



# Parameter Estimation

Chris Piech

CS109, Stanford University

# MACHINE LEARNING



<Review>

# General “Inference”



# General “Inference”

WebMD Symptom Checker BETA

INFO

SYMPTOMS

QUESTIONS

CONDITIONS

DETAILS

TREATMENT

## Add more symptoms

Type your main symptom here

or Choose common symptoms

bloating

cough

diarrhea

dizziness

fatigue

fever

headache

muscle cramp

nausea

throat irritation

AGE 30

GENDER Male

MY SYMPTOMS

cough ×

throat irritation ×

sneezing ×

Results Strength: **MODERATE**

< Previous

Continue >

Info



## Conditions that match your symptoms

UNDERSTANDING YOUR RESULTS 

### Influenza (flu) adults



Moderate match



### Pneumococcal Infections



Moderate match



### H1N1 Flu Virus (Swine Flu)



Moderate match



### Bacterial Pneumonia



Moderate match



### Sepsis (blood infection)



Moderate match



Gender **Male**

Age **30**

[Edit](#)

My Symptoms

[Edit](#)

fever 103f to 104f, dizziness,

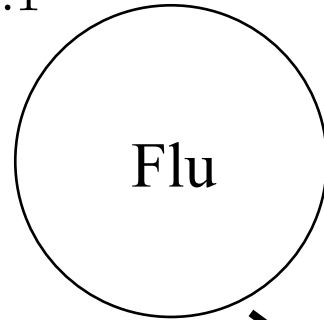
throat irritation, migraine headache



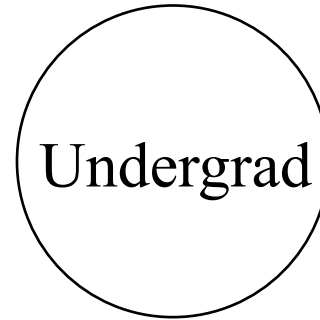
Start Over

# Probabilistic Model

$$P(Fl = 1) = 0.1$$



$$P(U = 1) = 0.8$$



Flu

Undergrad

$$Fev|Flu = 0 \sim N(100.0, 1.81)$$

$$Fev|Flu = 1 \sim N(98.25, 0.73)$$



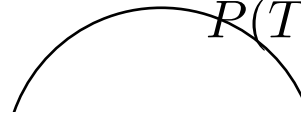
Fever

$$P(T = 1|Flu = 0, U = 0) = 0.1$$

$$P(T = 1|Flu = 0, U = 1) = 0.8$$

$$P(T = 1|Flu = 1, U = 0) = 0.9$$

$$P(T = 1|Flu = 1, U = 1) = 1.0$$



If you know the probability of each random variables given the ones that directly cause it, you can joint sample!



# Alg #1: Joint Sampling

```
3 N_SAMPLES = 100000
4
5 # Program: Joint Sa
6 # -----
7 # we can answer any
8 # with multivariate
9 # where conditioned
10 def main():
11     obs = getObserv
12     print 'Observat
13
14     samples = sampl
15     prob = probFluG
16     print 'Pr(Flu)
```

```
webMd -- bash -- 38x22
[0, 0, 0, 0]
[0, 1, 0, 1]
[1, 0, 1, 0]
[1, 1, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 1, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[1, 1, 1, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[1, 1, 1, 1]
[0, 1, 0, 0]
Observation = [None, None, None, 1]
Pr(Flu | Obs) = 0.140635888502
>
```

Each one of these is one posterior sample:

[Flu, Ugrad, Fever, Tired]



# Alg #2: MCMC

```
webid --bash -- 10x20
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[0, 1, 101.0, 0]
[0, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
Pr(Flu) = 0.9773
>
```

MCMC is a way to sample with conditioned variables fixed

Each one of these is one posterior sample:

[Flu, Undergrad, Fever, Tired]



</ Review >

# BAE's Theorem?

$$P(A | B E) = \frac{P(B | A E) P(A | E)}{P(B | E)}$$



# Alg #3: Straight Math

Know:  $P(\text{symptom} \mid \text{flu}, \text{undergrad})$        $P(\text{flu})$        $P(\text{undergrad})$

Flu is independent of undergrad

Tired and fever are conditionally independent given flu, undergrad

---

$$\begin{aligned} & P(F = 1 \mid \text{all other rvs}) \\ &= P(F = 1 \mid \text{symptoms}, U = u) \\ &= \frac{P(\text{symptoms} \mid F = 1, U = u) P(F = 1 \mid U = u)}{P(\text{symptoms} \mid U = u)} \\ &\propto P(\text{symptoms} \mid F = 1, U = u) P(F = 1 \mid U = u) \\ &\propto P(F = 1) P(\text{symptoms} \mid F = 1, U = u) \\ &\propto P(F = 1) \prod_i P(\text{symptom}_i \mid F = 1, U = u) \end{aligned}$$

# $P(F = 1 \mid \text{all other rvs})$

Know:  $P(\text{symptom} \mid \text{flu}, \text{undergrad})$        $P(\text{flu})$        $P(\text{undergrad})$

Flu is independent of undergrad

Tired and fever are conditionally independent given flu, undergrad

---

$$\begin{aligned} & P(F = 1 \mid \text{all other rvs}) \\ &= P(F = 1 \mid \text{symptoms}, U = u) \\ &= \frac{P(\text{symptoms} \mid F = 1, U = u) P(F = 1 \mid U = u)}{P(\text{symptoms} \mid U = u)} \\ &\propto P(\text{symptoms} \mid F = 1, U = u) P(F = 1 \mid U = u) \\ &\propto P(F = 1) P(\text{symptoms} \mid F = 1, U = u) \\ &\propto P(F = 1) \prod_i P(\text{symptom}_i \mid F = 1, U = u) \end{aligned}$$



$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```

$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```

$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```



$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```

$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```



$$P(F = 1 | \text{all other rvs}) \propto P(F = 1) \prod_i P(\text{symptom}_i | F = 1, U = u)$$

```
120 def sampleFlu(sample):
121     f1 = getFluPr1(sample)
122     f0 = getFluPr0(sample)
123     p1 = f1 / (f1 + f0)
124     return bern(p1)
125
126 def getFluPr0(sample):
127     [_ , und, fev, tir] = sample
128     pFlu0 = 0.9
129     pFev = getPrFeverX(fev, flu=0)
130     pTir = getPrTiredX(tir, und=und, flu=0)
131     return pFlu0 * pFev * pTir
132
133 def getFluPr1(sample):
134     [_ , und, fev, tir] = sample
135     pFlu1 = 0.1
136     pFev = getPrFeverX(fev, flu=1)
137     pTir = getPrTiredX(tir, und=und, flu=1)
138     return pFlu1 * pFev * pTir
```

See you soon!

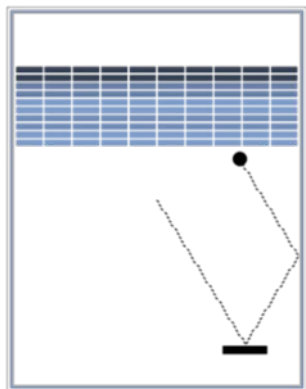
# Summary

# General Inference Summary



- **Straight Math** is fast, but can be prohibitively hard for complex models (see hw).
- **Joint Sampling** is really easy to program but fails for continuous variables (and when what you are conditioning on is rare)
- **MCMC** works well when conditioning on rare events, but is *much* harder to code / derive.
- All sampling is **slow**.

# Insight



Let  $x$  be a students program

Let  $y$  be student mistakes

## Label Console

✓ Num Done: 8273

### Strategy

- Beeper Boundry (most people do this)
- Triangle Strategy
- Recursive Strategy

### Looping

- Correct use of looping
- Doesn't use a while
- Doesn't have correct stop condition
- Body is missing statements
- Body has extra statements
- Body order is incorrect
- Sets up initial precondition
- Does not get nesting
- Loop post condition doesn't match precondition
- Repetition of bodies

