24: Naïve Bayes

Lisa Yan

November 15, 2019

Estimating our parameter directly

Maximum Likelihood Estimator (MLE)

What is the parameter θ that **maximizes the likelihood** of our observed data $(x_1, x_2, \dots, x_n)$?

$$\theta_{MLE} = \arg\max_{\theta} f(X_1, X_2, \dots, X_n | \theta) = \arg\max_{\theta} \underbrace{\sum_{i=1}^{n} \log f(X_i | \theta)}_{\text{log likelihood}}$$

Maximum a Posteriori (MAP) Estimator

Given our observed data $(x_1, x_2, \dots, x_n)$, what is the **most likely parameter** θ?

$$\theta_{MAP} = \arg\max_{\theta} \underbrace{f(\theta | X_1, X_2, \dots, X_n)}_{\text{posterior distribution of } \theta}$$

$$= \arg\max_{\theta} \underbrace{\log g(\theta)}_{\text{Log-prior of } \theta} + \underbrace{\sum_{i=1}^{n} \log f(X_i | \theta)}_{\text{log likelihood}}$$

Maximum A Posterior (MAP) Estimator

Review

The MAP estimator has 2 interpretations:

$$\theta_{MAP} = \arg\max_{\theta} f(\theta | X_1, X_2, \dots, X_n)$$

The mode of the posterior distribution of θ

$$= \arg\max_{\theta} \left(\log g(\theta) + \sum_{i=1}^{n} \log f(X_i | \theta) \right)$$

The θ that maximizes log prior + log-likelihood

In both cases, you must specify your prior, $g(\theta)$.

Key to MAP estimator: You should pick a prior $g(\theta)$ that makes computing the mode of the posterior distribution is **easy**.

(in this class) 👉 Use a conjugate distribution.

How does MAP work?

0. Observe data — n heads, m tails

1. Choose model — Bernoulli(p)

2. Choose prior of θ — (some $g(\theta)$) — ⭐ (choose conjugate) Beta$(a+1, b+1)$

3. Compute posterior of θ given data

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

Beta$(a+n+1, b+m+1)$

4. $\theta_{MAP} = \arg\max_{\theta} f(\theta|X_1, X_2, \dots, X_n)$

- Differentiate
- Solve

Mode of Beta:

$$\frac{a+n}{a+n+b+m}$$

This is nice!

How does MAP work?

0. Observe data — n heads, m tails

1. Choose model — Bernoulli(p)

2. Choose prior of θ — (some $g(\theta)$) — (choose conjugate) Beta$(a+1, b+1)$

3. Compute posterior of θ given data — $\log g(\theta) + \sum_{i=1}^{n} \log f(X_i|\theta)$ — Beta$(a+n+1, b+m+1)$

4. $\theta_{MAP} = \arg\max_{\theta} f(\theta|X_1, X_2, \ldots, X_n)$
   - Differentiate
   - Solve

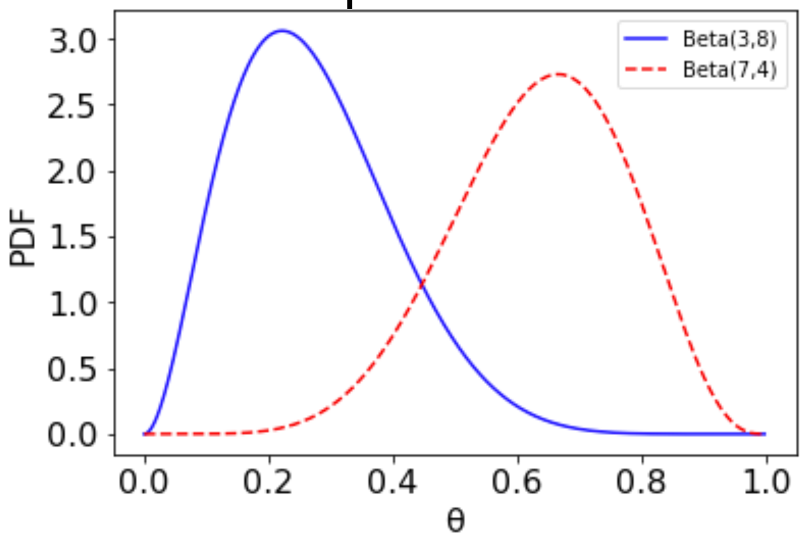
Mode of Beta:

$$\frac{a+n}{a+n+b+m}$$

Where'd you get them priors?

- Let θ be the probability a coin turns up heads.
- Model θ with 2 different priors:
  - Prior 1: Beta(3,8): 2 imaginary heads, 7 imaginary tails — mode: $\frac{2}{9}$
  - Prior 2: Beta(7,4): 6 imaginary heads, 3 imaginary tails — mode: $\frac{6}{9}$

prior

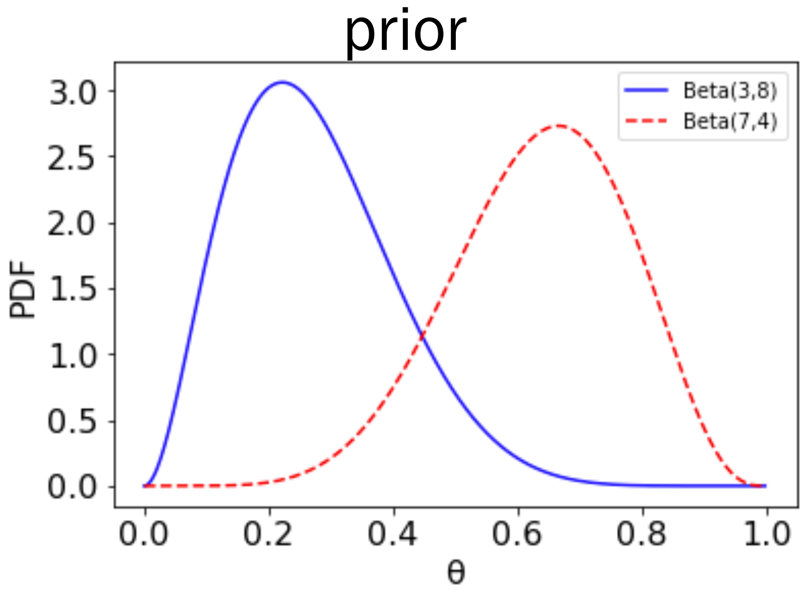


Now flip 100 coins and get 58 heads and 42 tails.

1. What are the two posterior distributions?
2. What are the modes of the two posterior distributions?

Where'd you get them priors?

- Let θ be the probability a coin turns up heads.
- Model θ with 2 different priors:
  - Prior 1: Beta(3,8): 2 imaginary heads, 7 imaginary tails mode: $\frac{2}{9}$
  - Prior 2: Beta(7,4): 6 imaginary heads, 3 imaginary tails mode: $\frac{6}{9}$



Now flip 100 coins and get 58 heads and 42 tails.

1. What are the two posterior distributions?
2. What are the modes of the two posterior distributions?

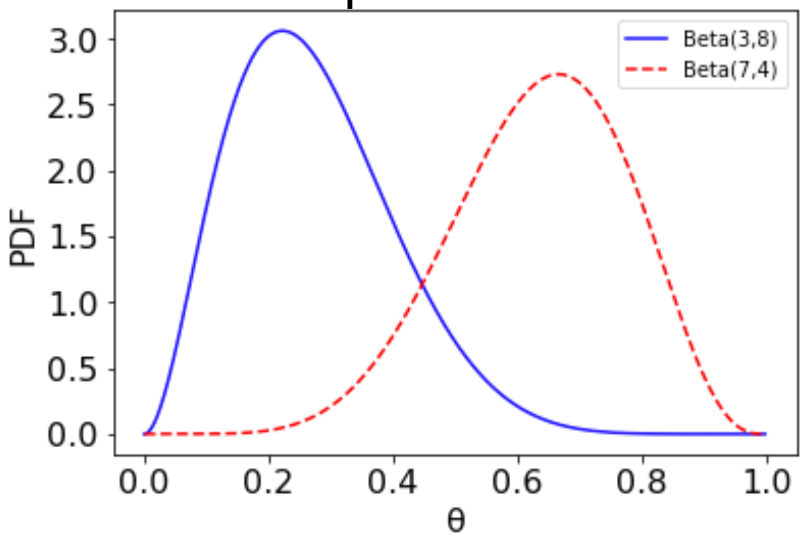
Posterior 1: Beta(61,50) mode: $\frac{60}{109}$

Posterior 2: Beta(65,46) mode: $\frac{64}{109}$

Where'd you get them priors?

