

# Risk Minimization

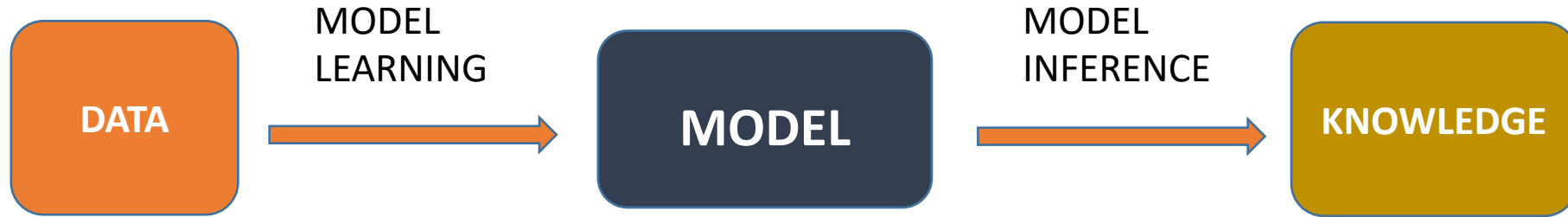
Prof. Pradeep Ravikumar

[pradeepr@cs.cmu.edu](mailto:pradeepr@cs.cmu.edu)

# Model to Knowledge

- You know how to learn a model from data, with guarantees
- How do we go from model to knowledge?
- i.e. How do we get the answers we seek from the model?
- E.g. Recall “coin flip” example
  - The model is the Bernoulli distribution
  - The Billionaire might not care about Bernoulli distribution per se, as much as answers to questions such as:
    - Which side is more likely in the next flip?
    - If a bookie gives 3 to 5 odds on tails, should he take the bet?

# Recall: Model-based ML



- Learning: From data to model
  - A model explains how the data was generated
  - **E.g. given (symptoms, diseases) data, a model explains how symptoms and diseases are related**
- Inference: From model to knowledge
  - Given the model, how can we answer questions relevant to us
  - **E.g. given (symptom, disease) model, given some symptoms, what is the disease?**

# Model to Knowledge: Plugin Estimates

- In most cases, the knowledge we seek is a fixed function  **$f(\mathbf{P})$**  of the model  **$\mathbf{P}$**  of the data
  - is the coin fair: Is the bias of coin  $p = \frac{1}{2}$ ?
  - does the coin have better odds than 3/5: Is bias of coin  $p \geq \frac{3}{5}$ ?

# Model to Knowledge: Plugin Estimates

- In most cases, the knowledge we seek is a fixed function  $\mathbf{f}(\mathbf{P})$  of the distribution  $\mathbf{P}$  of the data
  - is the coin fair:  $\mathbb{I}(p == 1/2)$ ?
  - does the coin have better odds than 3/5:  $\mathbb{I}(p >= 3/5)$
- Once we learn a model, we have an estimate of the distribution of the data:  $P_{\hat{\theta}}$
- So we can simply “plugin” the model for the distribution to get our answers:  $f(P_{\hat{\theta}})$
- Is the coin fair:  $\mathbb{I}(\theta == 1/2)$ 
  - Plugin Estimate:  $\mathbb{I}(\hat{\theta} == 1/2)$

# Specification of Knowledge

- In the previous, the specification of what knowledge we were seeking was through an explicit function of the distribution
  - E.g. is the coin fair? Does the coin have better odds than 3/5, etc.
- But such an explicit specification is not always possible
- Think of the knowledge we seek as some “decision” given the underlying data
- QUESTIONS:
  - How do we characterize such decisions?
  - What is the optimal decision we can make?
  - How do we characterize optimality?
  - Falls under **decision theory**