Feedback task:

$$P(y|x)$$

Hard for humans  
Hard for computers

Joint sample:

$$(\hat{x}, \hat{y}) \sim P(x, y)$$

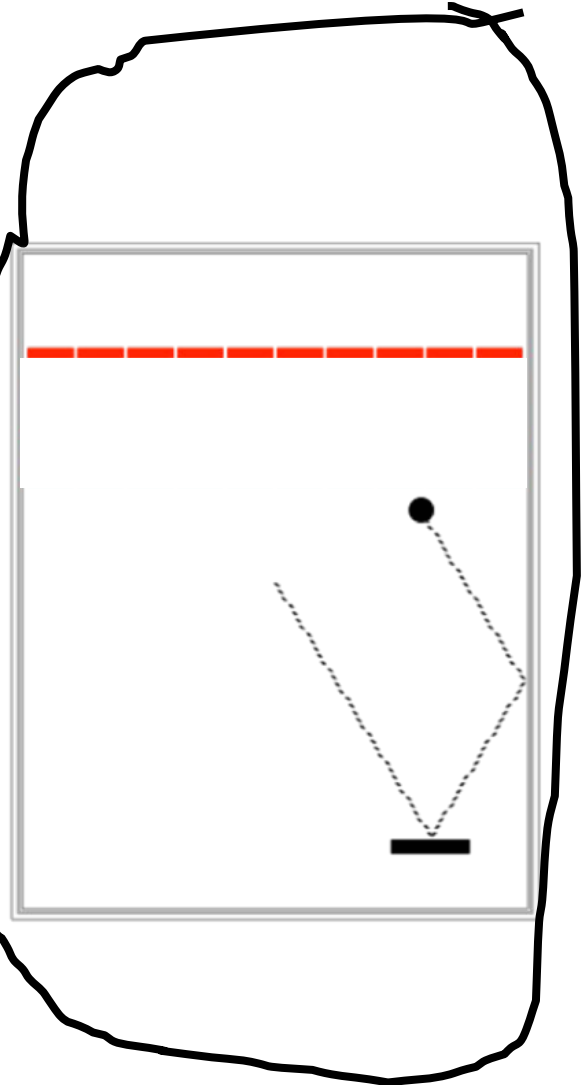
Easy for humans



# Imagine Students

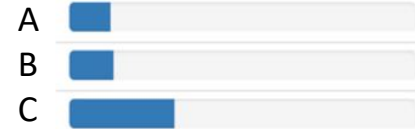
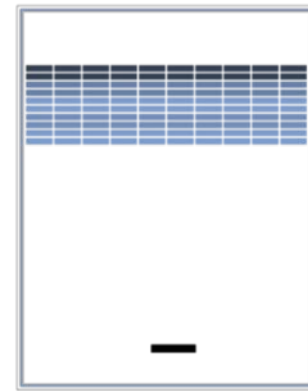
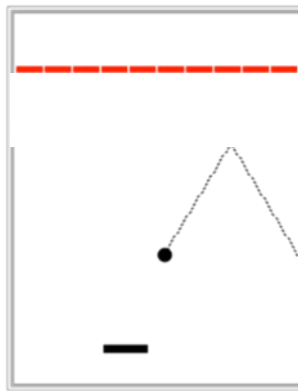
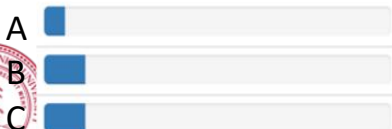
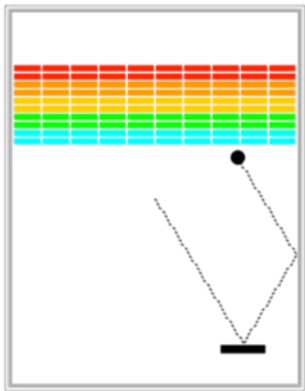
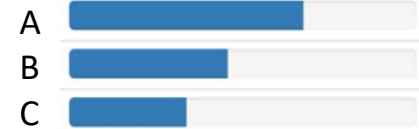
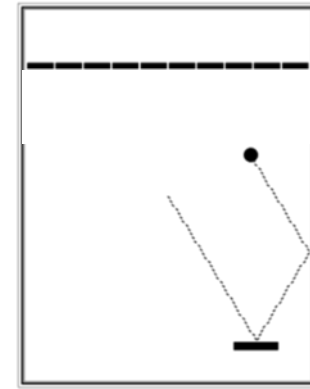
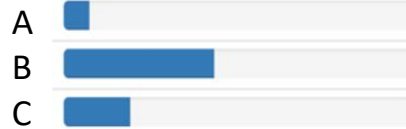
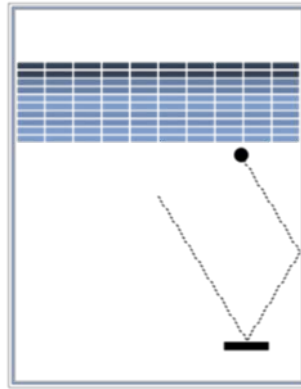
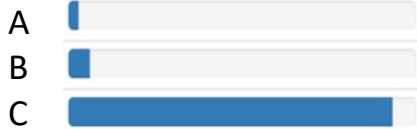
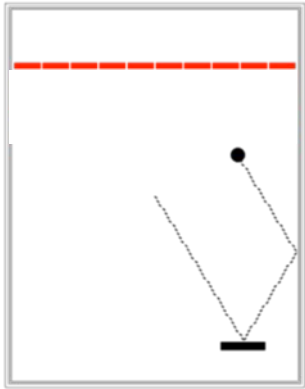
$$(\hat{x}, \hat{y}) \sim P(x, y)$$

- Struggle with double for loops
- Confuses logic for deleting bricks

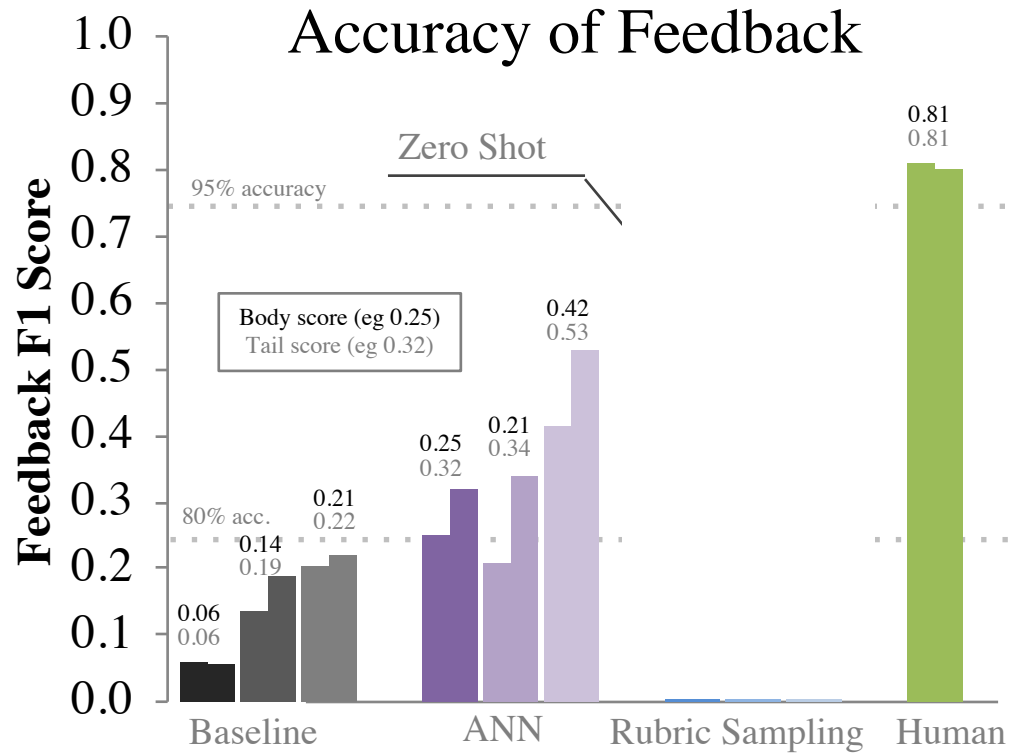




# Start Imagining Students



# Zero Shot Learning



Majority Vote

Syntactic Analysis

Program Output \*

FNN

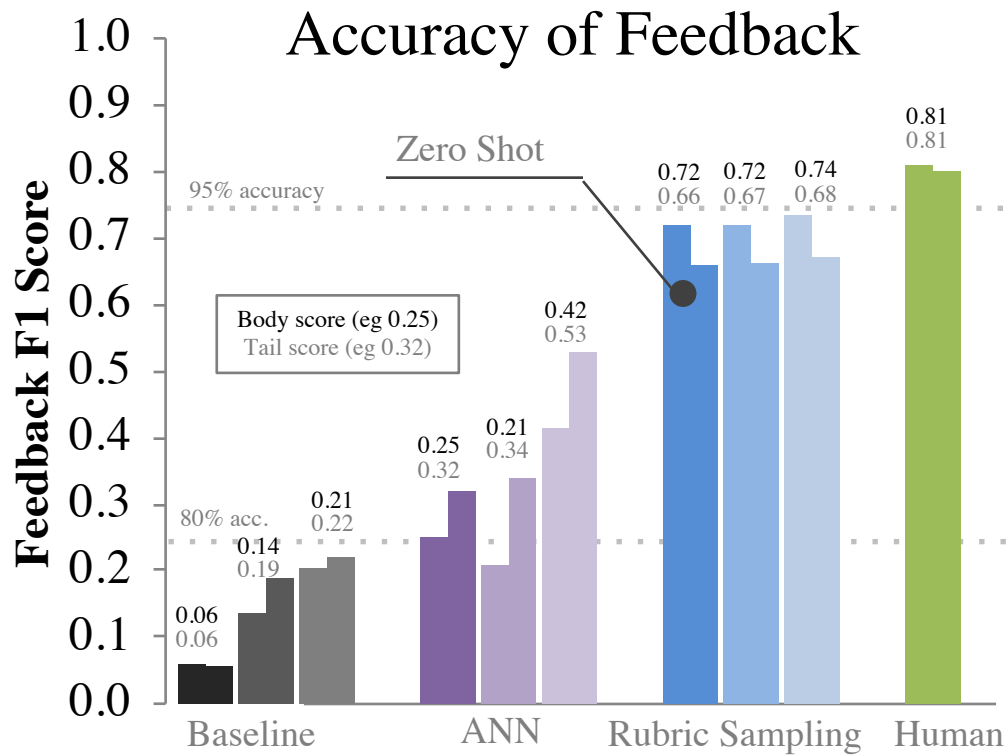
RNN

MVAE

Expert Human



# Zero Shot Learning



- Majority Vote
- Syntactic Analysis
- Program Output \*
- FNN
- RNN
- MVAE
- Rubric Sampling, Zero shot
- Rubric Sampling, Learned  $\theta$
- Rubric Sampling, MVAE
- Expert Human