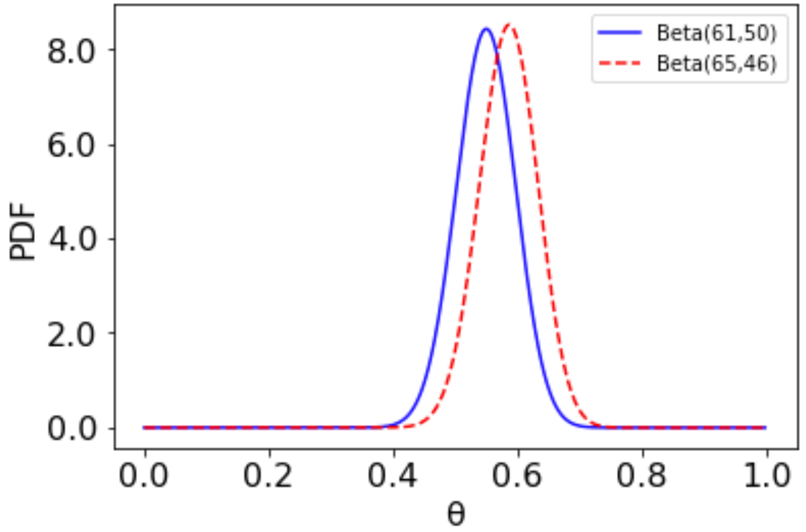
- Let θ be the probability a coin turns up heads.
- Model θ with 2 different priors:
  - Prior 1: Beta(3,8): 2 imaginary heads, 7 imaginary tails mode: $\frac{2}{9}$
  - Prior 2: Beta(7,4): 6 imaginary heads, 3 imaginary tails mode: $\frac{6}{9}$

prior



Now flip 100 coins and get 58 heads and 42 tails.

posterior



Posterior 1: Beta(61,50) mode: $\frac{60}{109}$

Posterior 2: Beta(65,46) mode: $\frac{64}{109}$

👉 As long as we collect enough data, posteriors will converge to the true value.

Today's plan

Maximum A Posteriori

- Picking a conjugate distribution as your prior
- Laplace smoothing

Naïve Bayes

Conjugate distributions

MAP estimator:

$$\theta_{MAP} = \arg\max_{\theta} f(\theta | X_1, X_2, \dots, X_n)$$

The mode of the posterior distribution of θ

Distribution parameter	Prior distribution for parameter
Bernoulli p	Beta
Binomial p	Beta
Multinomial p_i	Dirichlet
Poisson λ	Gamma
Exponential λ	Gamma
Normal μ	Normal
Normal σ^2	Inverse Gamma

Don't need to know Inverse Gamma…
but it will know you ☺

Multinomial is Multiple times the fun

Dirichlet$(a_1, a_2, \dots, a_m)$ is the conjugate for Multinomial.

- Generalizes Beta in the same way Multinomial generalizes Bernoulli/Binomial:

$$f(x_1, x_2, \dots, x_m) = \frac{1}{B(a_1, a_2, \dots, a_m)} \prod_{i=1}^{m} x_i^{a_i - 1}$$

Prior Dirichlet$(a_1 + 1, a_2 + 1, \dots, a_m + 1)$
Saw $\sum_{i=1}^{m} a_i$ imaginary trials, a_i of outcome i

Experiment Observe $n_1 + n_2 + \cdots + n_m$ new trials, with n_i of outcome i

Posterior Dirichlet$(a_1 + n_1 + 1, a_2 + n_2 + 1, \dots, a_m + n_m + 1)$

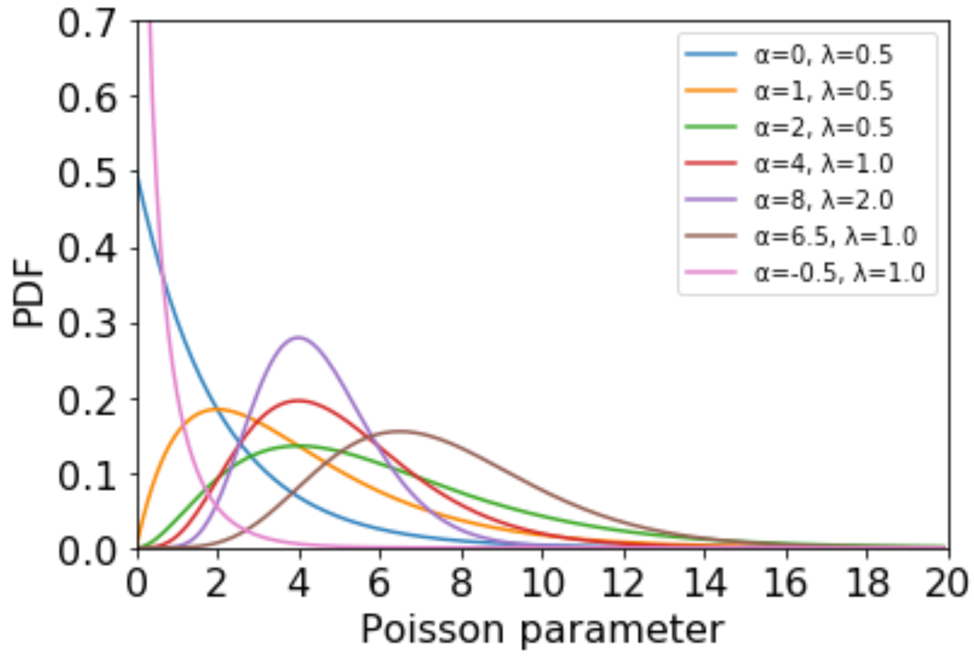
MAP:

$$p_i = \frac{a_i + n_i}{\sum_{i=1}^{m} a_i + \sum_{i=1}^{m} n_i}$$

Good times with Gamma

Gamma(α, λ) is the conjugate for Poisson.

- Also conjugate for Exponential, but we won't delve into that
- Mode of gamma: α/λ



Prior $\theta \sim$ Gamma(α, λ)
Saw α total imaginary events during λ prior time periods

Experiment Observe n events during next k time periods

Posterior $(\theta | n$ events in k periods$)$
$\sim$Gamma$(\alpha + n, \lambda + k)$

$$\theta_{MAP} = \frac{\alpha + n}{\lambda + k}$$

MAP for Poisson

$\text{Gamma}(\alpha, \lambda)$
is conjugate for Poisson

Mode: α/λ

Let λ be the average # of successes in a time period.

1. What does it mean to have a prior of $\theta \sim \text{Gamma}(10,5)$?

Observe 10 imaginary events
in 5 time periods,
i.e., observe at Poisson rate = 2

Now perform the experiment and see 11 events in next 2 time periods.

2. Given your prior, what is the posterior distribution?

3. What is θ_{MAP}?

MAP for Poisson

$\text{Gamma}(\alpha, \lambda)$ is conjugate for Poisson

Mode: α/λ

Let λ be the average # of successes in a time period.

1. What does it mean to have a prior of $\theta \sim \text{Gamma}(10,5)$?

   Observe 10 imaginary events in 5 time periods, i.e., observe at Poisson rate = 2

Now perform the experiment and see 11 events in next 2 time periods.

2. Given your prior, what is the posterior distribution?

   $(\theta | n \text{ events in } k \text{ periods}) \sim \text{Gamma}(21, 7)$

3. What is θ_{MAP}?

   $\theta_{MAP} = 3$, the updated Poisson rate

Today's plan

Maximum A Posteriori

- Picking a conjugate distribution as your prior
- Laplace smoothing

Machine Learning

- Inefficient classification: Brute force Bayes
- Naïve Bayes

Laplace smoothing

MAP with Laplace smoothing: a prior which represents **one** imagined observation of each outcome.

Consider our previous 6-sided die.

- Roll the dice $n = 12$ times.
- Observe: 3 ones, 2 twos, 0 threes, 3 fours, 1 fives, 3 sixes

Recall θ_{MLE}:

$p_1 = 3/12, p_2 = 2/12, p_3 = 0/12,$
$p_4 = 3/12, p_5 = 1/12, p_6 = 3/12$

θ_{MAP} with Laplace smoothing:

- Assume Dirichlet prior where each outcome seen $k = 1$ times.
- **Laplace estimate**:

$$p_i = \frac{X_i + 1}{n + m}$$

$p_1 = 4/18, p_2 = 3/18, p_3 = 1/18,$
$p_4 = 4/18, p_5 = 2/18, p_6 = 4/18$

👉 Laplace smoothing avoids the case where you estimate a parameter of 0.

Break for Friday/ announcements

Andy Warhol, *Campbell's Soup Cans* (1962)

Announcements

Problem Set 6

Released:	this afternoon
Due:	Wednesday 12/4 (after break)
Covers:	Up to next Wed. 11/20

Late day reminder: No late days permitted past last day of the quarter, 12/6 (Friday)