# Example: Finance



STOCK MARKETS									
Price	Chng	52 week High	Low	Yld	P/E	Vol	Stk	Chg	Stk
607.50	-4.50	607.50	421.25	4.1	23.5	20844	ChlAerob	4	10.50
14	1.705	1.316	2.7	12.5	4987	ChlAerob	ChlAerob	11.00	11.00
4.50	1.705	1.316	2.7	12.5	4987	ChlAerob	ChlAerob	11.00	11.00
19.4	88166	19.4	88166	19.4	88166	19.4	88166	19.4	88166
5889	ChlPac	5889	ChlPac	5889	ChlPac	5889	ChlPac	5889	ChlPac
4041	ChlPac	4041	ChlPac	4041	ChlPac	4041	ChlPac	4041	ChlPac
318	ChlPac	318	ChlPac	318	ChlPac	318	ChlPac	318	ChlPac
1437	ChlPac	1437	ChlPac	1437	ChlPac	1437	ChlPac	1437	ChlPac



- Suppose you have a model that specifies the value of apple stock, or whether the stock price will go up in next 1 hour (instead of bias of Billionaire's coins)
- Your set of actions are how much apple stock to buy or sell
- Given the model, you have to minimize some **loss function** that balances risk and reward to decide the action (how much stock to buy or sell).

# Example: Electricity Generation



- Suppose you have a model that predicts consumer demand of electricity
- Your set of actions are based on how to schedule the generators of power plant
- Given the model, have to minimize some loss function to decide which and how and when of the generation of electricity



# Specification of Knowledge

- **In machine learning**, we characterize the decision to take via **loss/utility functions** (also known as **performance measures**)
  - Whenever you encounter a task, you should automatically think about the appropriate **performance measure/loss function**

# From model to knowledge: the general case

Given model parameter  $\theta$ , we then pick our action  $a$  from a set  $A$  by solving a decision-theoretic optimization problem:

$$\min_{a \in A} \ell(\theta, a).$$

Here  $\ell(\theta, a)$  is the loss of taking action  $a$  when model parameter is  $\theta$ .

# Unsupervised vs Supervised

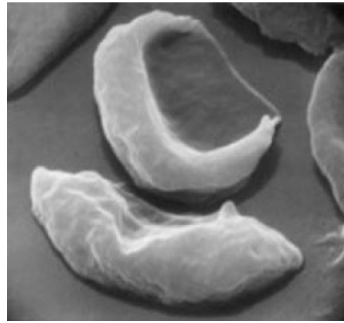
- Learning a Bernoulli distribution as the model for a sequence of coin flips is an example of an “unsupervised learning” problem
- In a “supervised learning” problem, you have an input and an output, and the goal is to predict an output **given an input**
- Coin Flips: Predict coin flips given “features” about coin
- Finance: predict how much apple stock will move up given economic indicators
- Electricity Generation: predict consumer demand given past demand, weather

# Supervised Learning, and Going from Model to Knowledge

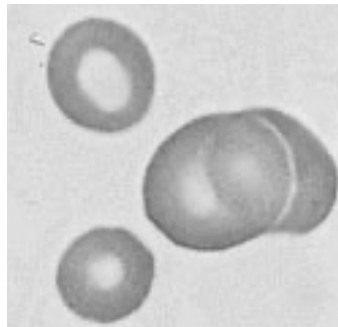
- For supervised learning problems, it is natural to talk about the knowledge first, and model second
- In particular, a **loss function** is part of the problem specification in supervised learning

# Supervised Learning Prediction Task

**Task:** Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .  
 $\equiv$  Construct **prediction rule**  $f : \mathcal{X} \rightarrow \mathcal{Y}$



“Lupus cell (0)”



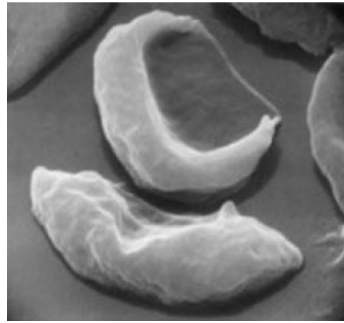
“Healthy cell (1)”

# Example: Supervised Learning Prediction Task

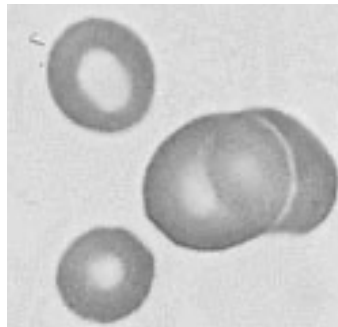
## Task:

Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .

