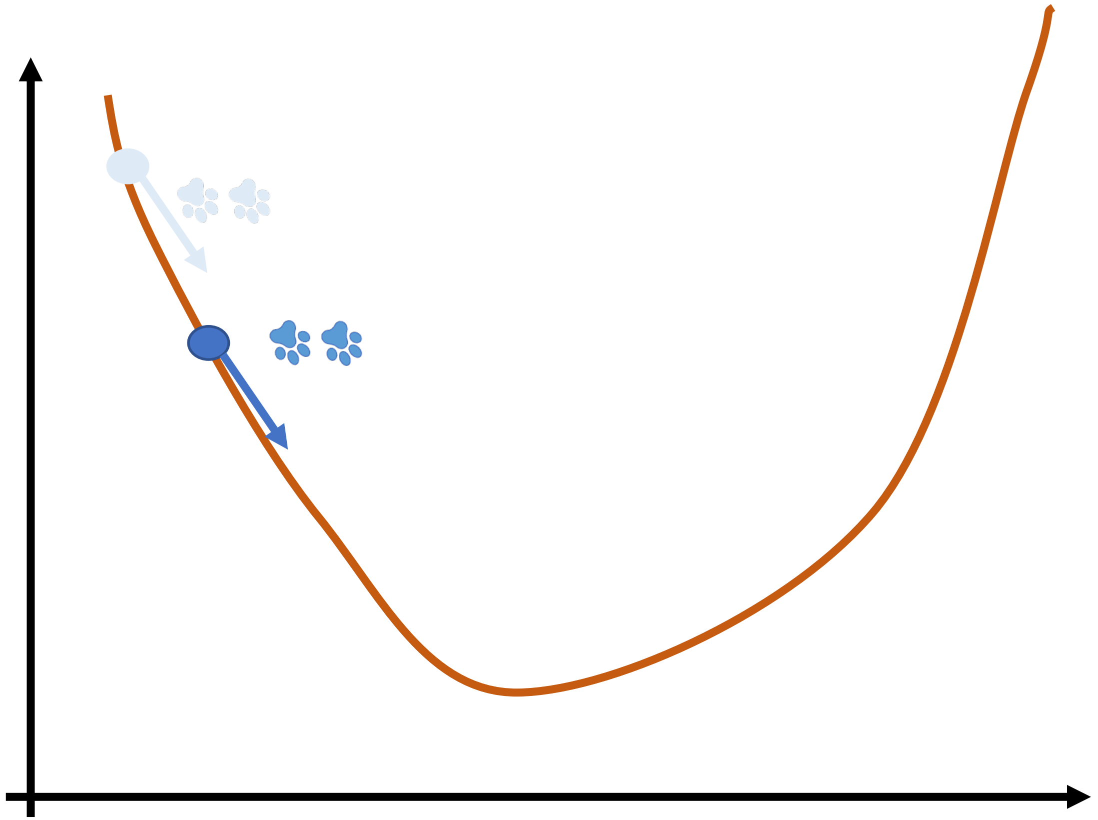
Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$



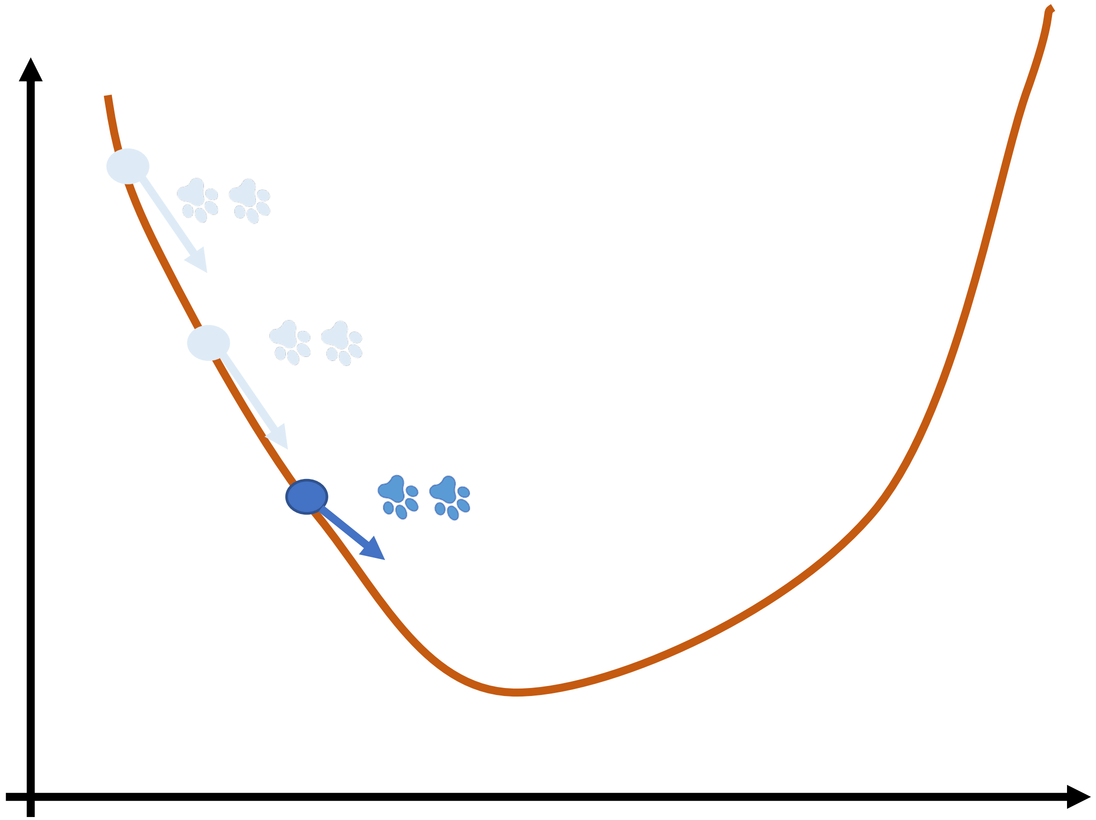
Why do we need regularization? Optimization is hard

$$\min_{\beta} L(f(X), Y)$$



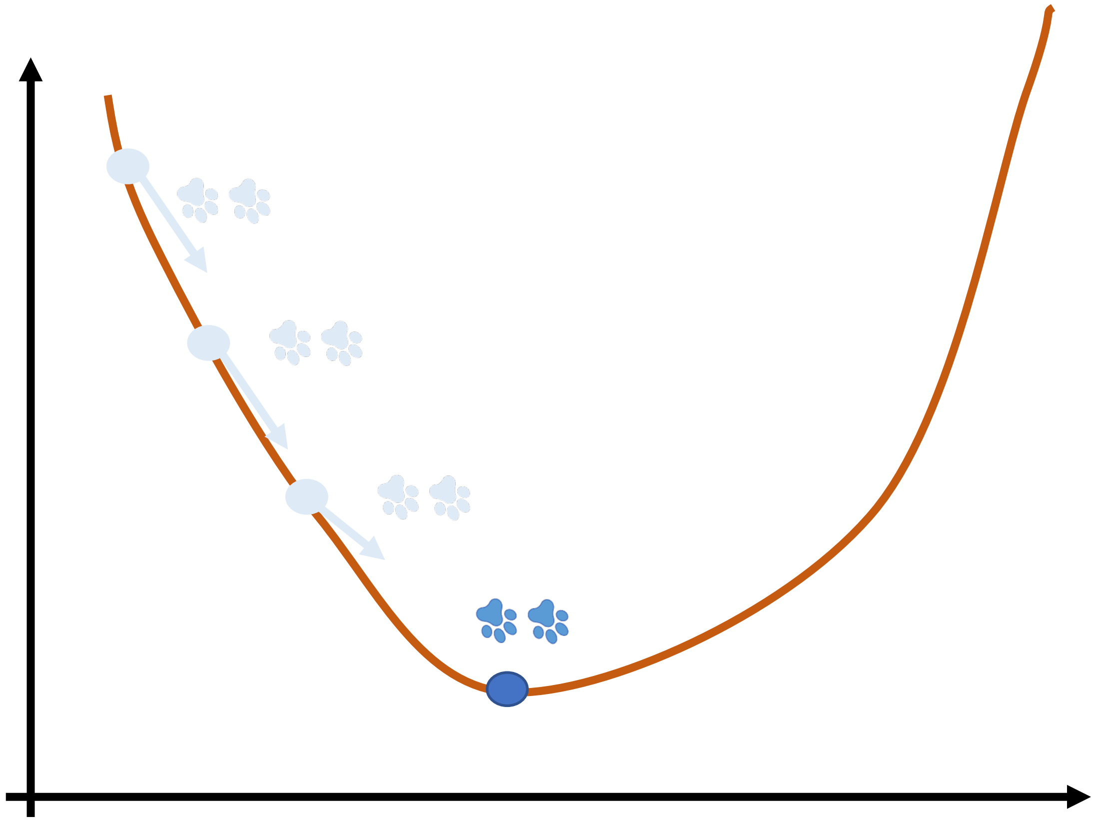
Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$



Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$

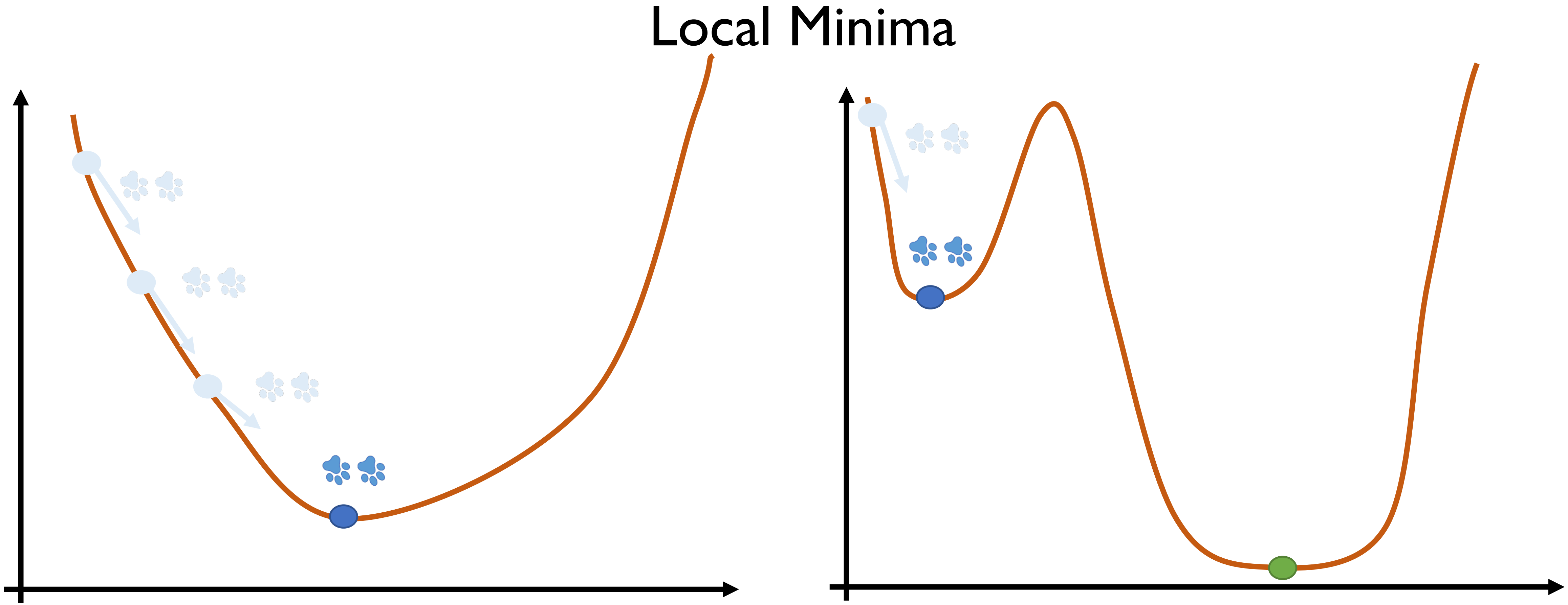


Optimization algorithm = Select descent direction

Gradient descent
Stochastic gradient descent
Coordinate descent
Newton's method

Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$



(Quasi) convex problem

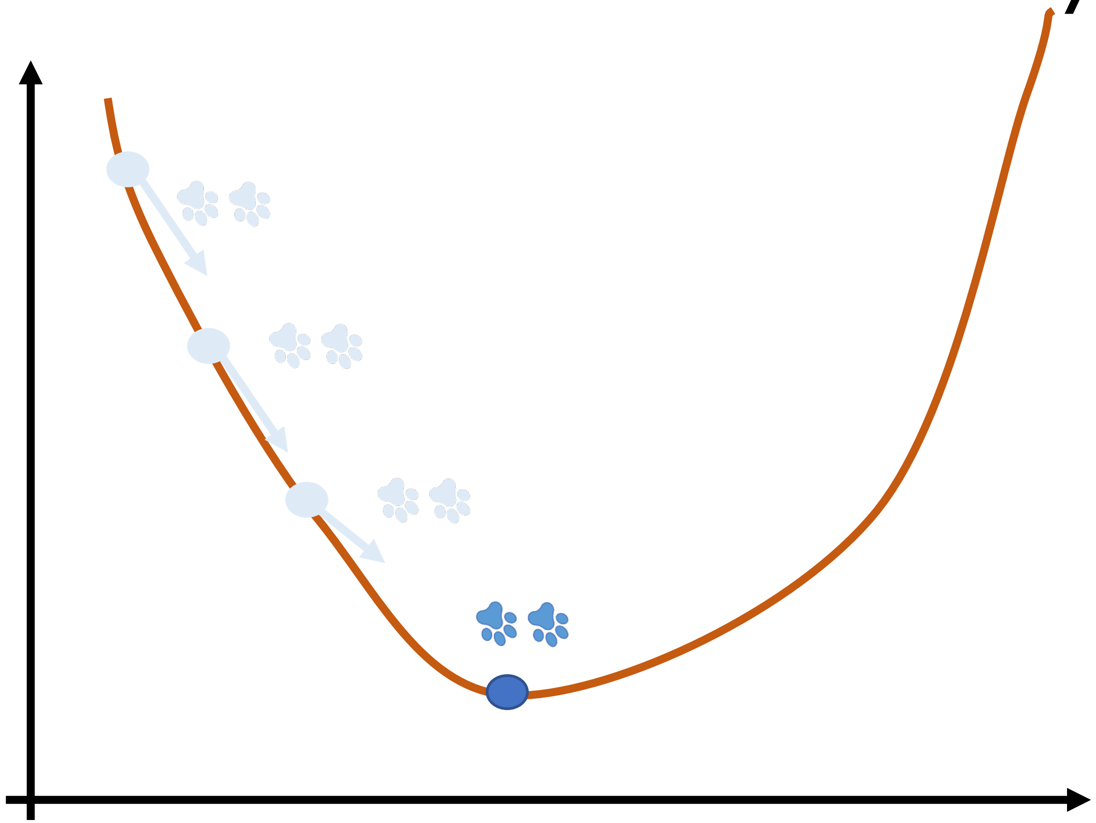
Non-convex problem

Difficult to solve

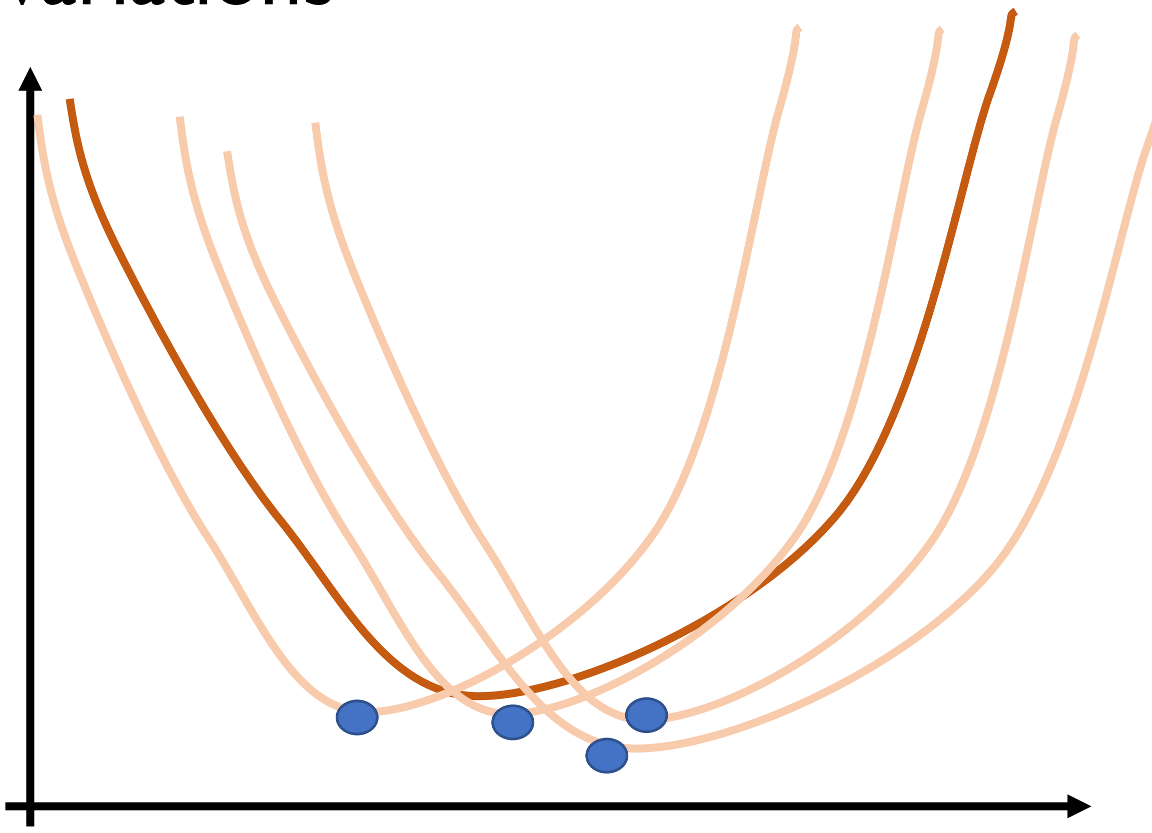
Why do we need regularization?
Optimization is hard