CS109 Contest

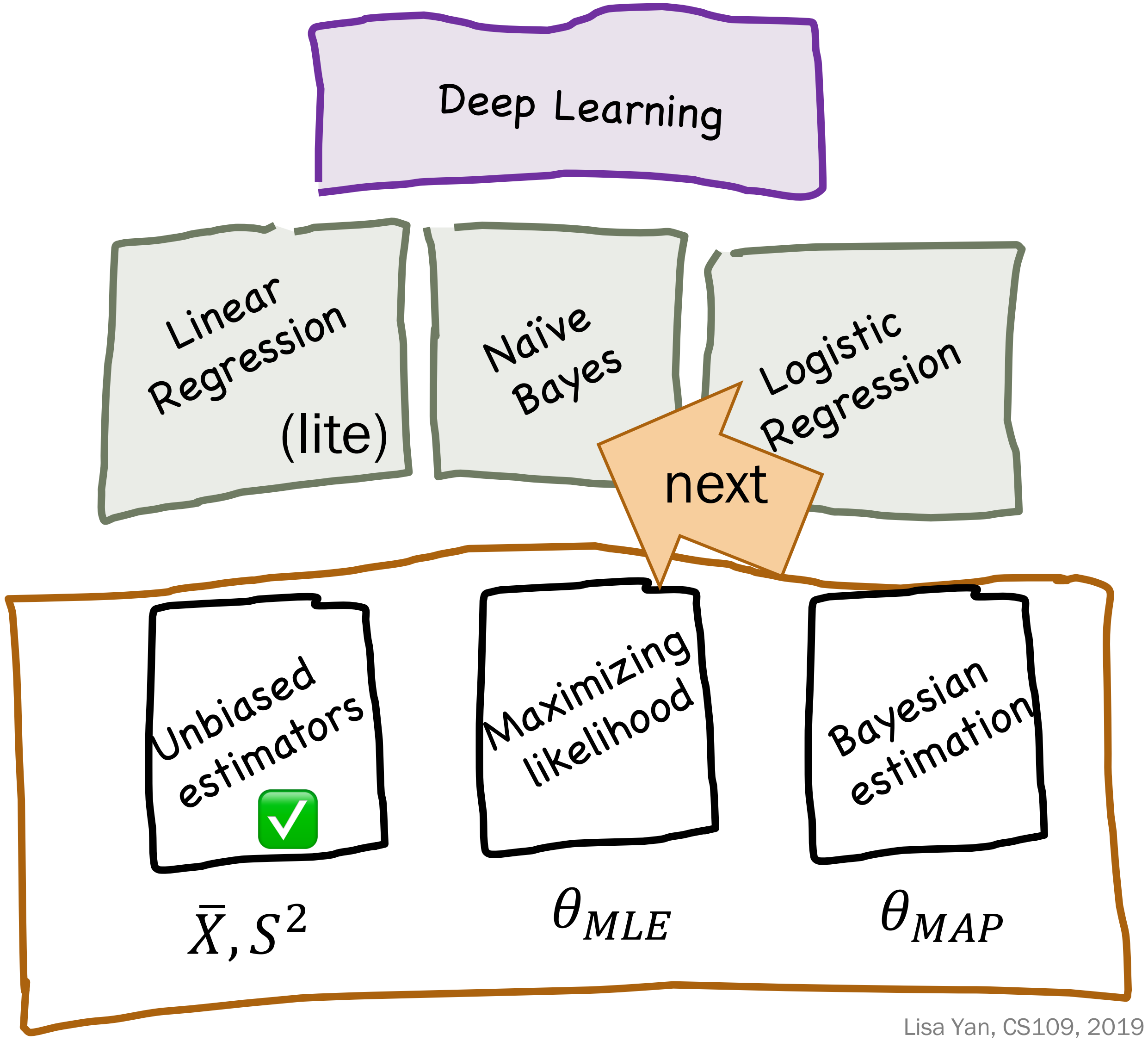
Due:	Monday 12/2 11:59pm
Note:	All serious submissions will get some extra credit

Today's plan

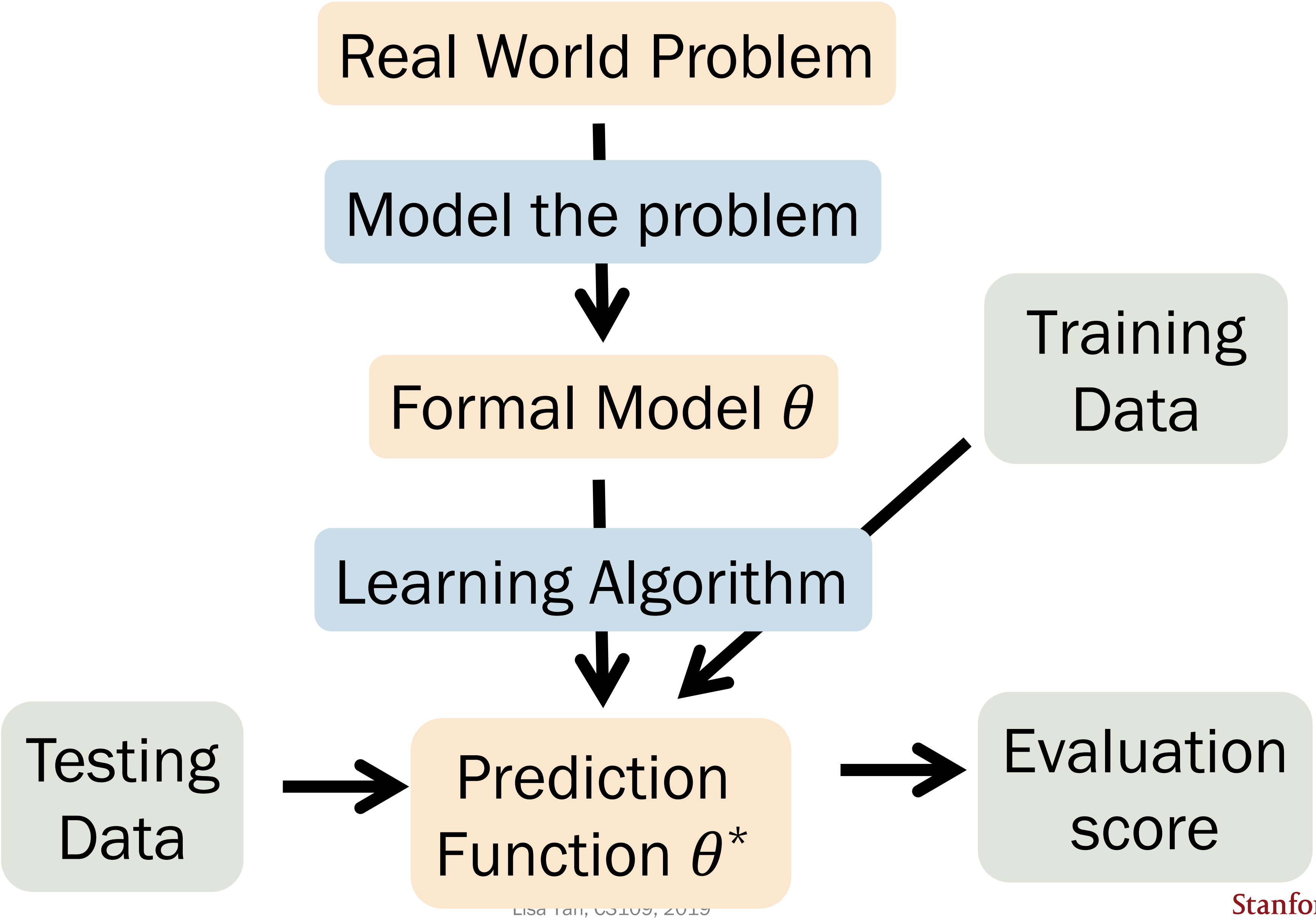
Maximum A Posteriori
- Picking a conjugate distribution as your prior
- Laplace smoothing

Machine Learning
- Inefficient classification: Brute force Bayes
- Naïve Bayes

Our path

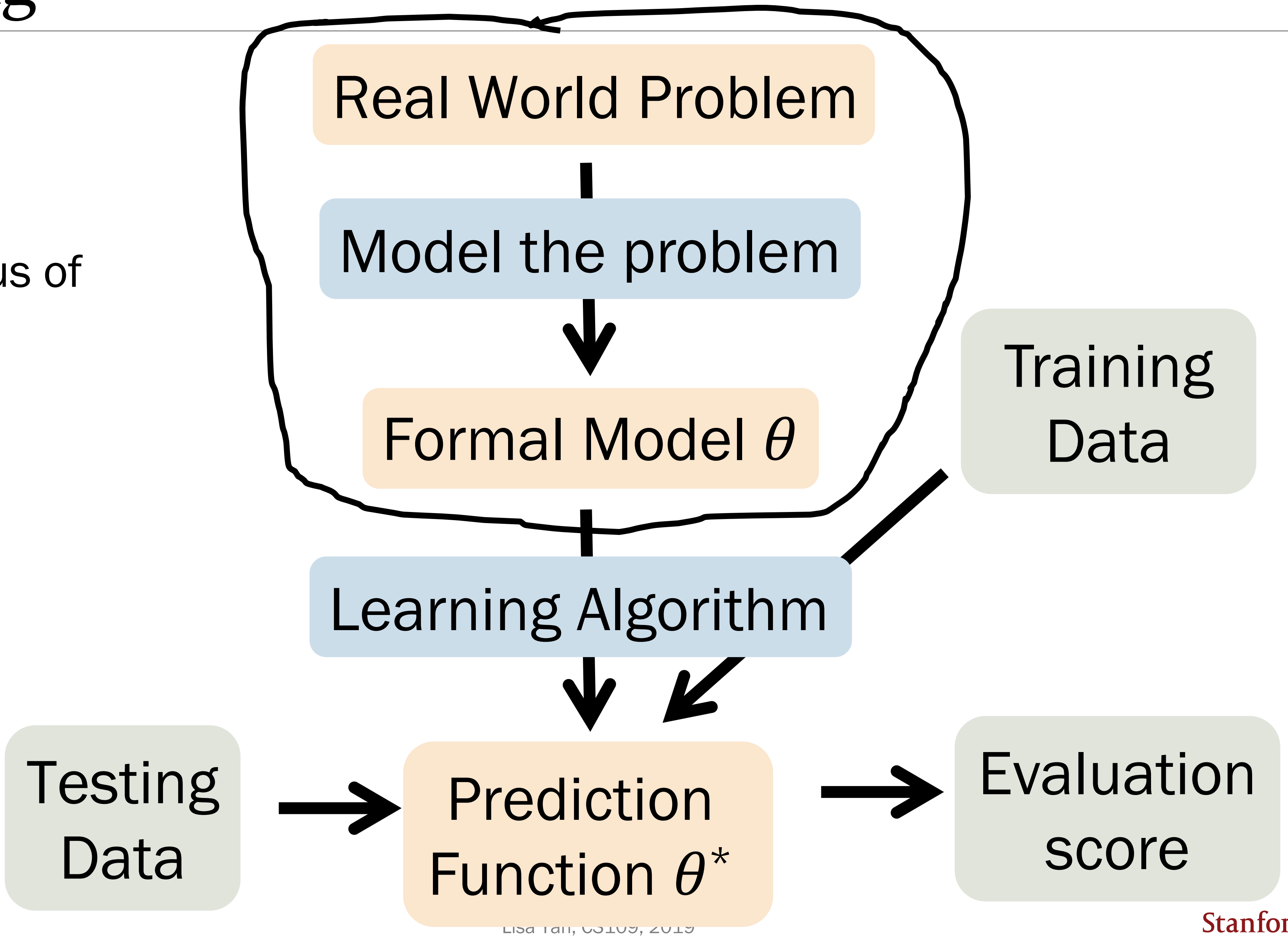

Deep Learning
Linear Regression (lite)
Naïve Bayes
Logistic Regression
next
Unbiased estimators
Maximizing likelihood
Bayesian estimation
$\bar{X}, S^2$
θ_{MLE}
θ_{MAP}