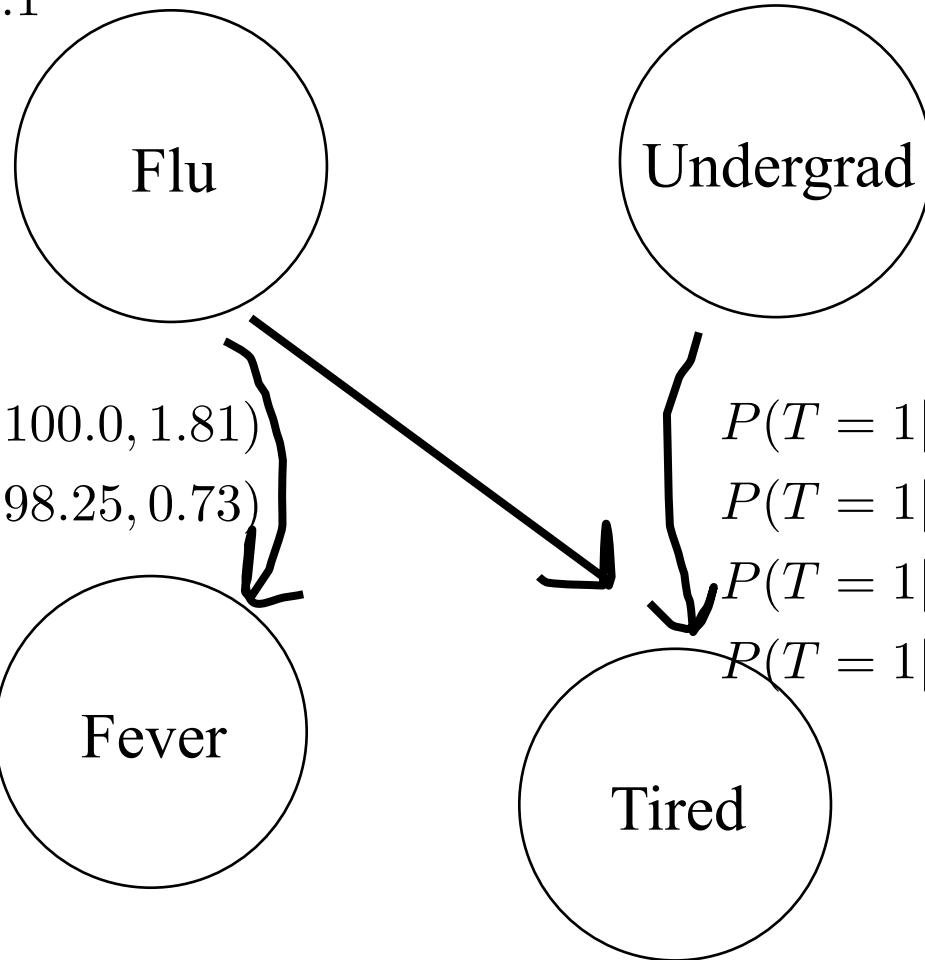
Model	Amount of Correct Feedback
Predicting from output	1,483,157 (86.0%)
Rubric sampling with MVAE	<b>1,610,020 (93.7%)</b>
Expert human	1,658,162 (96.2%)



# Where Do The Numbers Come From?

$$P(Fl = 1) = 0.1$$

$$P(U = 1) = 0.8$$



$$Fev|Flu = 0 \sim N(100.0, 1.81)$$

$$Fev|Flu = 1 \sim N(98.25, 0.73)$$

$$P(T = 1|Flu = 0, U = 0) = 0.1$$

$$P(T = 1|Flu = 0, U = 1) = 0.8$$

$$P(T = 1|Flu = 1, U = 0) = 0.9$$

$$P(T = 1|Flu = 1, U = 1) = 1.0$$

Pedagogical Pause

At this point, if you are given a *model*,  
with all the involved probabilities, you  
can make predictions



But what if you want to *learn* the probabilities in the model?

But what if you want to *learn* the probabilities in the model?

Oh can we also learn the *structure* of the model too?

But what if you want to *learn* the probabilities in the model?

~~Oh can we also learn the *structure* of the model too?~~

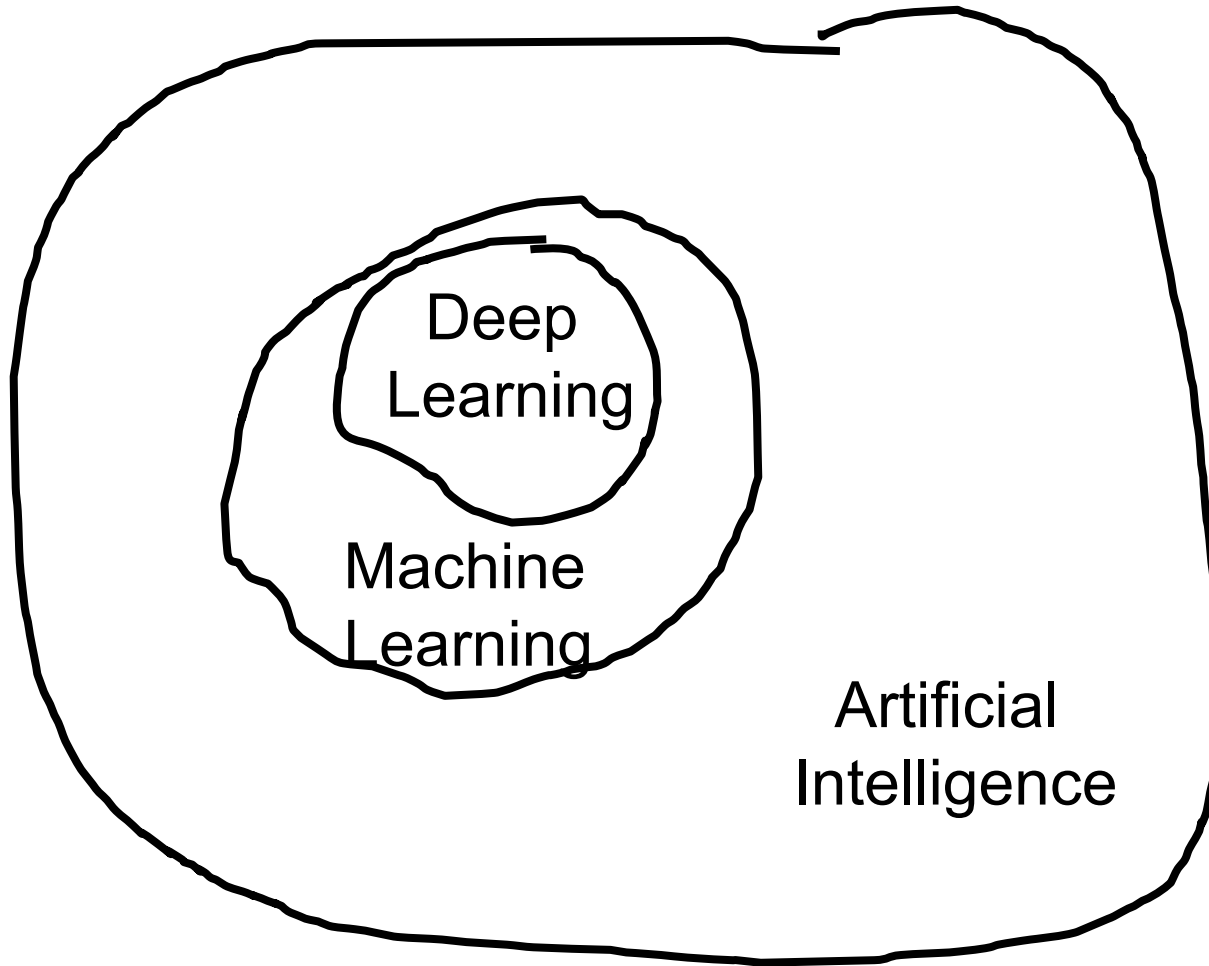
I wish. Another day 😊

But what if you want to *learn* the probabilities in the model?

# Machine Learning



# AI and Machine Learning



ML: Rooted in probability theory

# Our Path

Deep Learning

Linear  
Regression

Naive  
Bayes

Logistic  
Regression

Parameter Estimation

# Our Path

Deep Learning

Linear  
Regression

Naive  
Bayes

Logistic  
Regression

Unbiased  
estimators

Maximizing  
likelihood

Bayesian  
estimation

# Jump Straight to Deep Learning?

Tensor Flow



# Jump Straight to Deep Learning?





Understand the theory to help you debug

But another reason...

# Machine Learning Uses a Lot of Data



# One Shot Learning

Single training example:

अ

Test set:

व  
शु  
अ  
स  
र  
त  
अ  
य  
क  
कि  
ह  
शु  
अ  
न

# One Shot Learning

Single  
training  
example:



Computers struggle...

... especially for **human** problems.