$\equiv$  Construct **prediction rule**  $f : \mathcal{X} \rightarrow \mathcal{Y}$



“Lupus cell (0)”



“Healthy cell (1)”

But I can always come up with a prediction rule: always say it's not LUPUS!

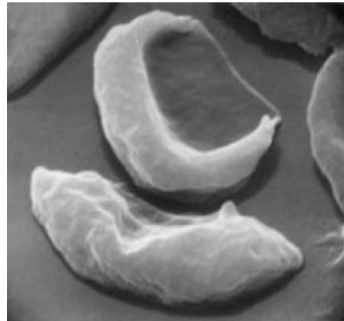


# Example: Supervised Learning Prediction Task

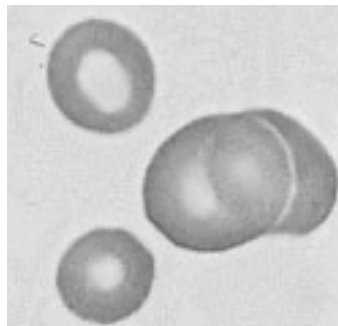
## Task:

Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .

$\equiv$  Construct **prediction rule**  $f : \mathcal{X} \rightarrow \mathcal{Y}$



“Lupus (0)”



“Healthy (1)”

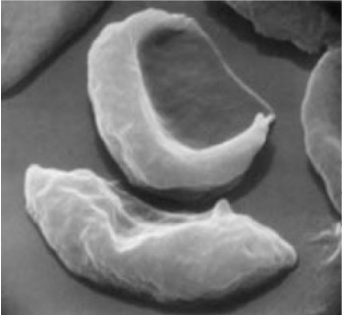
To complete the specification of the task, we need something more!!!

# Characterize Task using Performance Measures

## Performance Measure:

What is the “loss” I suffer when I take decision  $\mathbf{f}$ ?

$\text{loss}(Y, f(X))$  - Measure of closeness between true label  $Y$  and prediction  $f(X)$

$X$	$Y$	$f(X)$	$\text{loss}(Y, f(X))$
	“Lupus”	“Lupus”	0
		“Healthy”	1

$$\text{loss}(Y, f(X)) = 1_{\{f(X) \neq Y\}} \quad \text{0/1 loss}$$



# Characterize Task using Performance Measures

## Performance Measure:

What is the “loss” I suffer when I take decision  $f$ ?

$\text{loss}(Y, f(X))$  - Measure of closeness between true label  $Y$  and prediction  $f(X)$

$X$	Share price, $Y$	$f(X)$	$\text{loss}(Y, f(X))$
Past performance, trade volume etc. as of Sept 8, 2010	“\$24.50”	“\$24.50”	0
		“\$26.00”	1?
		“\$26.10”	2?

$$\text{loss}(Y, f(X)) = (f(X) - Y)^2 \quad \text{square loss}$$

# Performance Measures

**Performance:**

**Measure:**

$\text{loss}(Y, f(X))$  - Measure of closeness between true label  $Y$  and prediction  $f(X)$

We don't just want to correctly label one test sample (in this case, cell image), but most cell images  $X \in \mathcal{X}$

Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

$$\text{Risk } R(f) \equiv \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$$

# Performance Measures

**Performance:**

**Measure:**

$\text{loss}(Y, f(X))$  - Measure of closeness between true label  $Y$  and prediction  $f(X)$

We don't just want to correctly label one test sample (in this case, cell image), but most cell images  $X \in \mathcal{X}$

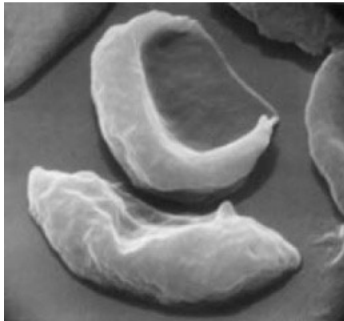
Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

What is the  
“risk” of taking  
decision  $\mathbf{f}$ ?