Supervised Learning

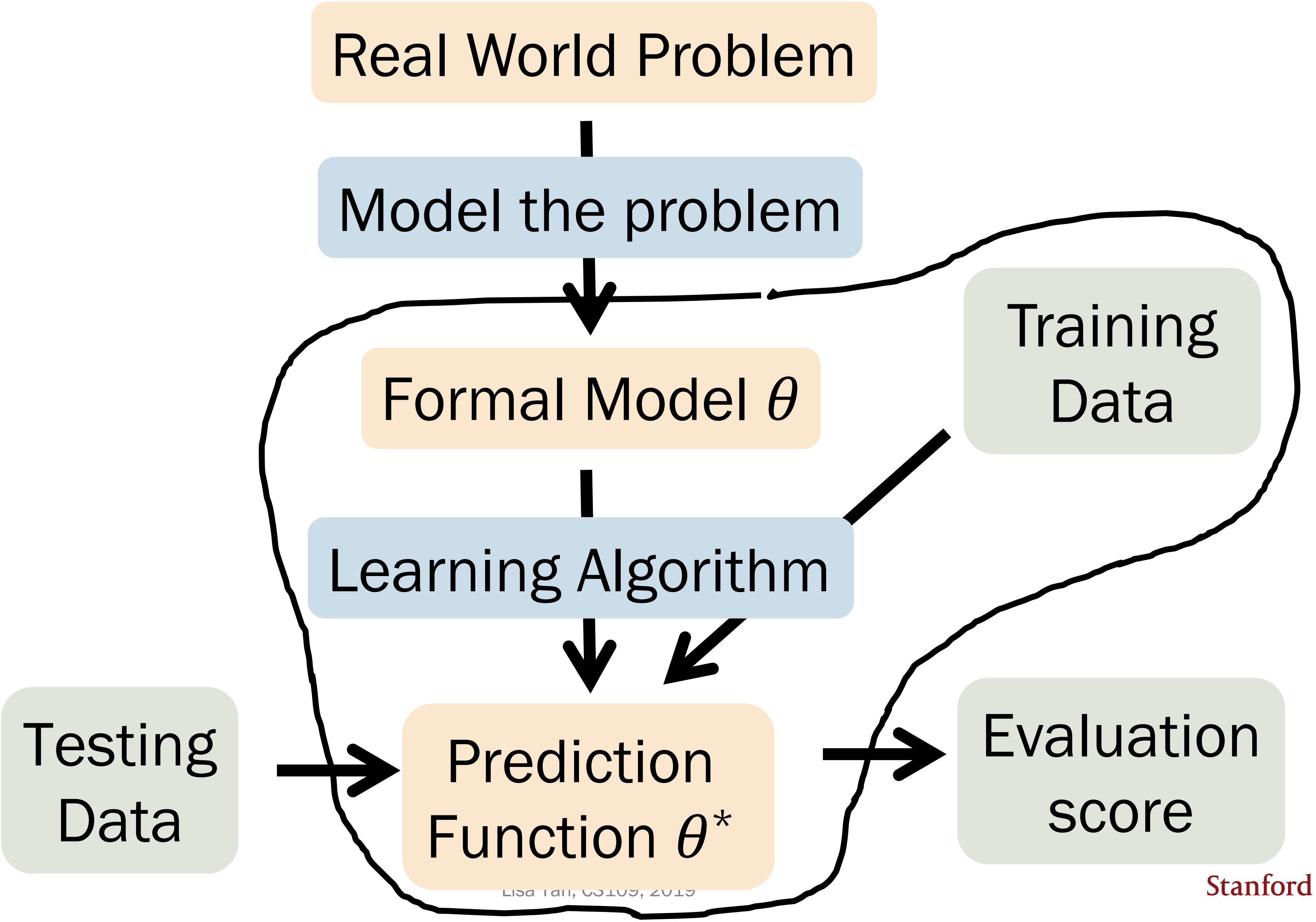


Modeling

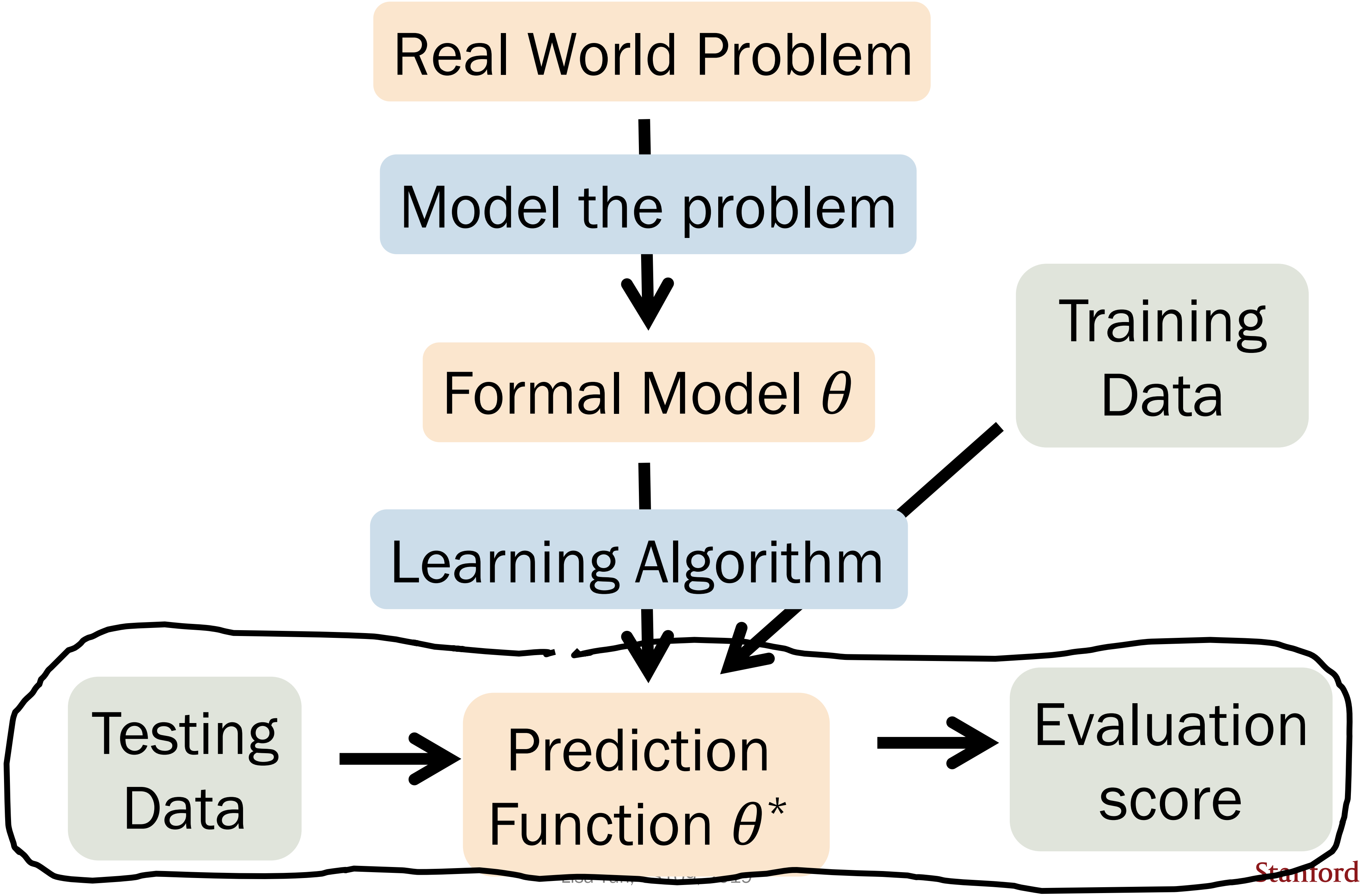
(not the focus of this class)



Training


Real World Problem
Model the problem
Formal Model θ
Training Data
Learning Algorithm
Testing Data
Prediction Function θ^*
Evaluation score

Testing


Real World Problem
Model the problem
Formal Model θ
Training Data
Learning Algorithm
Testing Data
Prediction Function θ^*
Evaluation score

Machine Learning (formally)

Many different forms of "Machine Learning"

- We focus on the problem of **prediction** based on observations.

Goal Based on observed $\boldsymbol{X}$, predict unseen Y

- **Features** Vector $\boldsymbol{X}$ of m observed variables $\boldsymbol{X} = (X_1, X_2, \dots, X_m)$
- **Output** Variable Y (also called **class label**)

Model $\hat{Y} = g(\boldsymbol{X})$, a function of observations $\boldsymbol{X}$

- **Classification** prediction when Y is discrete
- **Regression** prediction when Y is continuous

Training data

$$\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$$

n datapoints, generated i.i.d.