Understand the theory  
to push on the **grand challenges**

The image features the iconic Walt Disney Pictures logo. The background is a vibrant, stylized illustration of Cinderella Castle at night, illuminated with warm golden lights. The castle is set against a deep blue sky with wispy, pink-tinged clouds and a starry night sky. The castle's reflection is visible in the dark water in the foreground. The logo itself is rendered in a classic, white, cursive script. The word "WALT" is in a smaller, simpler font, while "DISNEY" is in a larger, more ornate script. Below "DISNEY", the word "PICTURES" is written in a clean, white, sans-serif, all-caps font. The entire scene is framed by a dark, starry border.

WALT DISNEY  
PICTURES



Once upon a time...

...there was parameter estimation

# What are Parameters?

- Consider some probability distributions:
  - $\text{Ber}(p)$   $\theta = p$
  - $\text{Poi}(\lambda)$   $\theta = \lambda$
  - $\text{Uni}(\alpha, \beta)$   $\theta = (\alpha, \beta)$
  - $\text{Normal}(\mu, \sigma^2)$   $\theta = (\mu, \sigma^2)$
  - $Y = \mathbf{m}X + \mathbf{b}$   $\theta = (m, b)$
  - etc...
- Call these “parametric models”
- Given model, **parameters** yield actual distribution
  - Usually refer to parameters of distribution as  $\theta$
  - Note that  $\theta$  that can be a vector of parameters

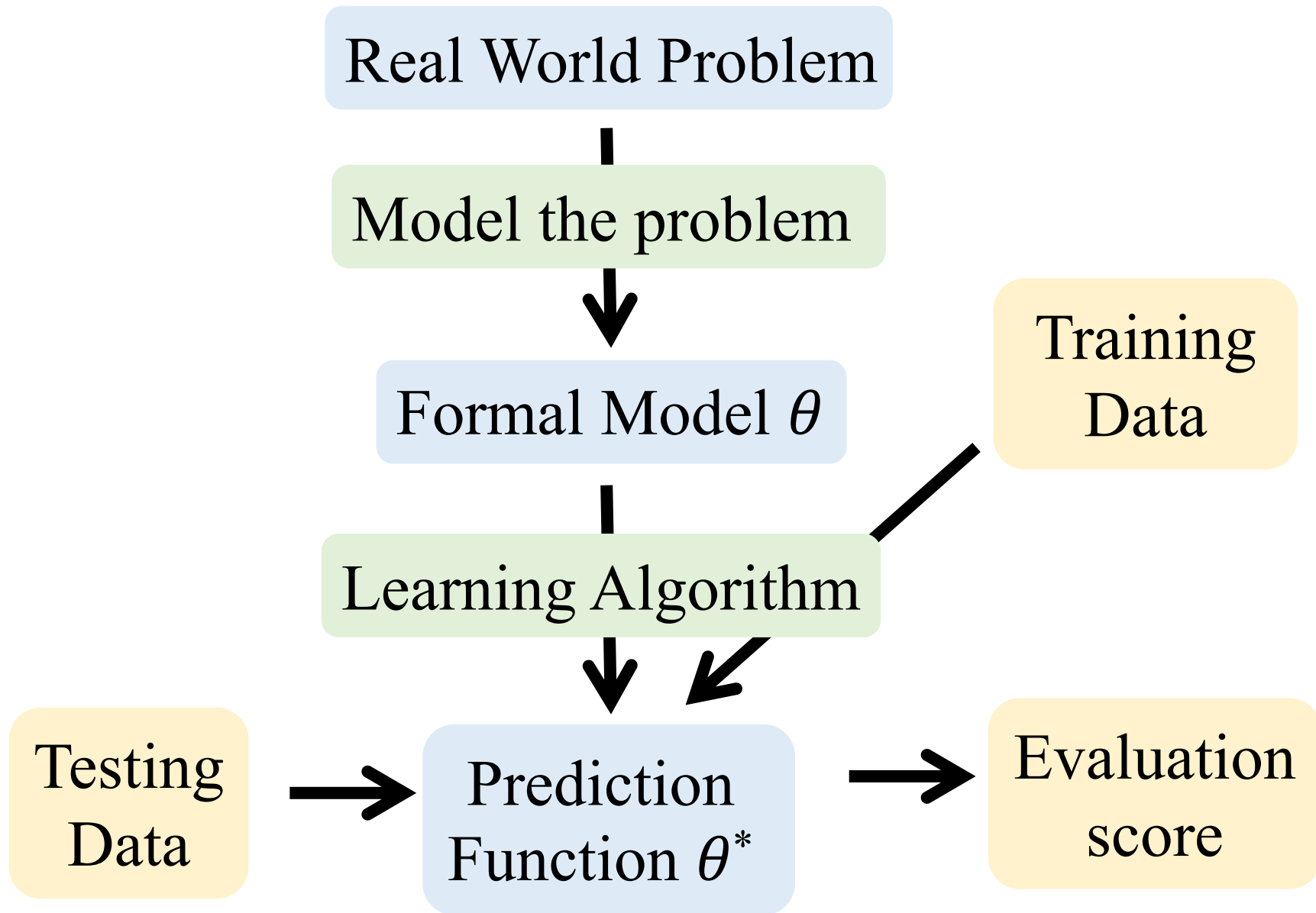


# Why Do We Care?

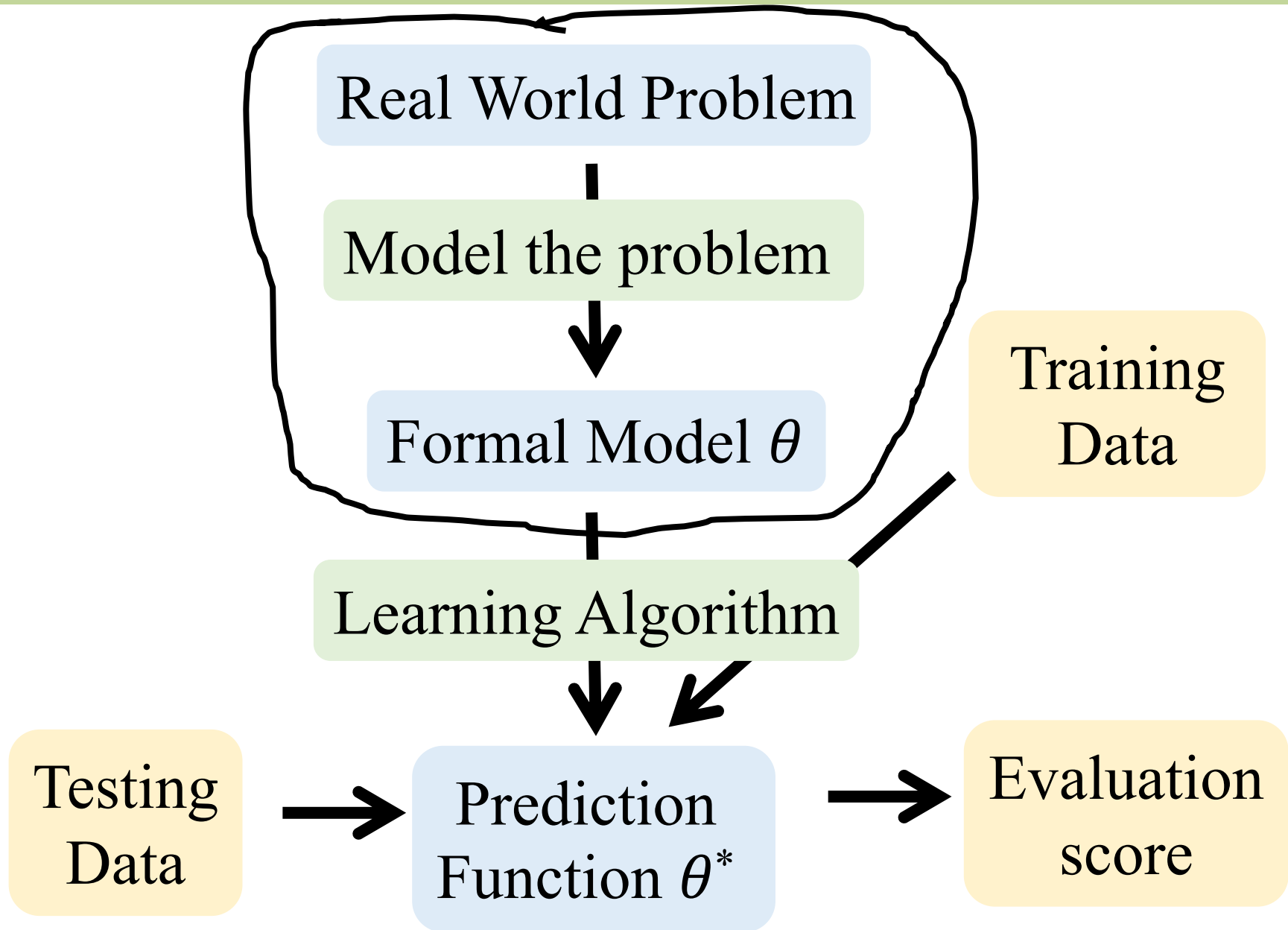
- In real world, don't know “true” parameters
  - But, **we do get to observe data**
    - E.g., number of times coin comes up heads, lifetimes of disk drives produced, number of visitors to web site per day, etc.
  - Need to estimate model parameters from data
  - “Estimator” is random variable estimating parameter
- Estimate of parameters allows:
  - Better understanding of process producing data
  - Future **predictions** based on model
  - Simulation of processes