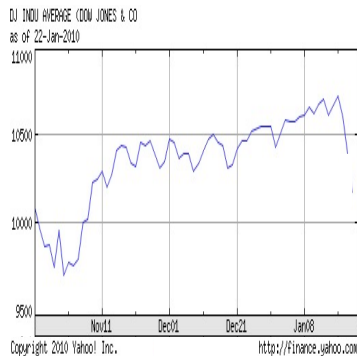
$$\text{Risk } R(f) \equiv \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$$

# Performance Measures

**Performance Measure:** Risk  $R(f) \equiv \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$



➞ “Anemic cell”



➞ Share Price  
“\$ 24.50”

$\text{loss}(Y, f(X))$	Risk $R(f)$
$1_{\{f(X) \neq Y\}}$ <b>0/1 loss</b>	$P(f(X) \neq Y)$ <b>Probability of Error</b>
$(f(X) - Y)^2$ <b>square loss</b>	$\mathbb{E}[(f(X) - Y)^2]$ <b>Mean Square Error</b>

# Bayes Optimal Rule

Knowledge

Construct **prediction rule**  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

That we seek:

$$f^*(P) = \arg \min_f \mathbb{E}_{(X,Y) \sim P} [\text{loss}(Y, f(X))]$$

We refer to this best possible rule as  
**the Bayes optimal rule**

Best possible performance:

**Bayes Risk:**  $R(f^*) \leq R(f) \text{ for all } f$

# Model-free Methods

Knowledge

Construct **prediction rule**  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

That we seek:

$$f^*(P) = \arg \min_f \mathbb{E}_{(X,Y) \sim P} [\text{loss}(Y, f(X))]$$

Bayes optimal rule

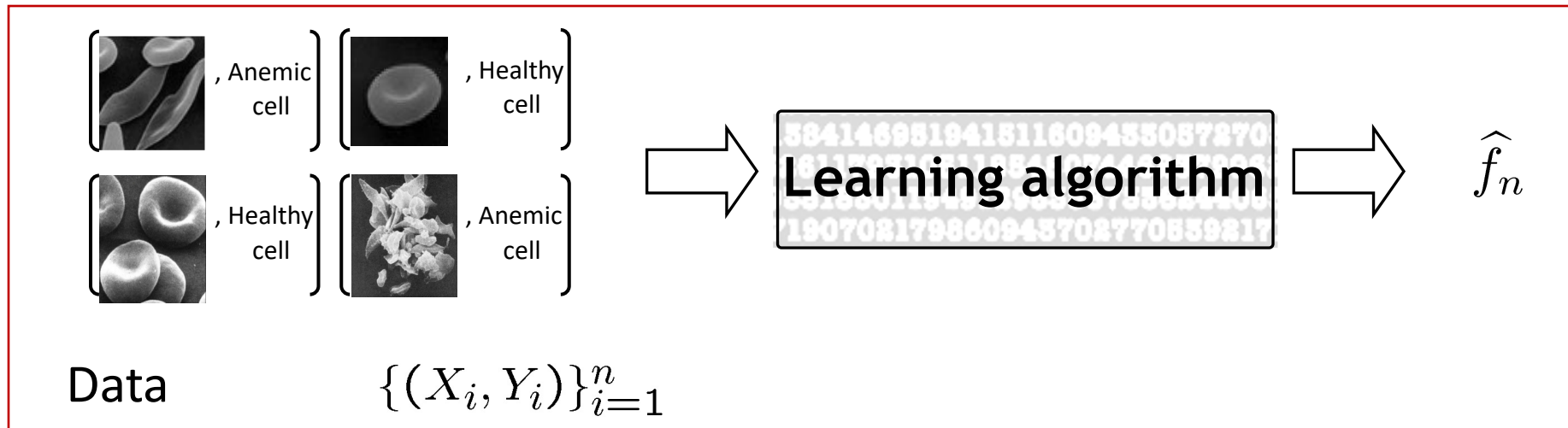
**Optimal rule is not computable**

- depends on unknown distribution  $P$  over  $(X,Y)$  !