Each datapoint i is $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$:

- m features: $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}\right)$
- A single output $y^{(i)}$
- Independent of all other datapoints

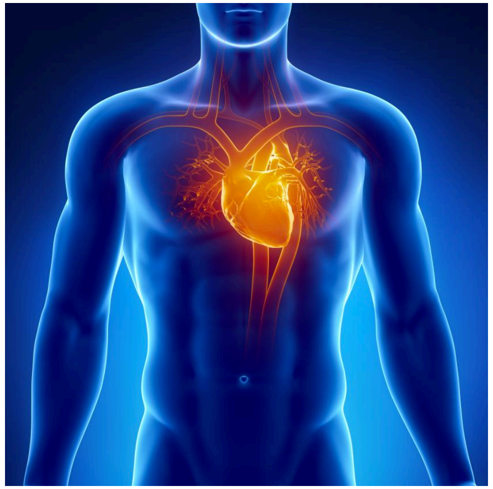
Training Goal:

Use these n datapoints to learn a model $\hat{Y} = g(\boldsymbol{X})$ that predicts Y

Errata: Switched i and j. Updated 11/16 to be consistent with lecture notes

Example datasets

Heart



Ancestry

23andMe

NETFLIX

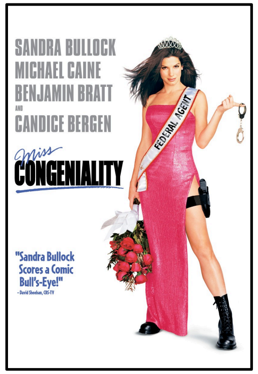
Netflix

Classification terminology check

A. $\boldsymbol{x}^{(i)}$
B. $y^{(i)}$
C. $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$
D. $x_j^{(i)}$

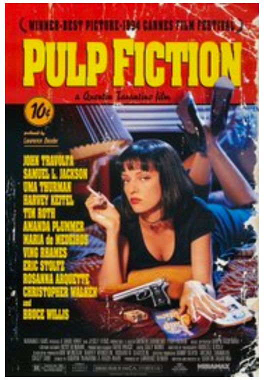
Training data: $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$

Movie 1



Movie 2

...



Movie m

Output

1: like movie
0: dislike movie

		Movie 1	Movie 2	...	Movie m		Output
User 1	1.	1	0	...	1	2.	1
User 2	3.	1	1	...	0		0
...				⋮			⋮
User n		0	4. 0	...	1		1

Errata: Switched i and j. Updated 11/16 to be consistent with lecture notes

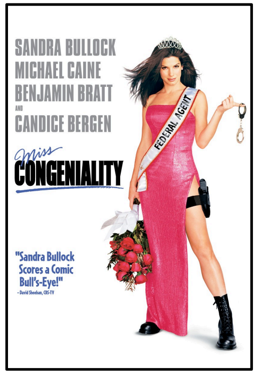
Classification terminology check

A. $\boldsymbol{x}^{(i)}$
B. $y^{(i)}$
C. $(\boldsymbol{x}^{(i)}, y^{(i)})$
D. $x_j^{(i)}$

Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$

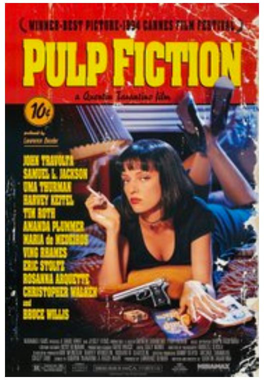
i: i-th user
j: movie j

Movie 1



Movie 2

…



Movie m

Output

1: like movie
0: dislike movie

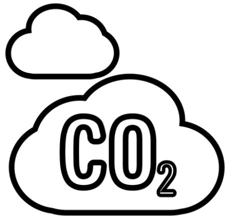
		Movie 1	Movie 2	…	Movie m		Output
User 1	1.	1	0	…	1	2.	1
User 2	3.	1	1	…	0		0
…				⋮			⋮
User n		0	4. 0	…	1		1

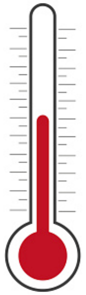
1. $\boldsymbol{x}^{(i)}$
2. $y^{(i)}$
3. $(\boldsymbol{x}^{(i)}, y^{(i)})$
4. $x_j^{(i)} = x_2^{(n)}$

Errata: Switched i and j. Updated 11/16 to be consistent with lecture notes

Regression: Predicting real numbers

Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$





Global Land-Ocean temperature

	CO2 levels	Sea level	…	Feature m	Output
Year 1	338.8	0	…	1	0.26
Year 2	340.0	1	…	0	0.32
…			⋮		⋮
Year n	340.76	0	…	1	0.14

Classification: Harry Potter Sorting Hat

$$\hat{Y} = 1$$

$$\boldsymbol{X} = (1, 1, 1, 0, 0, \dots, 1)$$

Today's plan

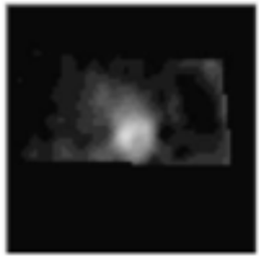
Maximum A Posteriori

- Picking a conjugate distribution as your prior
- Laplace smoothing

Machine Learning

- Inefficient classification: Brute force Bayes
- Naïve Bayes

Classification: Having a healthy heart

		
	Feature 1	Output
Patient 1	1	0
Patient 2	1	1
	⋮	⋮
Patient n	0	1

Feature 1: Region of Interest (ROI) is healthy (1) or unhealthy (0)

How can we predict the class label

heart is healthy (1) or unhealthy (0)?

One possible solution: Use Bayes.

Brute force Bayes

Classification (for one patient):

Choose the class label that is most likely given the data.

$$\hat{Y} = \arg\max_{y=\{0,1\}} \hat{P}(Y \mid \boldsymbol{X})$$

- $\hat{P}(Y = 1 \mid \boldsymbol{x})$: estimated probability a heart is healthy given $\boldsymbol{x}$
- $\boldsymbol{x}$: whether region of interest (ROI) is healthy (1) or unhealthy (0)

$$= \arg\max_{y=\{0,1\}} \frac{\hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)}{\hat{P}(\boldsymbol{X})}$$

(Bayes' Theorem)

$$= \arg\max_{y=\{0,1\}} \hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)$$

($1/\hat{P}(\boldsymbol{X})$ is a positive constant w.r.t Y)

Parameters for Brute Force Bayes

$$\hat{Y} = \underset{y=\{0,1\}}{\arg\max}\, \hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)$$

Parameters:

- $\hat{P}(\boldsymbol{X}|Y)$ for all $\boldsymbol{X}$ and Y
- $\hat{P}(Y)$ for all Y

Conditional probability tables $\hat{P}(\boldsymbol{X}|Y)$

	$\hat{P}(\boldsymbol{X}\|Y=0)$
$X_1 = 0$	θ_1
$X_1 = 1$	θ_2

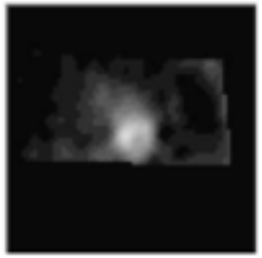
	$\hat{P}(\boldsymbol{X}\|Y=1)$
$X_1 = 0$	θ_3
$X_1 = 1$	θ_4

Marginal probability table $\hat{P}(Y)$

	$\hat{P}(Y)$
$Y = 0$	θ_5
$Y = 1$	θ_6

Training Goal: Use n datapoints to learn $2 \cdot 2 + 2 = 6$ parameters.

Training: Estimate parameters $\hat{P}(\boldsymbol{X}|Y)$



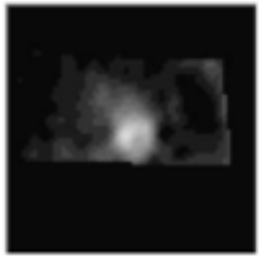
	Feature 1	Output
Patient 1	1	0
Patient 2	1	1
	$\vdots$	$\vdots$
Patient n	0	1

	$\hat{P}(\boldsymbol{X}\|Y=0)$	$\hat{P}(\boldsymbol{X}\|Y=1)$
$X_1 = 0$	θ_1	θ_3
$X_1 = 1$	θ_2	θ_4

$\hat{P}(\boldsymbol{X}|Y=0)$ and $\hat{P}(\boldsymbol{X}|Y=1)$
are both multinomials with 2 outcomes!

Use MLE or Laplace (MAP)
estimate for parameters $P(\boldsymbol{X}|Y)$

Training: MLE estimates, $\hat{P}(\mathbf{X}|Y)$



Feature 1

Output

	Feature 1	Output
Patient 1	1	0
Patient 2	1	1
	⋮	⋮
Patient n	0	1

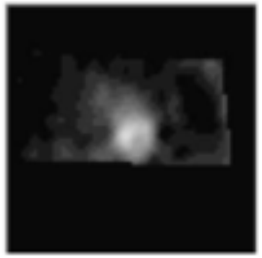
MLE

	$\hat{P}(\mathbf{X}\|Y=0)$	$\hat{P}(\mathbf{X}\|Y=1)$
$X_1 = 0$	0.4	0.0
$X_1 = 1$	0.6	1.0

MLE of $\hat{P}(X_1 = x|Y = y) = \dfrac{\#(X_1 = x, Y = y)}{\#(Y = y)}$

Just count!