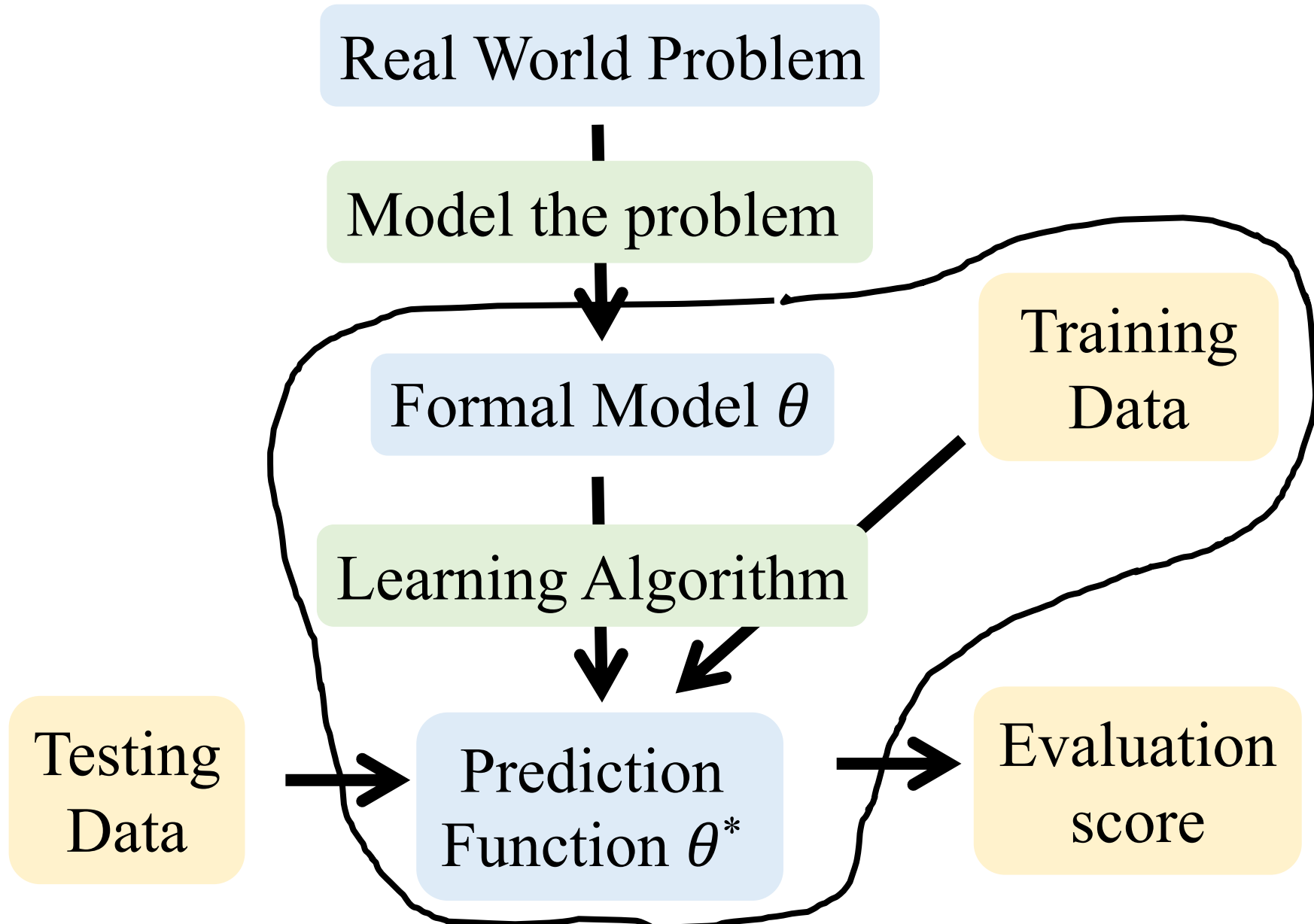
# Supervised Learning



# Modelling



# Training



# Testing

Real World Problem

Model the problem

Formal Model  $\theta$

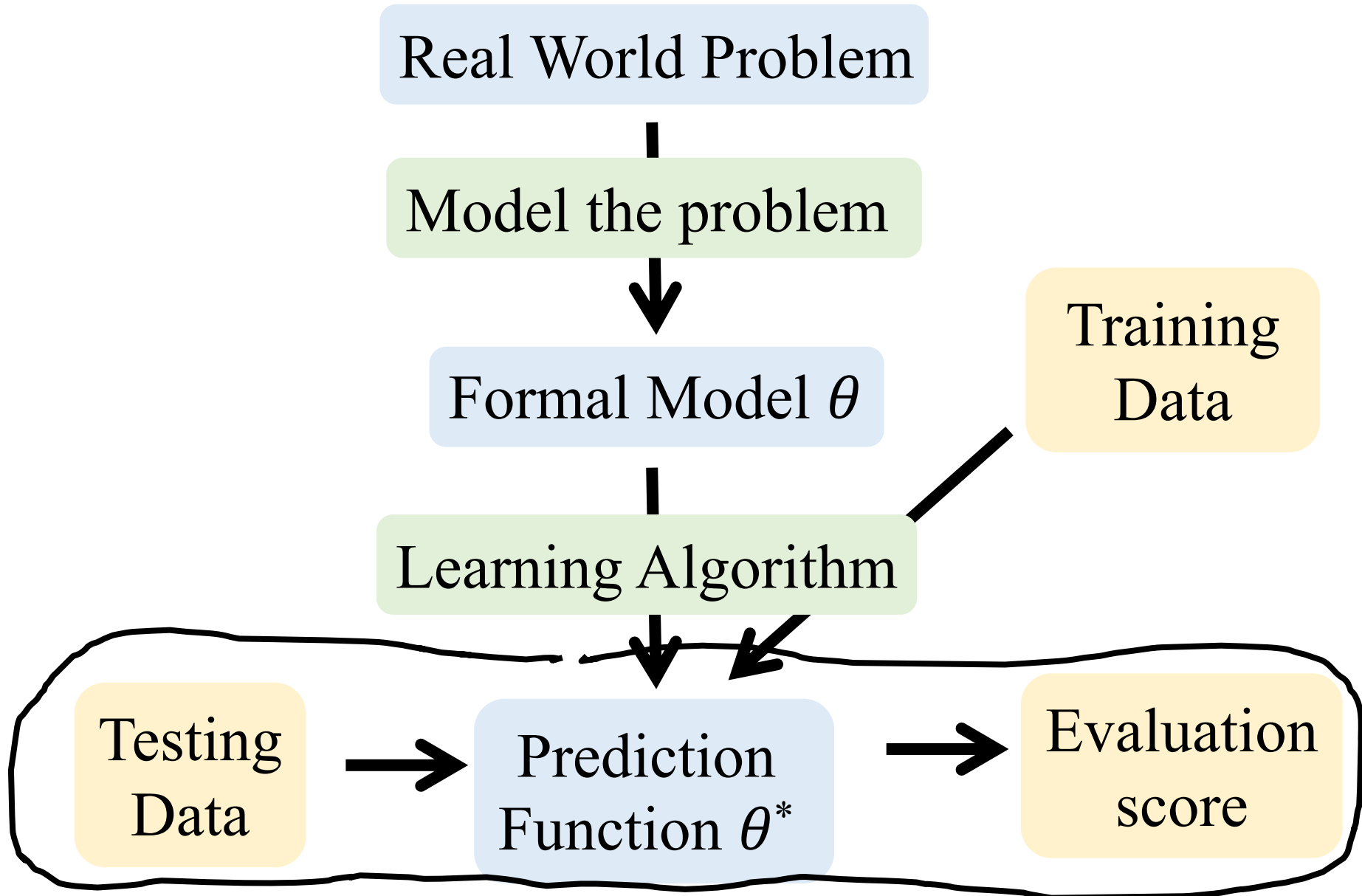
Learning Algorithm

Training  
Data

Testing  
Data

Prediction  
Function  $\theta^*$

Evaluation  
score



Basis for learning from data

# Our Path

Neural Networks

Linear  
Regression

Naive  
Bayes

Logistic  
Regression

Unbiased  
estimators

Maximizing  
likelihood

Bayesian  
estimation

# Parameter Estimation

Neural Networks

Linear  
Regression

Naïve  
Bayes

Logistic  
Regression

Unbiased  
estimators

Maximizing  
likelihood

Bayesian  
estimation



# Recall Sample Mean + Variance?

- Consider  $n$  I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i$  have distribution  $F$  with  $E[X_i] = \mu$  and  $\text{Var}(X_i) = \sigma^2$
  - We call sequence of  $X_i$  a **sample** from distribution  $F$

- Recall sample mean:  $\bar{X} = \sum_{i=1}^n \frac{X_i}{n}$  where  $E[\bar{X}] = \mu$

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right) \text{ as } n \rightarrow \infty$$

- Recall sample variance:

$$S^2 = \sum_{i=1}^n \frac{(X_i - \bar{X})^2}{n-1} = \text{undefined}$$

Estimate parameters for  
Bernoulli and Normal

Limited tool: how could we use that for fitting a “Mixture of Gaussians”?