**MODEL BASED METHODS:** Use a model for  $P_{XY}$  !

**MODEL-FREE METHODS:** Estimate the knowledge through some learning algorithm that does not go through a model for  $P_{XY}$

# Model-free Methods



$\hat{f}_n$  is a mapping from  $\mathcal{X} \rightarrow \mathcal{Y}$

$\hat{f}_n \left[ \begin{array}{c} \text{Image of two anemic cells} \end{array} \right] = \text{"Anemic cell"}$   
 Test data  $X$

# Popular Approach for model-free ML: Empirical Risk Minimization

Knowledge

Construct **prediction rule**  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

That we seek:

$$f^*(P) = \arg \min_f \mathbb{E}_{(X,Y) \sim P} [\text{loss}(Y, f(X))] \quad \text{Bayes optimal rule}$$

Given  $\{X_i, Y_i\}_{i=1}^n$ , **learn** prediction rule  
 $\hat{f}_n : \mathcal{X} \rightarrow \mathcal{Y}$

Empirical Risk

Minimizer:

$$\hat{f}_n = \arg \min_f \frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(X_i))]$$

$$\frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(X_i))] \xrightarrow[\text{Numbers}]{\text{Law of Large}} \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$$



# Empirical Risk Minimization

- Very Popular Approach in ML
- Given a loss function, and data, estimate decision function by minimizing “empirical risk”
- Typically restrict decision to lie within some restricted set
  - Could capture our prior information
  - Or just be for computational convenience

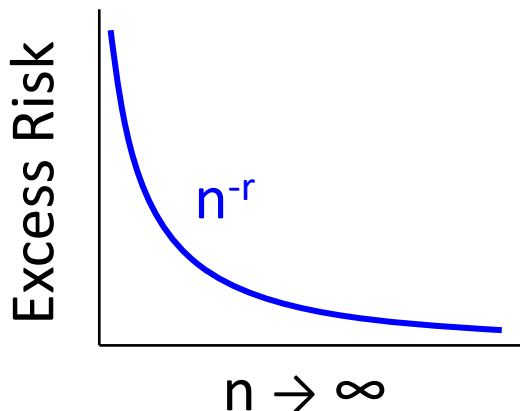
$$\hat{f} = \arg \inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^n \text{loss}(Y_i, f(X_i)) \right\}$$

# Statistical Considerations: Consistency and Rate of Convergence

- How does the performance of the algorithm compare with ideal performance?

$$\text{Excess Risk} \quad \mathbb{E}_{D_n} [R(\hat{f}_n)] - R(f^*)$$

- **Consistent** algorithm if Excess Risk  $\rightarrow 0$  as  $n \rightarrow \infty$
- **Rate of Convergence**



# True Risk vs. Empirical Risk

True Risk: Performance on “unseen” data

Classification – Probability of misclassification  $P(f(X) \neq Y)$

Regression – Mean Squared Error  $\mathbb{E}[(f(X) - Y)^2]$

# True Risk vs. Empirical Risk

True Risk: Performance on “unseen” data

Classification – Probability of misclassification  $P(f(X) \neq Y)$

Regression – Mean Squared Error  $\mathbb{E}[(f(X) - Y)^2]$

Empirical Risk: Performance on training data

Classification – Proportion of misclassified examples  $\frac{1}{n} \sum_{i=1}^n 1_{f(X_i) \neq Y_i}$

Regression – Average Squared Error  $\frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2$

# True Risk vs Empirical Risk

- Example: E-commerce: Ad-click prediction based on demographics
  - We have demographic information on people who have clicked before but what about new people?
- Example: Biology: Predict from DNA whether a person is predisposed to diabetes
  - We know that certain people eventually developed diabetes, but what about a child or young adult who has not develop diabetes yet?
- What we really care about is NOT predicting the training data but predicting *unseen* data.