$$\min_{\beta} L(f(X), Y)$$

Sensitivity to variations

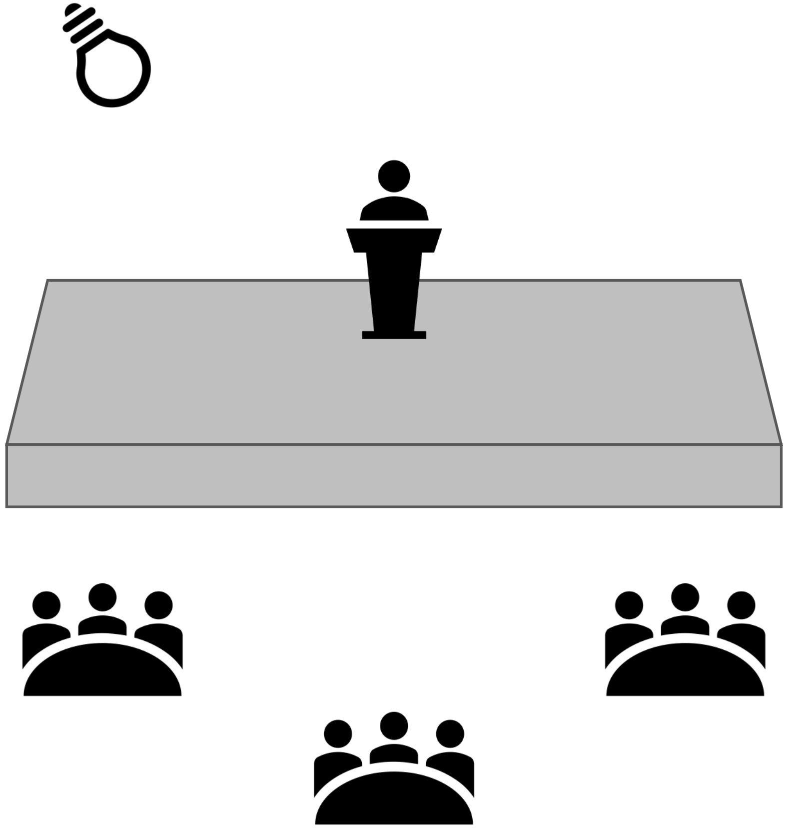


Deterministic problem

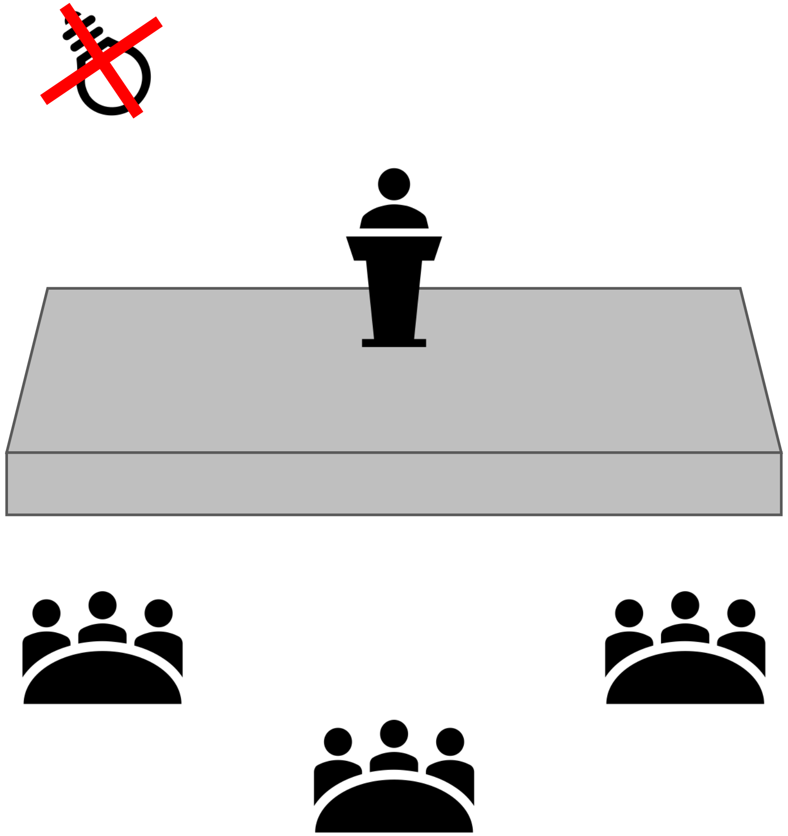


Random problem

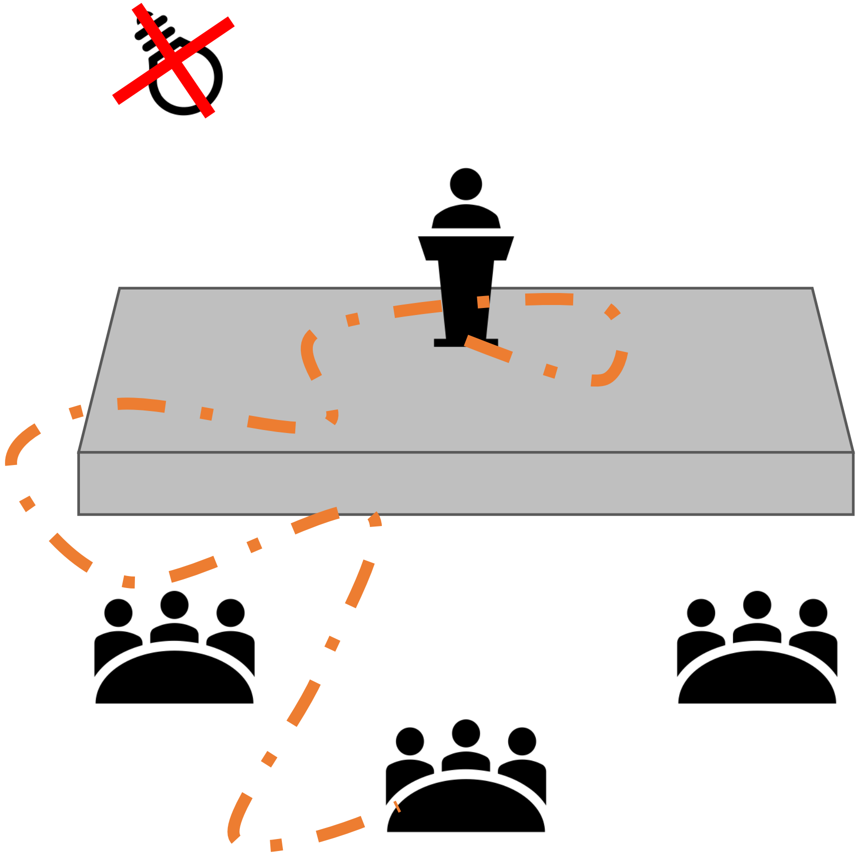
How regularization can simplify optimization?