Training: Laplace (MAP) estimates, $\hat{P}(\boldsymbol{X}|Y)$



	Feature 1	Output
Patient 1	1	0
Patient 2	1	1
	⋮	⋮
Patient n	0	1

MLE

	$\hat{P}(\boldsymbol{X}\|Y=0)$	$\hat{P}(\boldsymbol{X}\|Y=1)$
$X_1 = 0$	0.4	0.0
$X_1 = 1$	0.6	1.0

MLE of $\hat{P}(X_1 = x|Y = y) = \dfrac{\#(X_1 = x, Y = y)}{\#(Y = y)}$

Just count!

MAP

	$\hat{P}(\boldsymbol{X}\|Y=0)$	$\hat{P}(\boldsymbol{X}\|Y=1)$
$X_1 = 0$	0.42	0.01
$X_1 = 1$	0.58	0.99

Laplace of $\hat{P}(X_1 = x|Y = y) = \dfrac{\#(X_1 = x, Y = y) + 1}{\#(Y = y) + 2}$

Just count + add imaginary trials!

Testing

$$\hat{Y} = \arg\max_{y=\{0,1\}} \hat{P}(\mathbf{X}|Y)\hat{P}(Y)$$

(MAP)	$\hat{P}(\mathbf{X}\|Y=0)$	$\hat{P}(\mathbf{X}\|Y=1)$
$X_1 = 0$	0.42	0.01
$X_1 = 1$	0.58	0.99

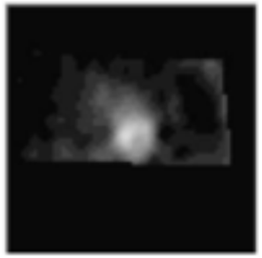
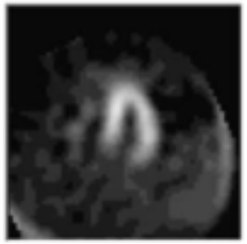
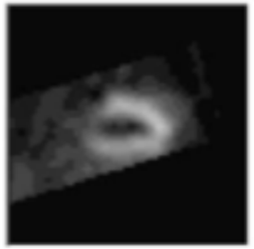
(MAP)	$\hat{P}(Y)$
$Y = 0$	0.10
$Y = 1$	0.90

New patient has a healthy ROI ($X_1 = 1$). What is your prediction, $\hat{Y}$?

$$\hat{P}(X_1 = 1|Y = 0)\hat{P}(Y = 0) = 0.58 \cdot 0.10 \approx 0.058$$
$$\hat{P}(X_1 = 1|Y = 1)\hat{P}(Y = 1) = 0.99 \cdot 0.90 \approx 0.891$$

A. $0.058 < 0.5 \Rightarrow \hat{Y} = 1$
B. $0.891 > 0.5 \Rightarrow \hat{Y} = 1$
C. $0.058 < 0.891 \Rightarrow \hat{Y} = 1$

Brute force Bayes: $m = 100$ (# features)


	Feature 1	Feature 2	...	Feature 100	Output
Patient 1	1	0	...	1	1
Patient 2	1	1	...	0	0
...			⋮		⋮
Patient n	0	0	...	1	1

This won't be too bad, right?

Brute force Bayes: $m = 100$ (# features)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \hat{P}(Y \mid \boldsymbol{X})$$

$$= \arg\max_{y=\{0,1\}} \frac{\hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)}{\hat{P}(\boldsymbol{X})}$$

$$= \arg\max_{y=\{0,1\}} \underbrace{\hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)}_{}$$

Learn parameters through MLE or MAP

- $\hat{P}(Y = 1 \mid \boldsymbol{x})$: estimated probability a heart is healthy given $\boldsymbol{x}$
- $\boldsymbol{X} = (X_1, X_2, \dots, X_{100})$: whether 100 regions of interest (ROI) are healthy (1) or unhealthy (0)

How many parameters do we have to learn?

	$\hat{P}(\boldsymbol{X}\|Y)$	$\hat{P}(Y)$	
A.	$2 \cdot 2$	$+ 2$	$= 6$
B.	$2 \cdot 100$	$+ 2$	$= 202$
C.	$2 \cdot 2^{100}$	$+ 2$	= a lot

This approach requires you to learn $O(2^m)$ parameters.

The problem with our Brute force Bayes classifier

$$\hat{Y} = \arg\max_{y=\{0,1\}} \hat{P}(Y \mid \boldsymbol{X})$$

$$= \arg\max_{y=\{0,1\}} \frac{\hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)}{\hat{P}(\boldsymbol{X})}$$

$$= \arg\max_{y=\{0,1\}} \boxed{\hat{P}(\boldsymbol{X}|Y)}\hat{P}(Y)$$

$$\hat{P}(X_1, X_2, \ldots, X_m|Y)$$

Estimating this joint conditional distribution will require too many parameters.

What if we could make a simplifying (but naïve) assumption– that $X_1, \ldots, X_m$ are **conditionally independent** given Y?

Today's plan

Maximum A Posteriori

- Picking a conjugate distribution as your prior
- Laplace smoothing

Machine Learning

- Inefficient classification: Brute force Bayes
- Naïve Bayes

The Naïve Bayes assumption

$X_1, \ldots, X_m$ are **conditionally independent** given Y.

Our prediction for Y is a function of $\boldsymbol{X}$

Choose the Y that is most likely given $\boldsymbol{X}$

$$\underbrace{\hat{Y} = g(\boldsymbol{X})}_{\text{Our prediction for } Y \text{ is a function of } \boldsymbol{X}} = \underbrace{\arg\max_{y=\{0,1\}} \hat{P}(Y \mid \boldsymbol{X})}_{\text{Choose the } Y \text{ that is most likely given } \boldsymbol{X}} = \arg\max_{y=\{0,1\}} \frac{\hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)}{\hat{P}(\boldsymbol{X})}$$ (Bayes)

$$= \arg\max_{y=\{0,1\}} \hat{P}(\boldsymbol{X}|Y)\hat{P}(Y)$$ (Normalization constant)

$$= \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y)\right)\hat{P}(Y)$$ Naïve Bayes Assumption

Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Training

What is the Big-O of # of parameters we need to learn?

A. $O(m)$
B. $O(2^m)$
C. other

Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Training

Use MLE or Laplace (MAP)

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

$$\hat{P}(X_i|Y = 0), \hat{P}(X_i|Y = 1)$$

$$\hat{P}(Y = 0), \hat{P}(Y = 1)$$

Testing

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\log \hat{P}(Y) + \sum_{i=1}^{m} \log \hat{P}(X_i|Y) \right)$$

(for numeric stability)

NETFLIX

and Learn

Naïve Bayes for TV shows

Will a user like the Pokémon TV series?

Observe indicator variables $\boldsymbol{X} = (X_1, X_2)$:

$X_1 = 1$:
"likes Star Wars"

$X_2 = 1$:
"likes Harry Potter"

Output Y indicator:

$Y = 1$:
"likes Pokémon"

Training: Naïve Bayes for TV shows (**MLE**)

$$\hat{Y} = \underset{y=\{0,1\}}{\arg\max} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: “likes Star Wars”
- X_2: “likes Harry Potter”

Predict Y: “likes Pokémon”

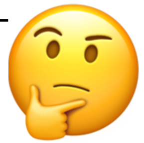
$Y \backslash X_1$	0	1
0	3	10
1	4	13

$Y \backslash X_2$	0	1
0	5	8
1	7	10

Training data counts

1. How many datapoints (n) are in our train data?
2. Compute MLE estimates for $\hat{P}(X_1|Y)$:

$Y \backslash X_1$	0	1
0	$\hat{P}(X_1 = 0 \mid Y = 0)$	$\hat{P}(X_1 = 1 \mid Y = 0)$
1	$\hat{P}(X_1 = 0 \mid Y = 1)$	$\hat{P}(X_1 = 1 \mid Y = 1)$



Training: Naïve Bayes for TV shows (MLE)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	3	10
1	4	13

Y \ X_2	0	1
0	5	8
1	7	10

Training data counts

1. How many datapoints (n) are in our train data?

$$n = 30$$

2. Compute MLE estimates for $\hat{P}(X_1|Y)$:

Y \ X_1	0	1
0	$3/13 \approx 0.23$	$10/13 \approx 0.77$
1	$4/17 \approx 0.24$	$13/17 \approx 0.76$

Training: Naïve Bayes for TV shows (MLE)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	3	10
1	4	13

Y \ X_2	0	1
0	5	8
1	7	10

Training data counts

Y \ X_1	0	1
0	0.23	0.77
1	0.24	0.76

Y \ X_2	0	1
0	$5/13 \approx 0.38$	$8/13 \approx 0.62$
1	$7/17 \approx 0.41$	$10/17 \approx 0.59$

Y	
0	$13/30 \approx 0.43$
1	$17/30 \approx 0.57$

👉 Training MLE estimates: just count.

$$\hat{P}(X_i = x|Y = y) = \frac{\#(X_i = x, Y = y)}{\#(Y = y)}$$

$$\hat{P}(Y = y) = \frac{\#(Y = y)}{n}$$

Training : Naïve Bayes for TV shows (MLE)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	0.23	0.77
1	0.24	0.76

Y \ X_2	0	1
0	0.38	0.62
1	0.41	0.59

Y	
0	0.43
1	0.57

Now that we've trained and found parameters,
It's time to classify new users!