# True Risk vs Empirical Risk

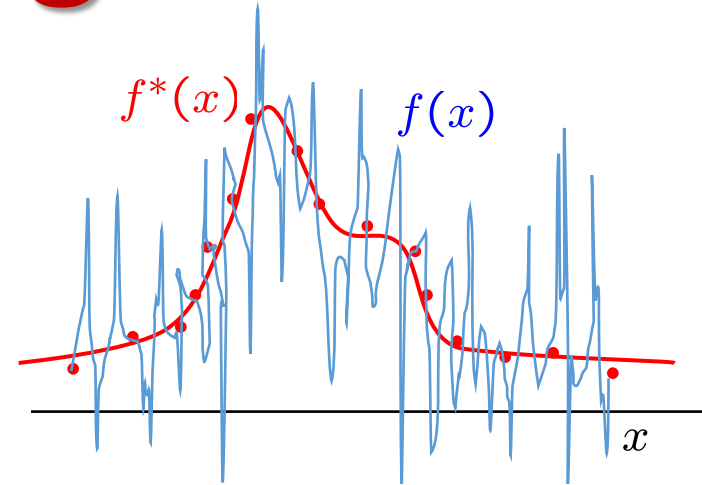
- So we minimize with respect to empirical risk
- But we care about true risk
- Is there any danger to this mismatch?
  - Overfitting!!



# Overfitting

Is the following predictor a good one?

$$f(x) = \begin{cases} Y_i, & x = X_i \text{ for } i = 1, \dots, n \\ \text{any value,} & \text{otherwise} \end{cases}$$



What is its empirical risk? (performance on training data)

zero !

What about true risk?

>> zero

Will predict very poorly on new random test point:

Large generalization error !

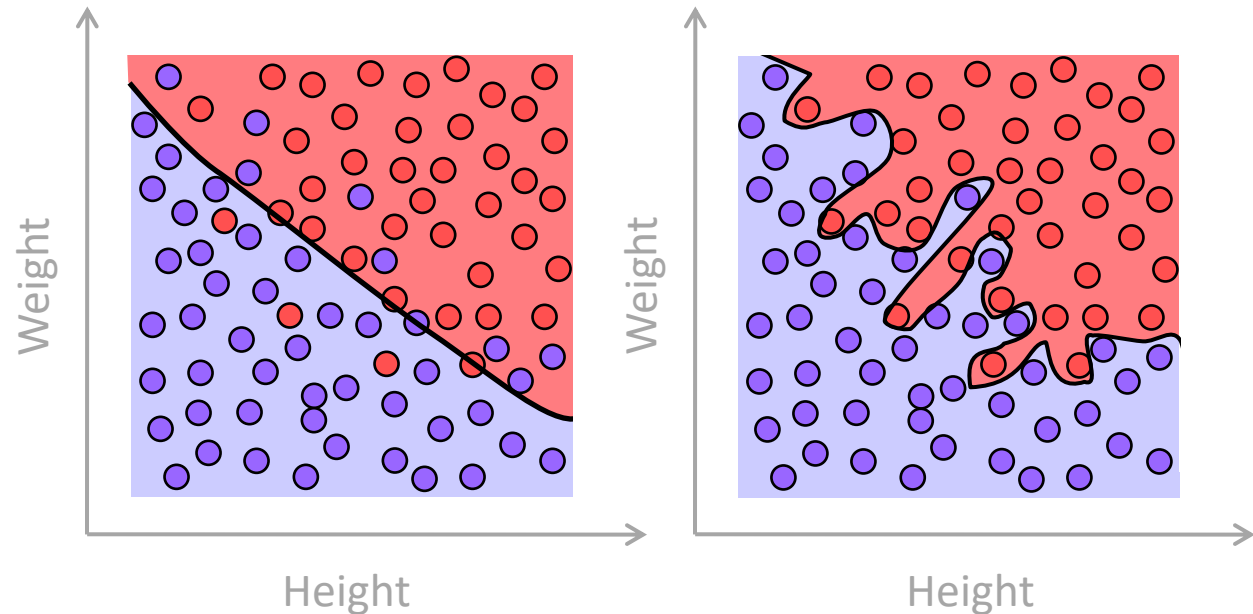
# Overfitting

If we allow very complicated predictors, we could overfit the training data.