Great idea in Machine Learning

# Likelihood of Data

- Consider  $n$  I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i$  is a sample from density function  $f(X_i | \theta)$ 
    - Note: now explicitly specify parameter  $\theta$  of distribution



Likelihood question:

How likely is the data given the samples?

$$\text{Likelihood}(\theta) = f(\text{Samples}|\theta)$$

[Demo](#)





# Likelihood of Data

- Consider  $n$  I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i$  is a sample from density function  $f(X_i | \theta)$ 
    - Note: now explicitly specify parameter  $\theta$  of distribution
  - We want to determine how “likely” the observed data  $(x_1, x_2, \dots, x_n)$  is based on density  $f(X_i | \theta)$

- Define the **Likelihood function**,  $L(\theta)$ :

$$L(\theta) = \prod_{i=1}^n f(X_i | \theta)$$

- This is just a product since  $X_i$  are I.I.D.
- Intuitively: what is probability of observed data using density function  $f(X_i | \theta)$ , for some choice of  $\theta$



# Maximum Likelihood Estimator

- The **Maximum Likelihood Estimator** (MLE) of  $\theta$ , is the value of  $\theta$  that maximizes  $L(\theta)$ 
  - More formally:  $\theta_{MLE} = \arg \max_{\theta} L(\theta)$



Likelihood (of data given parameters):

$$L(\theta) = \prod_{i=1}^n f(X_i | \theta)$$



Likelihood (of data given parameters):

$$L(\theta) = \prod_{i=1}^n f(X_i | \theta)$$

Either the  
PDF (continuous) or  
PMF (discrete)