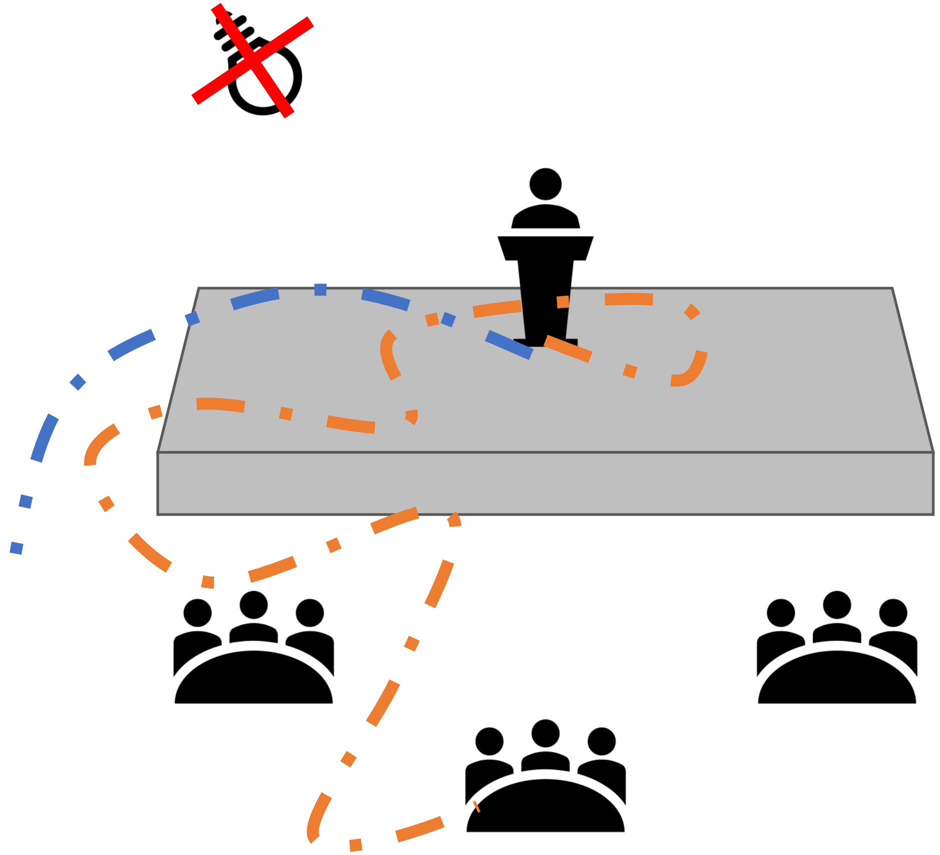
How regularization can simplify optimization?



How regularization can simplify optimization?

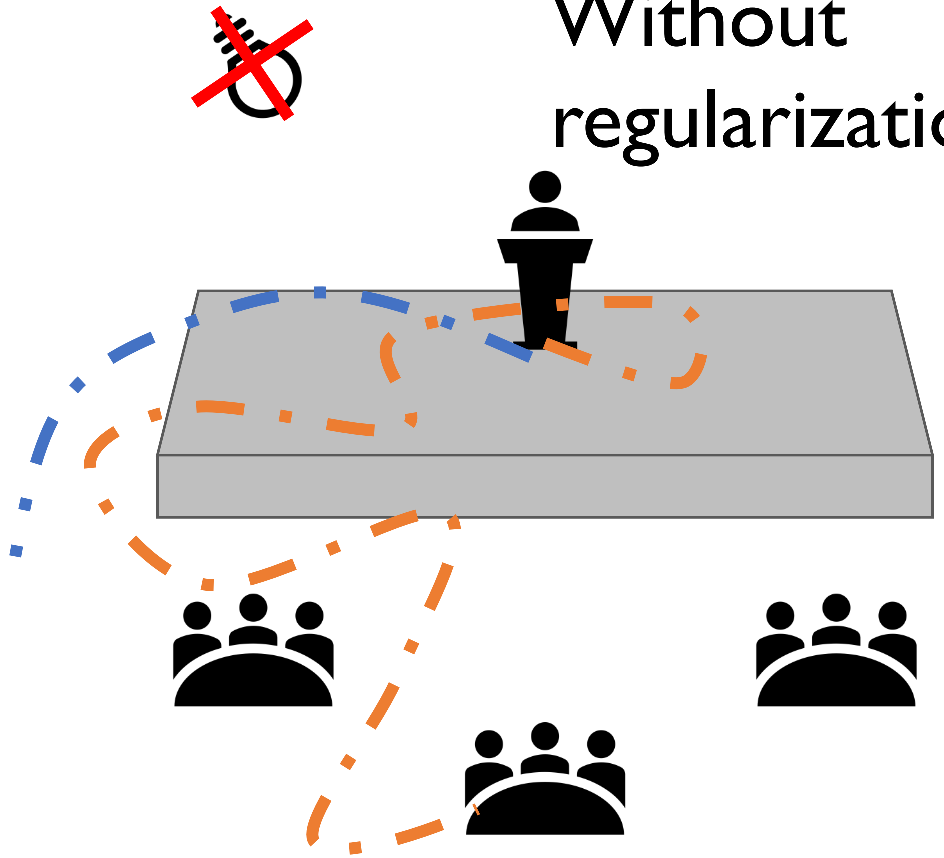


How regularization can simplify optimization?