Examples: Classification

Football player ?

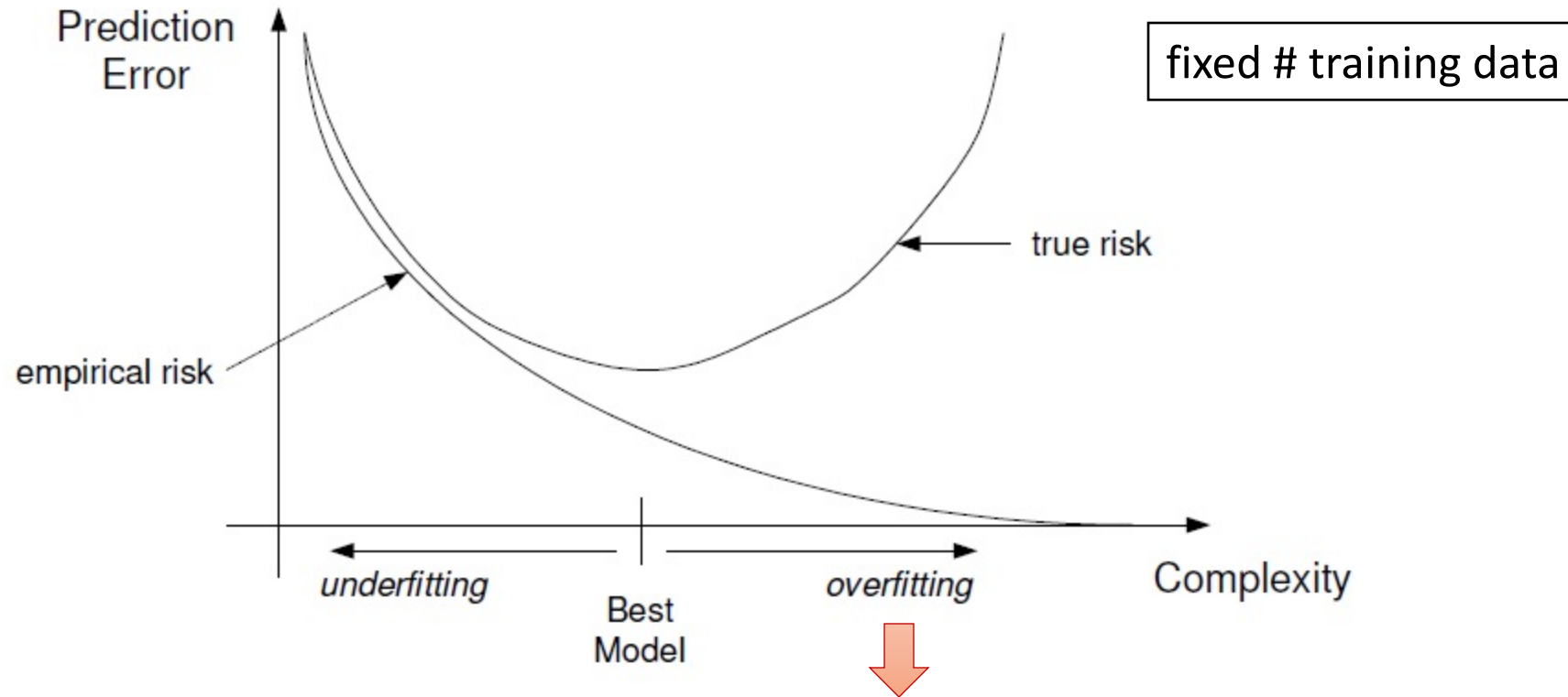
- No
- Yes





# Overfitting: Effect of difference between empirical and true risks

If we allow very complicated predictors, we could overfit the training data.



Empirical risk is no longer a good indicator of true risk