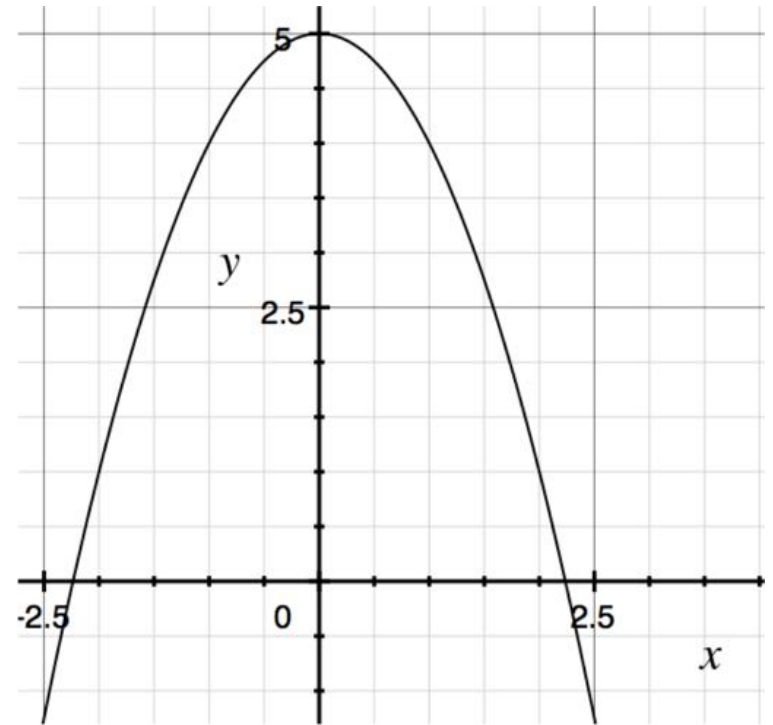


# Argmax

$$f(x) = -x^2 + 5$$

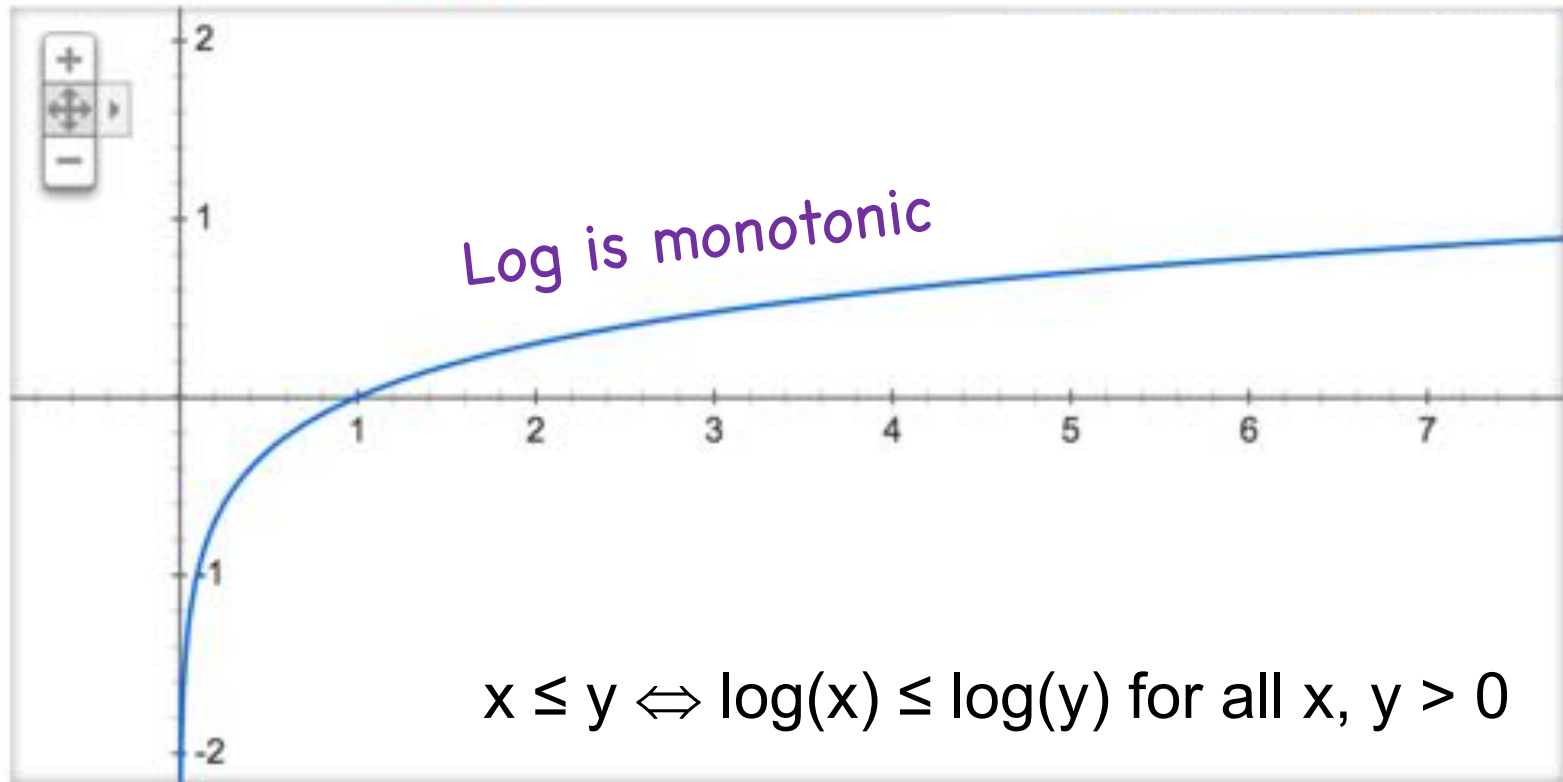
$$\max_x -x^2 + 5 = 5$$

$$\operatorname{argmax}_x -x^2 + 5 = 0$$



# Argmax of Log

Graph for  $\log(x)$



Claim:  $\operatorname{argmax}_x f(x) = \operatorname{argmax}_x \log f(x)$

# Argmax of Log



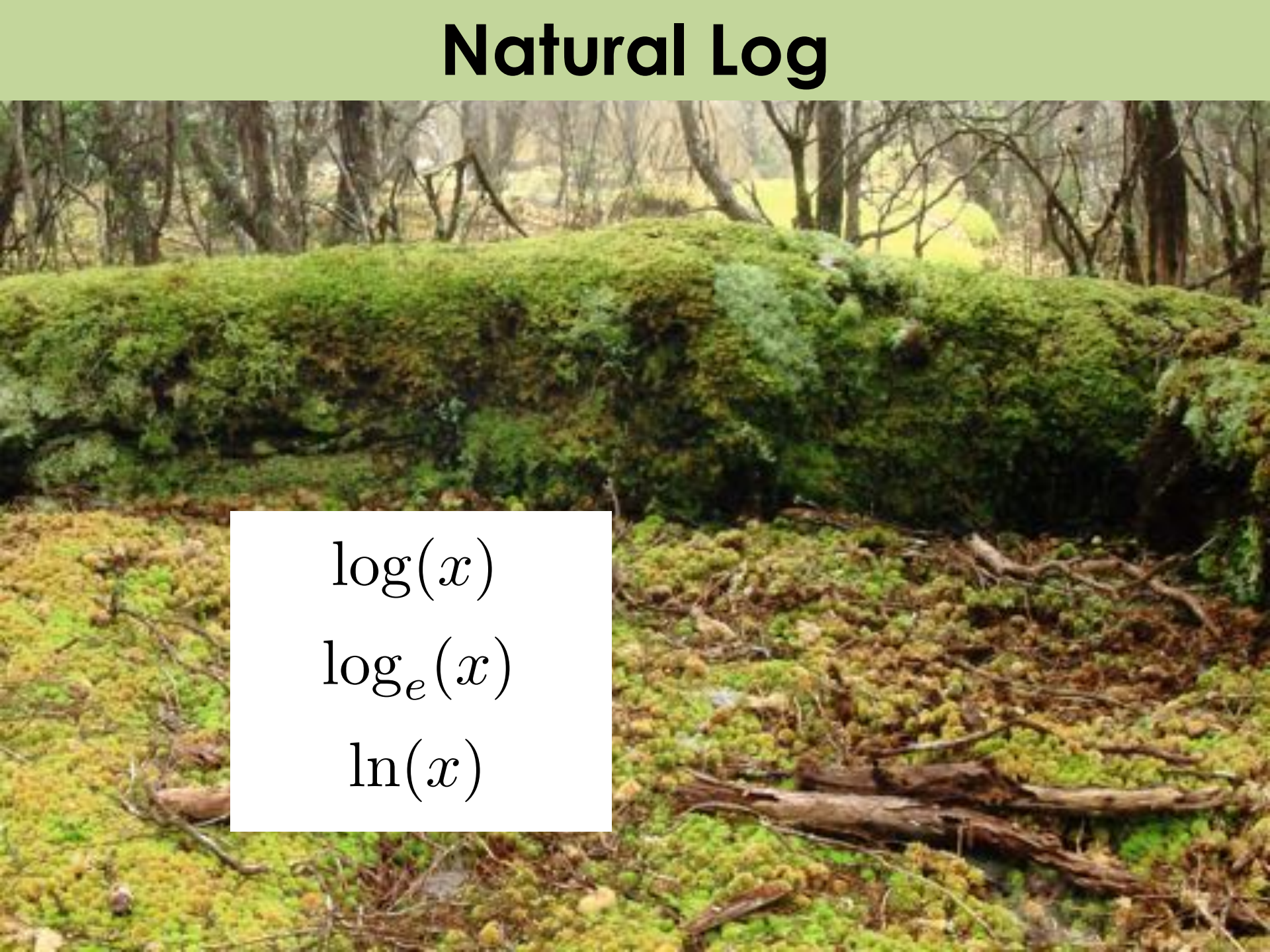
$$\operatorname{argmax}_x f(x) = \operatorname{argmax}_x \log f(x)$$

# Log I Love You

$$\log(ab) = \log(a) + \log(b)$$



# Natural Log



$\log(x)$   
 $\log_e(x)$   
 $\ln(x)$



# Maximum Likelihood Estimator

- The **Maximum Likelihood Estimator** (MLE) of  $\theta$ , is the value of  $\theta$  that maximizes  $L(\theta)$ 
  - More formally:  $\theta_{MLE} = \arg \max_{\theta} L(\theta)$
  - More convenient to use **log-likelihood function**,  $LL(\theta)$ :

$$LL(\theta) = \log L(\theta) = \log \prod_{i=1}^n f(X_i | \theta) = \sum_{i=1}^n \log f(X_i | \theta)$$

- $\theta$  that maximizes  $LL(\theta)$  also maximizes  $L(\theta)$ 
  - Formally:  $\arg \max_{\theta} LL(\theta) = \arg \max_{\theta} L(\theta)$
  - Similarly, for any positive constant  $c$  (not dependent on  $\theta$ ):

$$\arg \max_{\theta} (c \cdot LL(\theta)) = \arg \max_{\theta} LL(\theta) = \arg \max_{\theta} L(\theta)$$

Story so far: We can choose parameters by finding the argmax of the log likelihood of our data

# Maximum Likelihood



$$L(\theta) = \prod_{i=1}^n f(X_i | \theta)$$

$$LL(\theta) = \sum_{i=1}^n \log f(X_i | \theta)$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} LL(\theta)$$





But how do we compute  $\operatorname{argmax}$ ?

Option #1: Straight optimization

# Computing the MLE

- General approach for finding MLE of  $\theta$ 
  - Determine formula for  $LL(\theta)$
  - Differentiate  $LL(\theta)$  w.r.t. (each)  $\theta$ :  $\frac{\partial LL(\theta)}{\partial \theta}$
  - To maximize, set  $\frac{\partial LL(\theta)}{\partial \theta} = 0$
  - Solve resulting (simultaneous) equations to get  $\theta_{MLE}$ 
    - Make sure that derived  $\hat{\theta}_{MLE}$  is actually a maximum (and not a minimum or saddle point). E.g., check  $LL(\theta_{MLE} \pm \varepsilon) < LL(\theta_{MLE})$ 
      - This step often ignored in expository derivations
      - So, we'll ignore it here too (and won't require it in this class)

# Maximizing Likelihood with Bernoulli

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i \sim \text{Ber}(p)$
  - Probability mass function,  $f(X_i | p)$ :

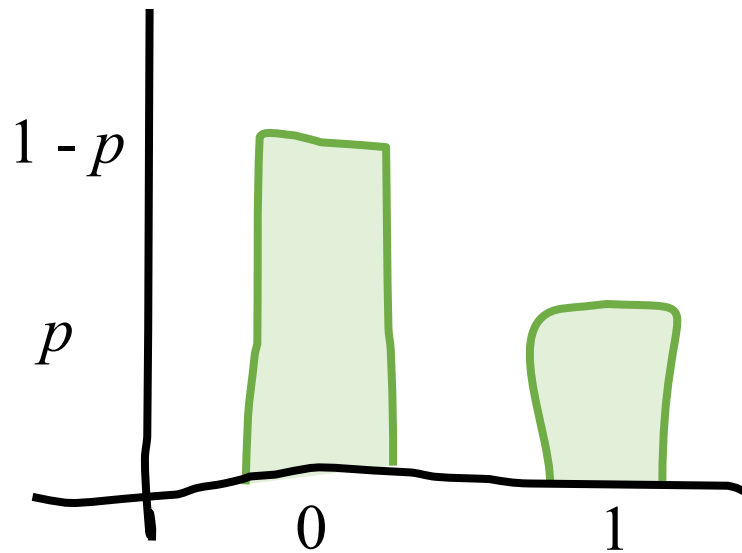




# Maximizing Likelihood with Bernoulli

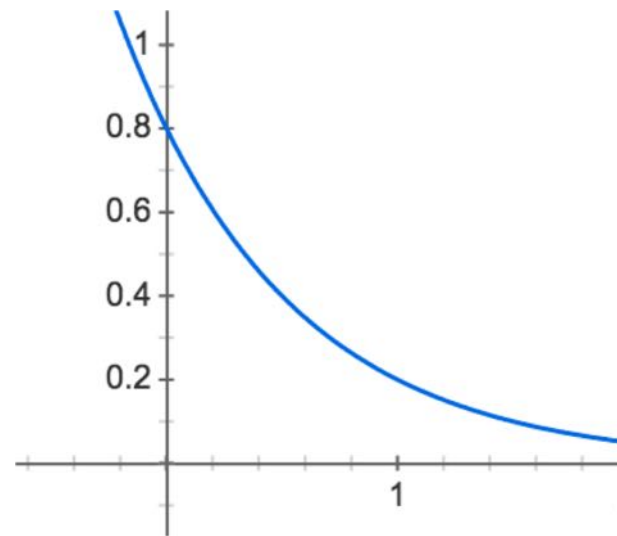
- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i \sim \text{Ber}(p)$
  - Probability mass function,  $f(X_i | p)$ :

PMF of Bernoulli



$$f(X_i | p) = p^{x_i} (1-p)^{1-x_i}$$

PMF of Bernoulli ( $p = 0.2$ )



$$f(x) = 0.2^x (1 - 0.2)^{1-x}$$

# Bernoulli PMF

$$X \sim \text{Ber}(p)$$



$$f(X = x|p) = p^x (1 - p)^{1-x}$$

# Maximizing Likelihood with Bernoulli

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i \sim \text{Ber}(p)$
  - Probability mass function,  $f(X_i | p)$ , can be written as:

$$f(X_i | p) = p^{x_i} (1-p)^{1-x_i} \quad \text{where } x_i = 0 \text{ or } 1$$

- Likelihood:  $L(\theta) = \prod_{i=1}^n p^{X_i} (1-p)^{1-X_i}$

- Log-likelihood:

$$\begin{aligned} LL(\theta) &= \sum_{i=1}^n \log(p^{X_i} (1-p)^{1-X_i}) = \sum_{i=1}^n [X_i (\log p) + (1-X_i) \log(1-p)] \\ &= Y (\log p) + (n-Y) \log(1-p) \quad \text{where } Y = \sum_{i=1}^n X_i \end{aligned}$$

- Differentiate w.r.t.  $p$ , and set to 0:

$$\frac{\partial LL(p)}{\partial p} = Y \frac{1}{p} + (n-Y) \frac{-1}{1-p} = 0 \quad \Rightarrow \quad p_{MLE} = \frac{Y}{n} = \frac{1}{n} \sum_{i=1}^n X_i$$

Isn't that the same as  
unbiased estimator?

Yes. For Bernoulli.