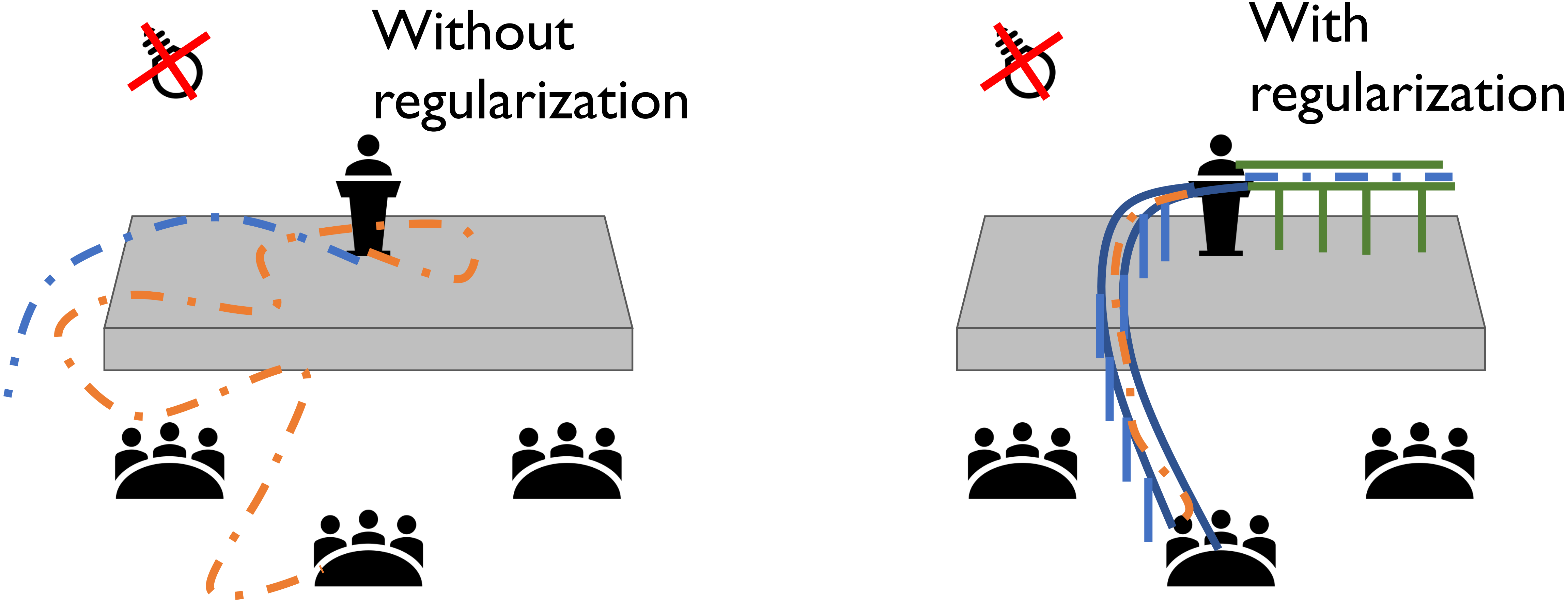


How regularization can simplify optimization?

Without regularization



How regularization can simplify optimization?



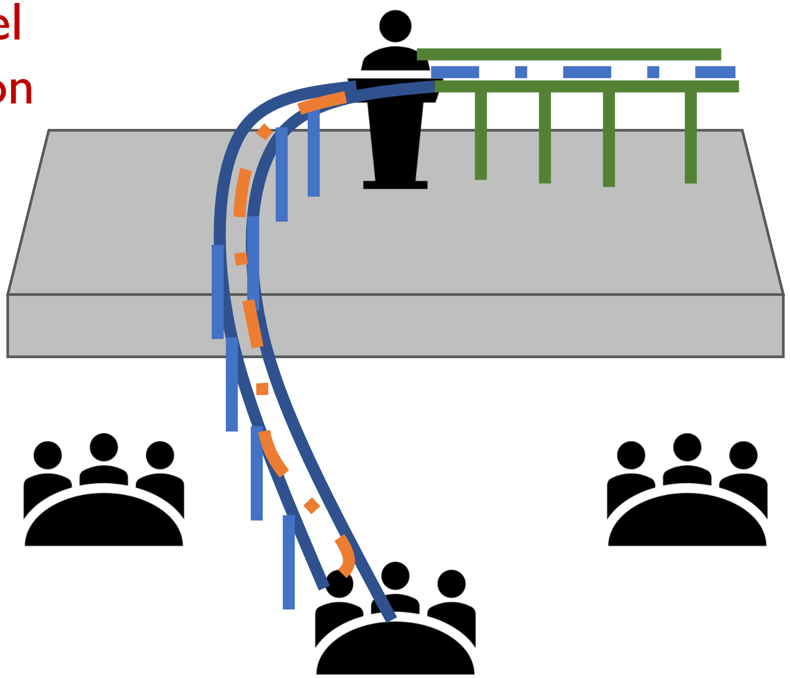
How regularization can simplify optimization?

Bias $\overbrace{(f(x) - x^T\beta_*)^2}^{\text{Model Bias}} + \overbrace{\left(x^T\beta_* - E\left[x^T\widehat{\beta_\alpha}\right]\right)^2}^{\text{Estimation Bias}}$

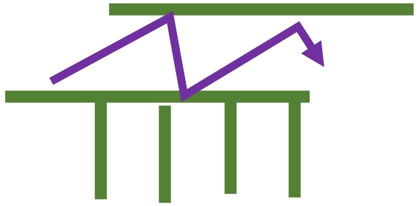
Best linear model

Best linear model with regularization

With regularization



Variance $E\left[\left(x^T\widehat{\beta_\alpha} - E\left[x^T\widehat{\beta_\alpha}\right]\right)^2\right]$



The most used regularizations

Ridge regression
(l_2 regularization)

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} \beta_p^2$$

"Everyone is important"

Lasso regression
(l_1 regularization)

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_p|$$

"Only some are important"

Intuition of regularizations

$$\boxed{\min_{X_1, X_2} 4X_1 + 3\,X_2}$$

Ridge regression
(l$_2$ regularization)

$$X_1^2 + X_2^2 \leq 1$$

"Everyone is important"

Lasso regression
(l$_1$ regularization)

$$|X_1| + |X_2| \leq 1$$

"Only some are important"

Intuition of regularizations

$$\boxed{\min_{X_1, X_2} 4X_1 + 3\,X_2}$$

Ridge regression
(l$_2$ regularization)

$$X_1^2 + X_2^2 \le 1$$

Lasso regression
(l$_1$ regularization)

$$|X_1| + |X_2| \le 1$$

Solution

$$X_1 = -\frac{4}{5}, X_2 = -\frac{3}{5}$$

"Everyone is important"

$$X_1 = -1, X_2 = 0$$

"Only some are important"

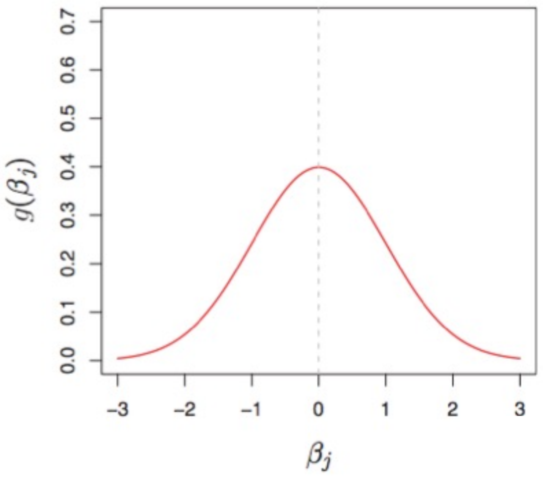
SPARSITY

Intuition of regularizations

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} \beta_p^2$$

Ridge regression

(Normal prior)



"Everyone is important"

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_p|$$

Lasso regression

(Laplace prior)

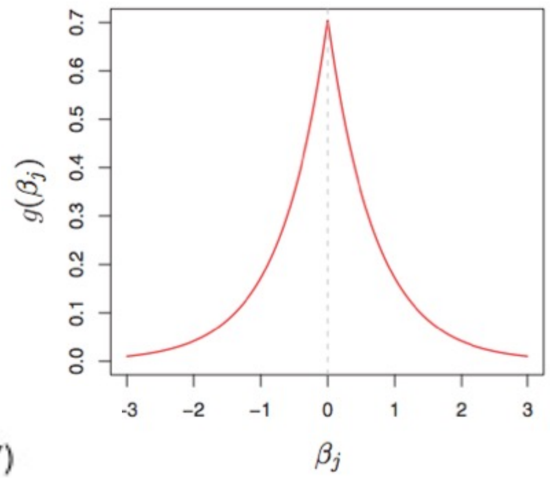


FIGURE 6.11, ISL (8th printing 2017)

"Only some are important"