Testing: Naïve Bayes for TV shows (MLE)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y)\right)\hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	0.23	0.77
1	0.24	0.76

Y \ X_2	0	1
0	0.38	0.62
1	0.41	0.59

Y	
0	0.43
1	0.57

Suppose a **new person** "likes Star Wars" ($X_1 = 1$) but "dislikes Harry Potter" ($X_2 = 0$).

Will they like Pokemon? Need to predict Y:

$$\hat{Y} = \arg\max_{y=\{0,1\}} \hat{P}(\boldsymbol{X}|Y)\hat{P}(Y) \quad = \arg\max_{y=\{0,1\}} \hat{P}(X_1|Y)\hat{P}(X_2|Y)\hat{P}(Y)$$

If $Y = 0$: $\quad \hat{P}(X_1 = 1|Y = 0)\hat{P}(X_2 = 0|Y = 0)\hat{P}(Y = 0) = 0.77 \cdot 0.38 \cdot 0.43 = 0.126$

If $Y = 1$: $\quad \hat{P}(X_1 = 1|Y = 1)\hat{P}(X_2 = 0|Y = 1)\hat{P}(Y = 1) = 0.76 \cdot 0.41 \cdot 0.57 = 0.178$

Since term is greatest when Y = 1, predict $\hat{Y} = 1$

Training: Naïve Bayes for TV shows (MAP)

$$\hat{Y} = \underset{y=\{0,1\}}{\arg\max} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	3	10
1	4	13

Y \ X_2	0	1
0	5	8
1	7	10

Training data counts

What are our MAP estimates using Laplace smoothing for $\hat{P}(X_i|Y)$ and $\hat{P}(Y)$?

$\hat{P}(X_i = x|Y = y)$:

A. $\frac{\#(X_i=x, Y=y)}{\#(Y=y)}$

B. $\frac{\#(X_i=x, Y=y)+1}{\#(Y=y)+2}$

C. $\frac{\#(X_i=x, Y=y)+1}{\#(Y=y)+4}$

$\hat{P}(Y = y)$:

A. $\frac{\#(Y=y)}{\#(Y=y)+2}$

B. $\frac{\#(Y=y)+1}{n}$

C. $\frac{\#(Y=y)+1}{n+2}$

Training: Naïve Bayes for TV shows (MAP)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y)\right)\hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

$Y \backslash X_1$	0	1
0	3	10
1	4	13

$Y \backslash X_2$	0	1
0	5	8
1	7	10

Training data

What are our MAP estimates using Laplace smoothing for $\hat{P}(X_i|Y)$ and $\hat{P}(Y)$?

$\hat{P}(X_i = x|Y = y)$:

A. $\dfrac{\#(X_i=x, Y=y)}{\#(Y=y)}$

B. $\dfrac{\#(X_i=x, Y=y)+1}{\#(Y=y)+2}$

C. $\dfrac{\#(X_i=x, Y=y)+1}{\#(Y=y)+4}$

$\hat{P}(Y = y)$:

A. $\dfrac{\#(Y=y)}{\#(Y=y)+2}$

B. $\dfrac{\#(Y=y)+1}{n}$

C. $\dfrac{\#(Y=y)+1}{n+2}$

Training: Naïve Bayes for TV shows (MAP)

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y)\right)\hat{P}(Y)$$

Observe indicator vars. $\boldsymbol{X} = (X_1, X_2)$:

- X_1: "likes Star Wars"
- X_2: "likes Harry Potter"

Predict Y: "likes Pokémon"

Y \ X_1	0	1
0	3	10
1	4	13

Y \ X_2	0	1
0	5	8
1	7	10

Training data

Y \ X_1	0	1
0	0.27	0.73
1	0.26	0.74

Y \ X_2	0	1
0	0.40	0.60
1	0.42	0.58

Y	
0	$14/32 \approx 0.44$
1	$18/32 \approx 0.56$

👉 Training MAP estimates: just count + imaginary trials.

$$\hat{P}(X_i = x|Y = y) = \frac{\#(X_i = x, Y = y) + 1}{\#(Y = y) + 2}$$

$$\hat{P}(Y = y) = \frac{\#(Y = y) + 1}{n + 2}$$

🤔

Naïve Bayes Model is a Bayesian Network

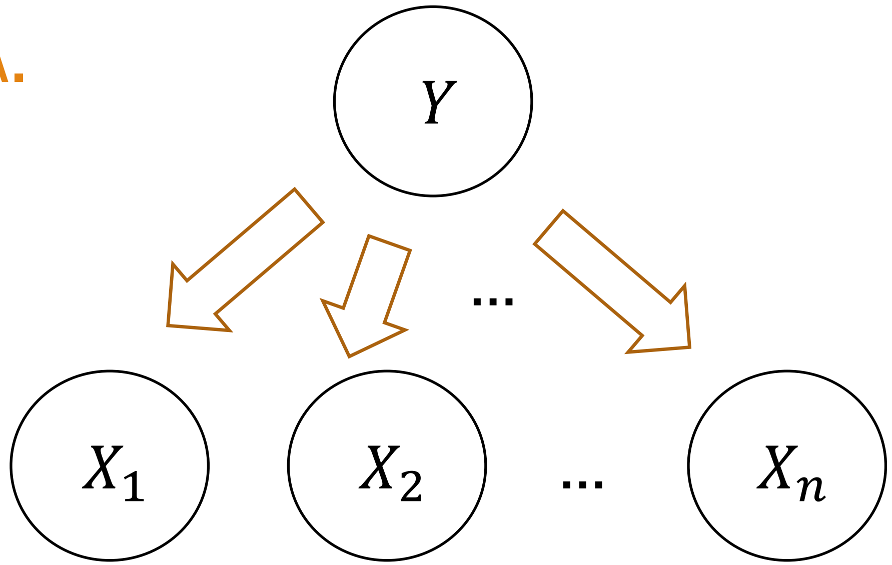
$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Naïve Bayes Assumption

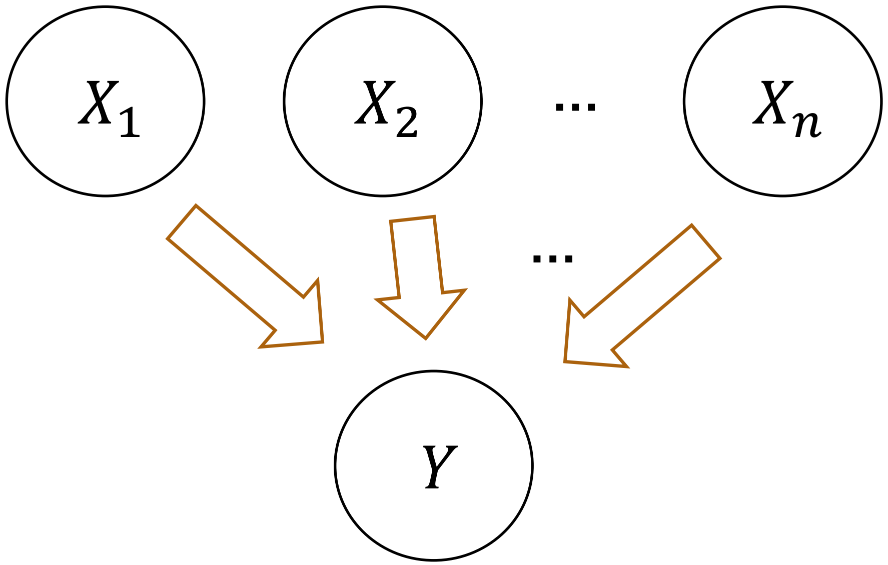
$$P(\mathbf{X}|Y) = \prod_{i=1}^{m} P(X_i|Y) \quad \Rightarrow \quad P(\mathbf{X}, Y) = P(Y) \prod_{i=1}^{m} P(X_i|Y)$$

Which Bayesian Network encodes this conditional independence?

A.



B.



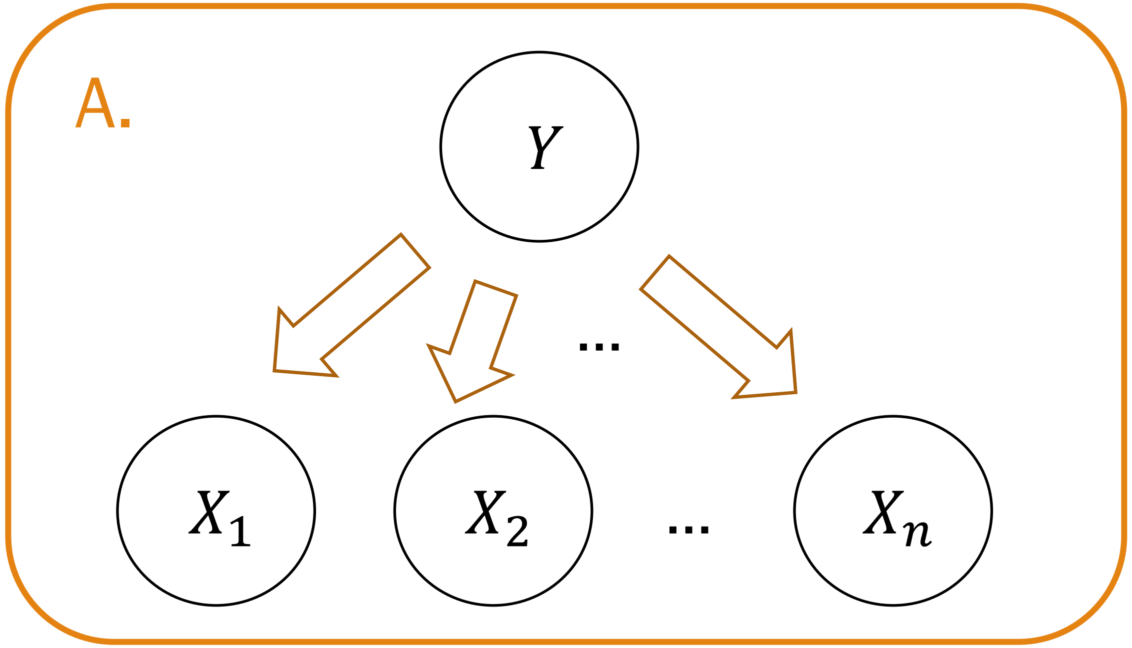
$$\hat{Y} = \underset{y=\{0,1\}}{\arg\max} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Naïve Bayes Model is a Bayesian Network