# Maximum Likelihood Algorithm

1. Decide on a model for the distribution of your samples. Define the PMF / PDF for your sample.

2. Write out the log likelihood function.

3. State that the optimal parameters are the argmax of the log likelihood function.

4. Use an optimization algorithm to calculate argmax





# Maximizing Likelihood with Poisson

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$

- $X_i \sim \text{Poi}(\lambda)$

- PMF:  $f(X_i | \lambda) = \frac{e^{-\lambda} \lambda^{X_i}}{X_i!}$       Likelihood:  $L(\theta) = \prod_{i=1}^n \frac{e^{-\lambda} \lambda^{X_i}}{X_i!}$

- Log-likelihood:

$$\begin{aligned} LL(\theta) &= \sum_{i=1}^n \log\left(\frac{e^{-\lambda} \lambda^{X_i}}{X_i!}\right) = \sum_{i=1}^n [-\lambda \log(e) + X_i \log(\lambda) - \log(X_i!)] \\ &= -n\lambda + \log(\lambda) \sum_{i=1}^n X_i - \sum_{i=1}^n \log(X_i!) \end{aligned}$$

- Differentiate w.r.t.  $\lambda$ , and set to 0:

$$\frac{\partial LL(\lambda)}{\partial \lambda} = -n + \frac{1}{\lambda} \sum_{i=1}^n X_i = 0 \quad \Rightarrow \quad \lambda_{MLE} = \frac{1}{n} \sum_{i=1}^n X_i$$

Its so general!

# Maximizing Likelihood with Normal

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$

- $X_i \sim N(\mu, \sigma^2)$

- PDF:  $f(X_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(X_i - \mu)^2 / (2\sigma^2)}$

- Log-likelihood:

$$LL(\theta) = \sum_{i=1}^n \log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-(X_i - \mu)^2 / (2\sigma^2)}\right) = \sum_{i=1}^n \left[ -\log(\sqrt{2\pi}\sigma) - (X_i - \mu)^2 / (2\sigma^2) \right]$$

- First, differentiate w.r.t.  $\mu$ , and set to 0:

$$\frac{\partial LL(\mu, \sigma^2)}{\partial \mu} = \sum_{i=1}^n 2(X_i - \mu) / (2\sigma^2) = \frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \mu) = 0$$

- Then, differentiate w.r.t.  $\sigma$ , and set to 0:

$$\frac{\partial LL(\mu, \sigma^2)}{\partial \sigma} = \sum_{i=1}^n -\frac{1}{\sigma} + 2(X_i - \mu)^2 / (2\sigma^3) = -\frac{n}{\sigma} + \sum_{i=1}^n (X_i - \mu)^2 / (\sigma^3) = 0$$

# Being Normal, Simultaneously

- Now have two equations, two unknowns:

$$\frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \mu) = 0 \quad -\frac{n}{\sigma} + \sum_{i=1}^n (X_i - \mu)^2 / (\sigma^3) = 0$$

- First, solve for  $\mu_{MLE}$ :

$$\frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \mu) = 0 \Rightarrow \sum_{i=1}^n X_i = n\mu \Rightarrow \mu_{MLE} = \frac{1}{n} \sum_{i=1}^n X_i$$

- Then, solve for  $\sigma^2_{MLE}$ :

$$-\frac{n}{\sigma} + \sum_{i=1}^n (X_i - \mu)^2 / (\sigma^3) = 0 \Rightarrow n\sigma^2 = \sum_{i=1}^n (X_i - \mu)^2$$

$$\sigma^2_{MLE} = \frac{1}{n} \sum_{i=1}^n (X_i - \mu_{MLE})^2$$

- Note:  $\mu_{MLE}$  unbiased, but  $\sigma^2_{MLE}$  biased

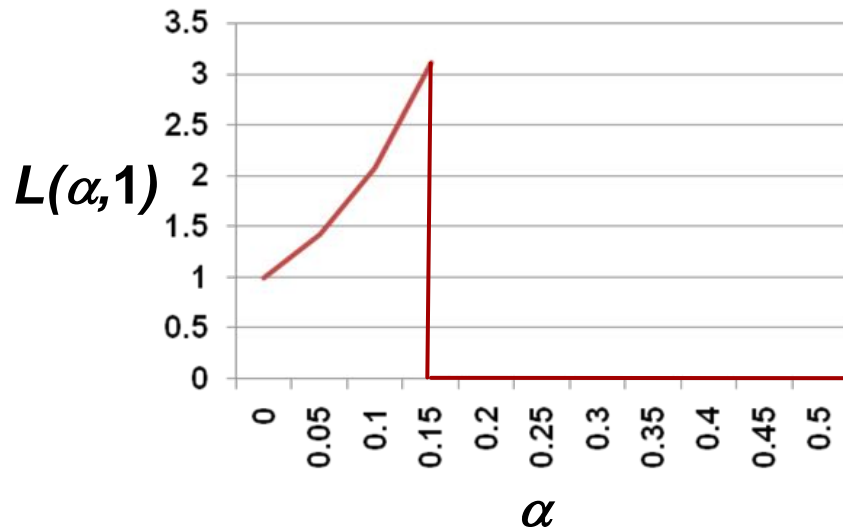
# Maximizing Likelihood with Uniform

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i \sim \text{Uni}(\alpha, \beta)$
  - PDF:  $f(X_i | \alpha, \beta) = \begin{cases} \frac{1}{\beta - \alpha} & \alpha \leq x_i \leq \beta \\ 0 & \text{otherwise} \end{cases}$
  - Likelihood:  $L(\theta) = \begin{cases} \left(\frac{1}{\beta - \alpha}\right)^n & \alpha \leq x_1, x_2, \dots, x_n \leq \beta \\ 0 & \text{otherwise} \end{cases}$ 
    - Constraint  $\alpha \leq x_1, x_2, \dots, x_n \leq \beta$  makes differentiation tricky
    - Intuition: want interval size  $(\beta - \alpha)$  to be as small as possible to maximize likelihood function for each data point
    - But need to make sure all observed data contained in interval
      - If all observed data not in interval, then  $L(\theta) = 0$
  - Solution:  $\alpha_{MLE} = \min(x_1, \dots, x_n)$      $\beta_{MLE} = \max(x_1, \dots, x_n)$

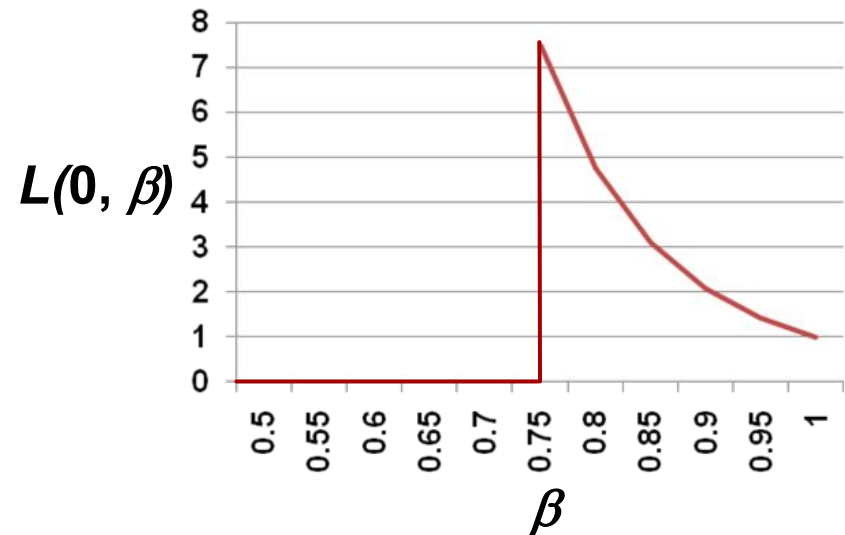
# Understanding MLE with Uniform

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$ 
  - $X_i \sim \text{Uni}(0, 1)$
  - Observe data:
    - 0.15, 0.20, 0.30, 0.40, 0.65, 0.70, 0.75

Likelihood:  $L(\alpha, 1)$



Likelihood:  $L(0, \beta)$



# Small Samples = Problems

- How do small samples affect MLE?

- In many cases,  $\mu_{MLE} = \frac{1}{n} \sum_{i=1}^n X_i = \text{sample mean}$ 
  - Unbiased. Not too shabby...
- As seen with Normal,  $\sigma_{MLE}^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \mu_{MLE})^2$ 
  - Biased. Underestimates for small  $n$  (e.g., 0 for  $n = 1$ )
- As seen with Uniform,  $\alpha_{MLE} \geq \alpha$  and  $\beta_{MLE} \leq \beta$ 
  - Biased. Problematic for small  $n$  (e.g.,  $\alpha = \beta$  when  $n = 1$ )
- Small sample phenomena intuitively make sense:
  - Maximum likelihood  $\Rightarrow$  best explain data we've seen
  - Does not attempt to generalize to unseen data

# Properties of MLE

- Maximum Likelihood Estimators are generally:
  - **Consistent:**  $\lim_{n \rightarrow \infty} P(|\hat{\theta} - \theta| < \varepsilon) = 1$  for  $\varepsilon > 0$
  - Potentially biased (though asymptotically less so)
  - **Asymptotically optimal**
    - Has smallest variance of “good” estimators for large samples
  - **Often used in practice** where sample size is large relative to parameter space
    - But be careful, there are some very large parameter spaces