SPARSITY

When to use regularizations

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} \beta_p^2$$

Ridge regression

All of coefficients have around equal contribution

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_p|$$

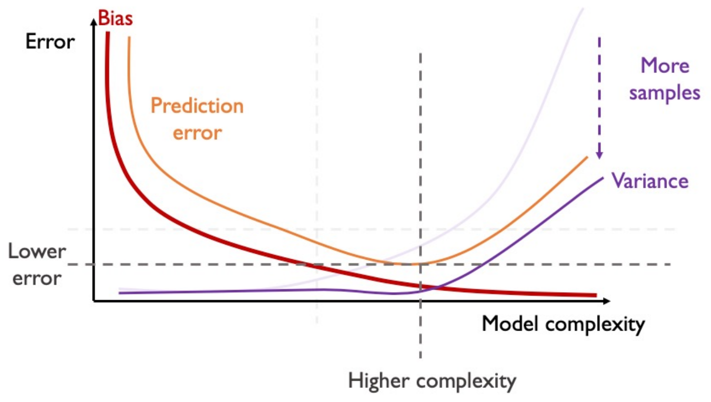
Lasso regression

Only some coefficients are non zero

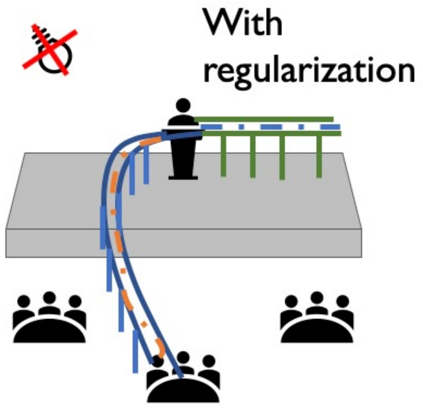
To choose: λ = Model complexity

How to pick model complexity?

1) We understand prediction error

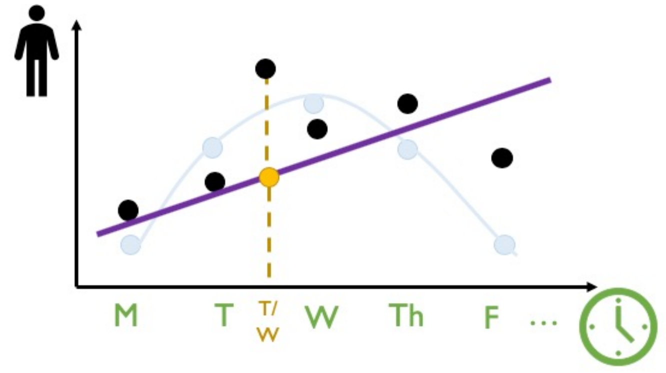


2) We can control model complexity



Hyperparameters

3) How to estimate prediction error?



How to estimate prediction error?

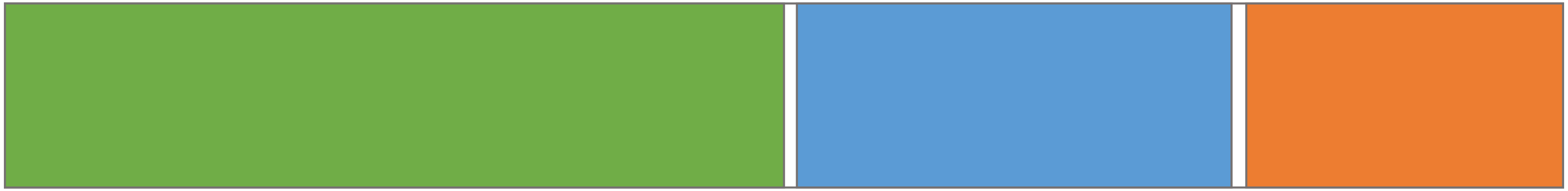
We need to generalize to **unseen** data

Your data

How to estimate prediction error?