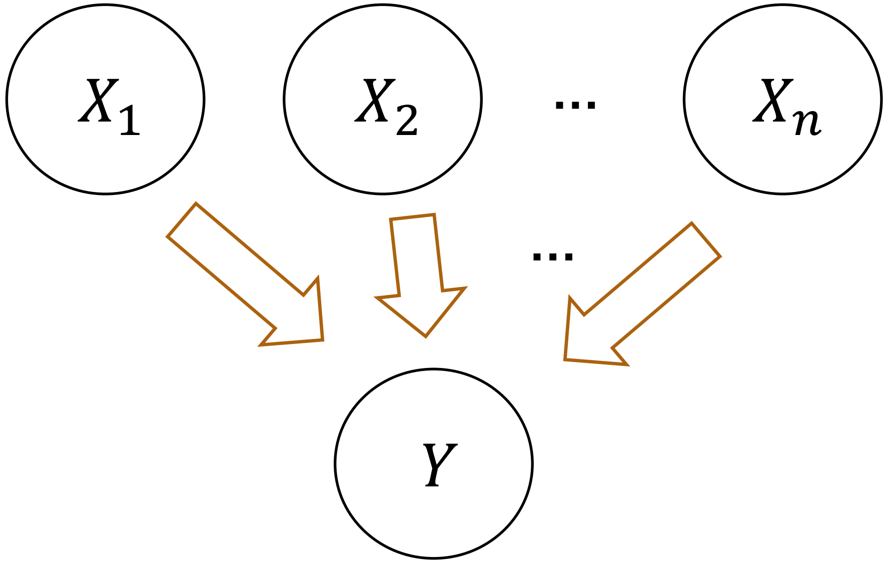
Naïve Bayes Assumption

$$P(\boldsymbol{X}|Y) = \prod_{i=1}^{m} P(X_i|Y) \quad \Rightarrow \quad P(\boldsymbol{X}, Y) = P(Y) \prod_{i=1}^{m} P(X_i|Y)$$

Which Bayesian Network encodes this conditional independence?



B.

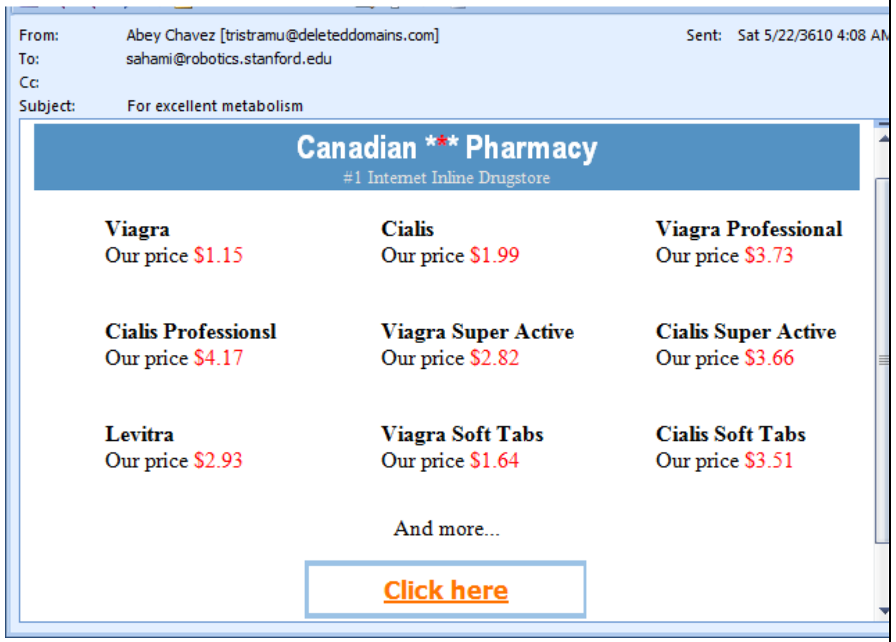


X_i are conditionally independent given parent Y

Extra slides

Naïve Bayes with spam classification

What is Bayes doing in my mail server?



Let's get Bayesian on your spam:

```
Content analysis details:   (49.5 hits, 7.0 required)
 0.9 RCVD_IN_PBL            RBL: Received via a relay in Spamhaus PBL
                            [93.40.189.29 listed in zen.spamhaus.org]
 1.5 URIBL_WS_SURBL         Contains an URL listed in the WS SURBL blocklist
                            [URIs: recragas.cn]
 5.0 URIBL_JP_SURBL         Contains an URL listed in the JP SURBL blocklist
                            [URIs: recragas.cn]
 5.0 URIBL_OB_SURBL         Contains an URL listed in the OB SURBL blocklist
                            [URIs: recragas.cn]
 5.0 URIBL_SC_SURBL         Contains an URL listed in the SC SURBL blocklist
                            [URIs: recragas.cn]
 2.0 URIBL_BLACK            Contains an URL listed in the URIBL blacklist
                            [URIs: recragas.cn]
 8.0 BAYES_99               BODY: Bayesian spam probability is 99 to 100%
                            [score: 1.0000]
```

A Bayesian Approach to Filtering Junk E-Mail

Mehran Sahami* Susan Dumais† David Heckerman† Eric Horvitz†

*Gates Building 1A
Computer Science Department
Stanford University
Stanford, CA 94305-9010
sahami@cs.stanford.edu

†Microsoft Research
Redmond, WA 98052-6399
{sdumais, heckerma, horvitz}@microsoft.com

Abstract

In addressing the growing problem of junk E-mail on the Internet, we examine methods for the automated

contain offensive material (such as graphic pornography), there is often a higher cost to users of actually viewing this mail than simply the time to sort out the

Email classification

Goal Based on email content $\boldsymbol{X}$, predict if email is spam or not.

Features Consider a lexicon m words (for English: $m \approx 100{,}000$).

$\boldsymbol{X} = (X_1, X_2, \dots, X_m)$, m indicator variables

$X_i = 1$ if word i appeared in document

Output $Y = 1$ if email is spam

Note: m is huge. Make Naïve Bayes assumption: $P(\boldsymbol{X}|\text{spam}) = \prod_{i=1}^{m} P(X_i|\text{spam})$

Appearances of words in email are conditionally independent given the email is spam or not

Naïve Bayes Email classification

Train set n previous emails $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ for each word, whether it appears in email j

$y^{(j)} = 1$ if spam, 0 if not spam

Training

Estimate probabilities $\hat{P}(Y)$ and $\hat{P}(X_i|Y)$ for all i

Which estimator should we use?

A. MLE
B. Laplace estimate (MAP)
C. Other MAP estimate
D. Both A and B

Naïve Bayes Email classification

Train set n previous emails $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(n)}, y^{(n)})$

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \dots, x_m^{(j)}\right)$ for each word, whether it appears in email j

$y^{(j)} = 1$ if spam, 0 if not spam

Training

Estimate probabilities
$\hat{P}(Y)$ and $\hat{P}(X_i|Y)$ for all i

Which estimator should we use?

A. MLE
B. Laplace estimate (MAP)
C. Other MAP estimate
D. Both A and B

- Many words are likely to not appear at all in the training set, so we want to avoid 0 probabilities.
- Laplace estimate is simple.

Naïve Bayes Email classification

Train set n previous emails $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ for each word, whether it appears in email j

$y^{(j)} = 1$ if spam, 0 if not spam

Training

Estimate probabilities
$\hat{P}(Y)$ and $\hat{P}(X_i|Y)$ for all i

Laplace estimate: $$\hat{P}(X_i = 1|Y = \text{spam}) = \frac{(\#\text{ spam emails with word } i) + 1}{(\text{total }\#\text{ spam emails}) + 2}$$

Testing (Classification)

For a new email:
- Generate $\boldsymbol{X} = (X_1, X_2, \ldots, X_m)$
- Classify as spam or not using Naïve Bayes assumption

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\log \hat{P}(Y) + \sum_{i=1}^{m} \log \hat{P}(X_i|Y) \right)$$

Use logs for numeric stability

How well does Naïve Bayes perform?

After training, you can test with another set of data, called the **test set**.

- Test set also has known values for Y so we can see how often we were right/wrong in our predictions $\hat{Y}$.

Typical work flow:

- Have a dataset of 1789 emails (1578 spam, 211 ham)
- Train set: First 1538 emails (by time)
- Test set: Next 251 messages

Evaluation criteria on test set:

$$\textbf{precision} = \frac{(\text{\# correctly predicted class } Y)}{(\text{\# predicted class } Y)}$$

$$\textbf{recall} = \frac{(\text{\# correctly predicted class } Y)}{(\text{\# real class } Y \text{ messages})}$$

	Spam		Non-spam	
	Prec.	Recall	Prec.	Recall
Words only	97.1%	94.3%	87.7%	93.4%
Words + addtl features	100%	98.3%	96.2%	100%