We need to generalize to **unseen** data

Your data



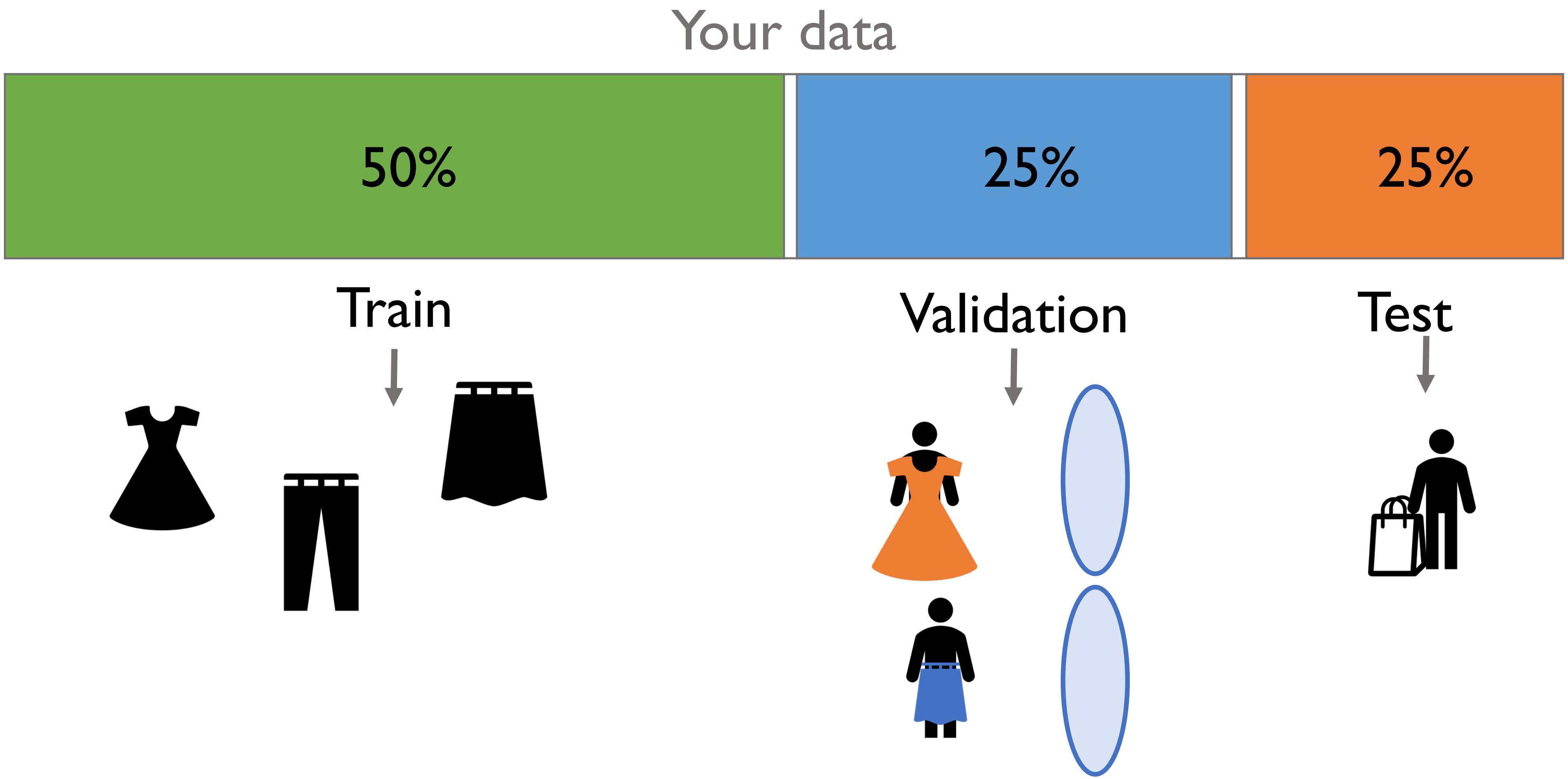
Train → Fit the models

Validation → **Model selection:** choosing hyperparameters (k,λ), models

Test → **Model assessment:** Prediction error of **final** model

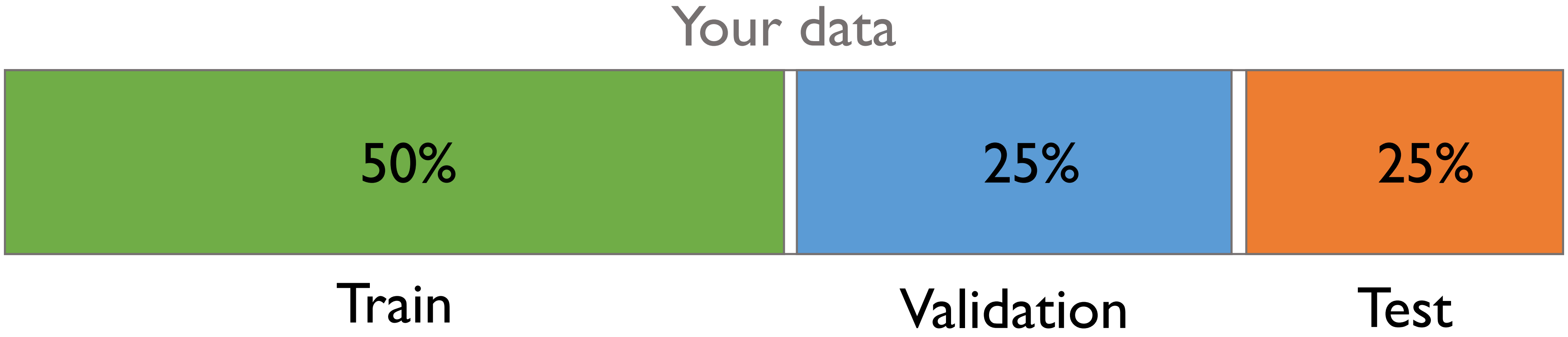
How to estimate prediction error?

We need to generalize to **unseen** data



How to estimate prediction error?

We need to generalize to **unseen** data

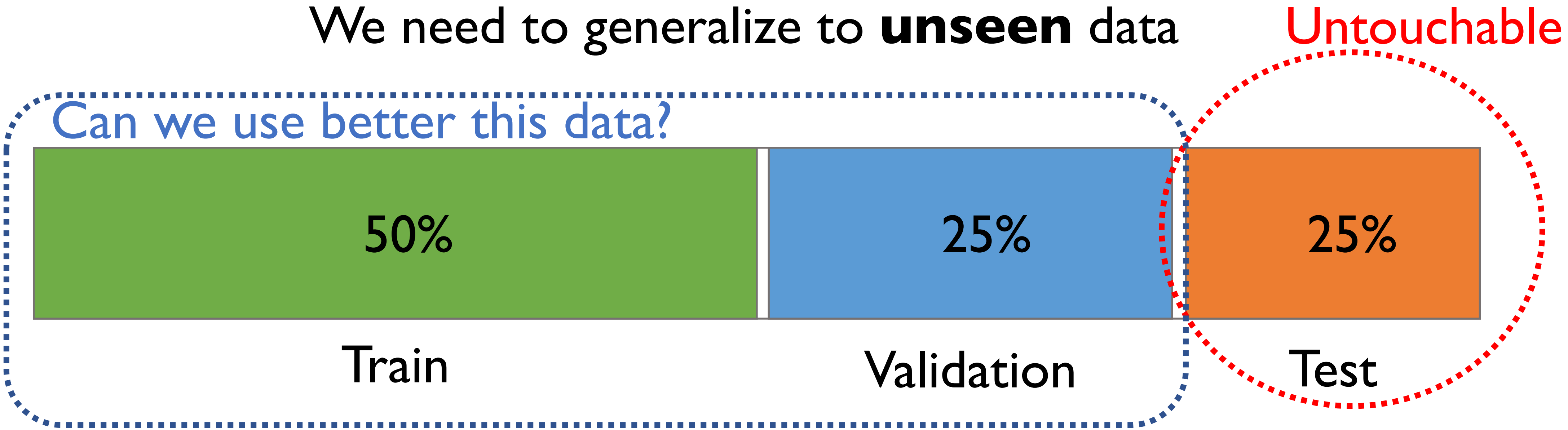


Challenge: what if we don't have enough data

More data → Less variance → Better prediction error

How to estimate prediction error?

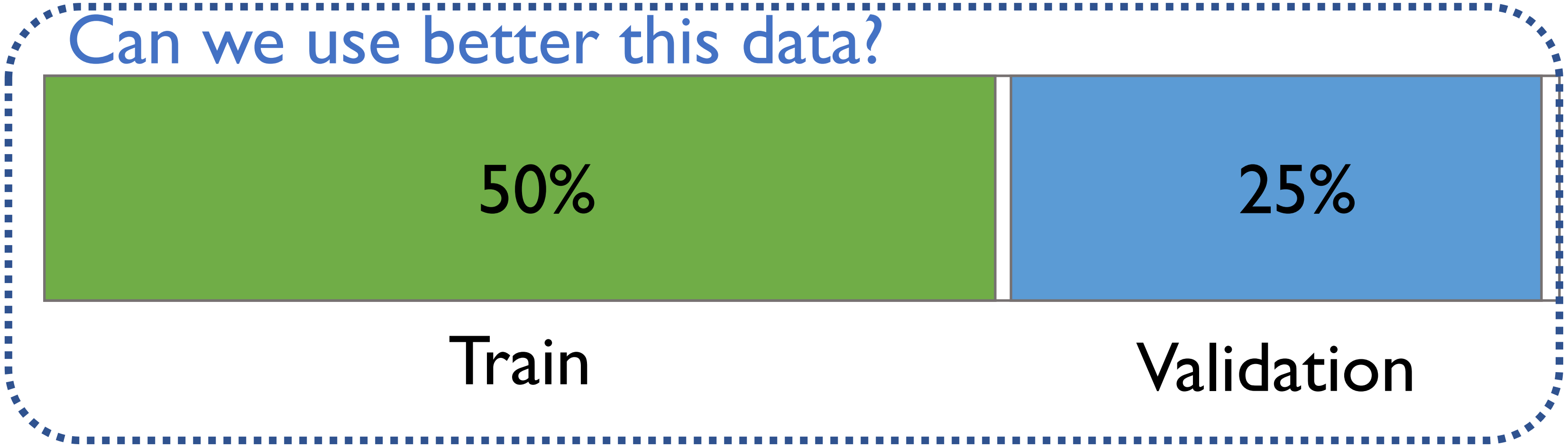
We need to generalize to **unseen** data



Challenge: what if we don't have enough data

More data ⟶ Less variance ⟶ Better prediction error

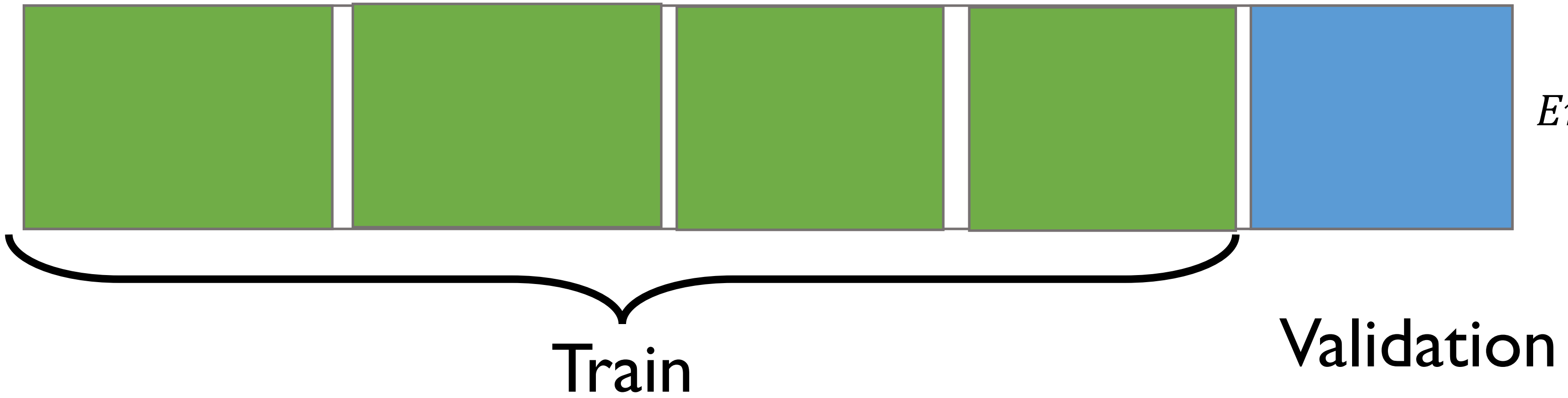
K-fold Cross validation



Challenge: what if we don't have enough data

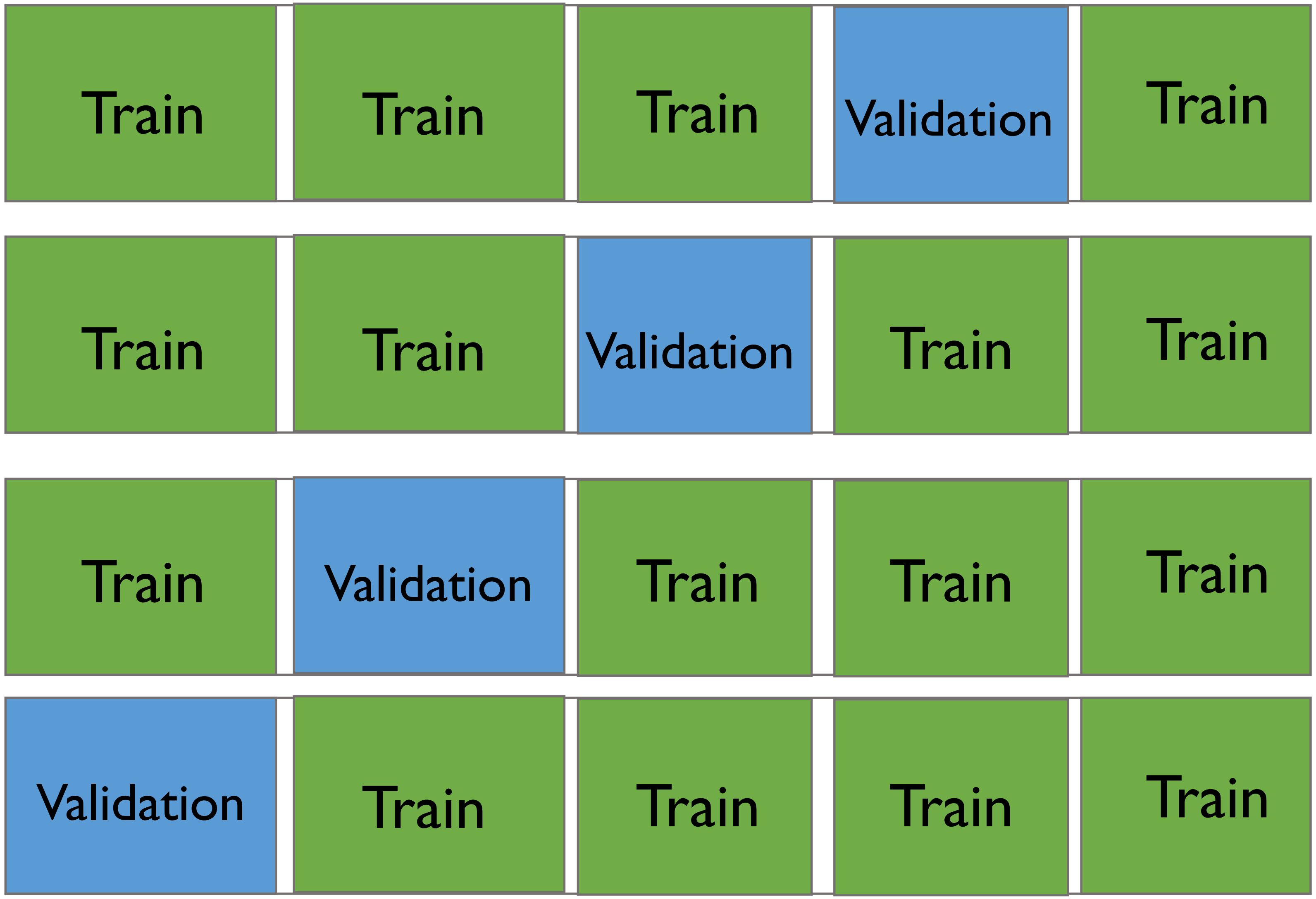
More data → Less variance → Better prediction error

K-fold Cross validation



$$Err_5 = \sum_{i \in S_5} \left(y^{(i)} - \widehat{f_5}\left(x^{(i)}\right) \right)^2$$

K-fold Cross validation



$$Err_4 = \sum_{i \in S_4} \left(y^{(i)} - \hat{f}_4\left(x^{(i)}\right)\right)^2$$

$$Err_3 = \sum_{i \in S_3} \left(y^{(i)} - \hat{f}_3\left(x^{(i)}\right)\right)^2$$

$$Err_2 = \sum_{i \in S_2} \left(y^{(i)} - \hat{f}_2\left(x^{(i)}\right)\right)^2$$

$$Err_1 = \sum_{i \in S_1} \left(y^{(i)} - \hat{f}_1\left(x^{(i)}\right)\right)^2$$

K-fold Cross validation

$$Err = \frac{1}{5}(Err_1 + Err_2 + Err_3 + Err_4 + Err_5)$$

Average of the errors reduces the variation of the prediction error

Better than just having a fixed train and validation set

K-fold Cross validation: How large k?

Leave-one-out (LOOCV)
K = N

High variance!
We are averaging over N models
But all **highly correlated**

Small K =5 or 10

Less Overlap!
We are averaging over less models
that are **less correlated**

Bonus: Less computationally expensive

K-fold Cross validation: Important Note

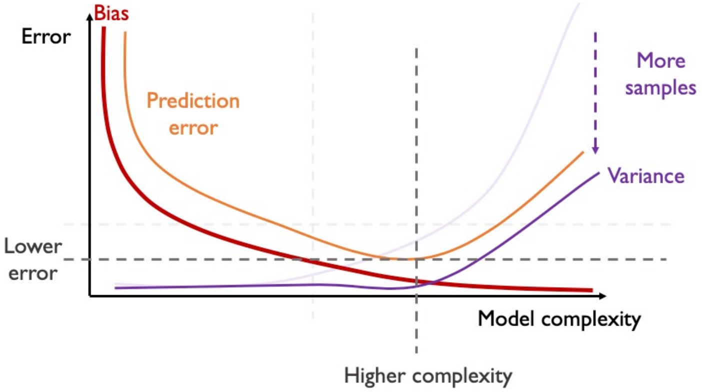
Once we have **selected the best** model / hyperparameter looking at the smallest error in **cross validation**

We train the selected model using train + validation data and we evaluate test error

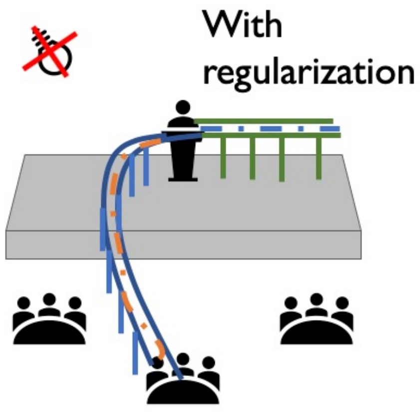
We cannot change the model any more!

Today’s recap

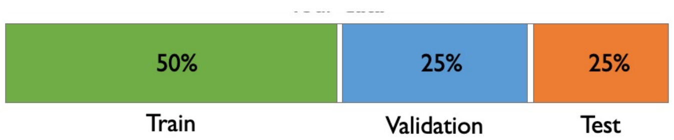
1) We understand prediction error



2) We can control model complexity



3) We can estimate prediction error



Cross-Validation