

24: Naïve Bayes

Jerry Cain

May 20, 2022

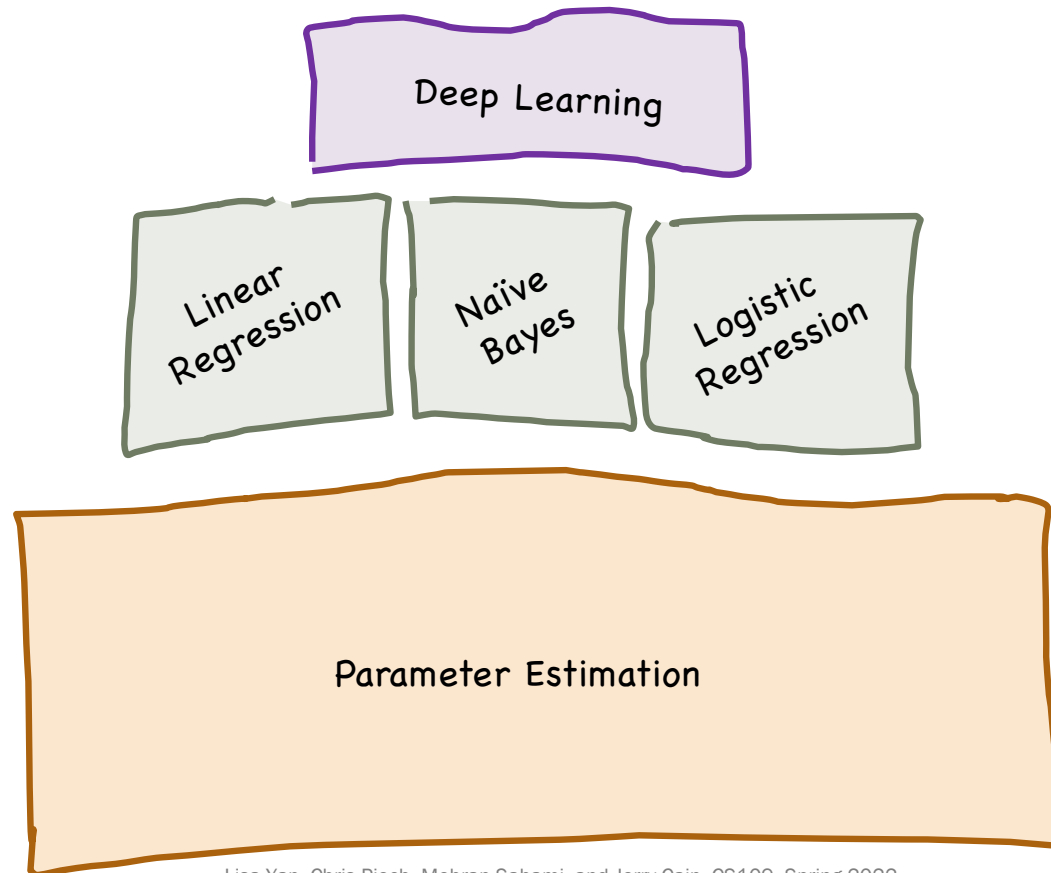
Table of Contents

| | |
|----|-------------------------|
| 2 | Intro: Machine Learning |
| 18 | Brute Force Bayes |
| 29 | Naïve Bayes |
| 38 | Netflix and Learning |
| 56 | Spam and Learning |



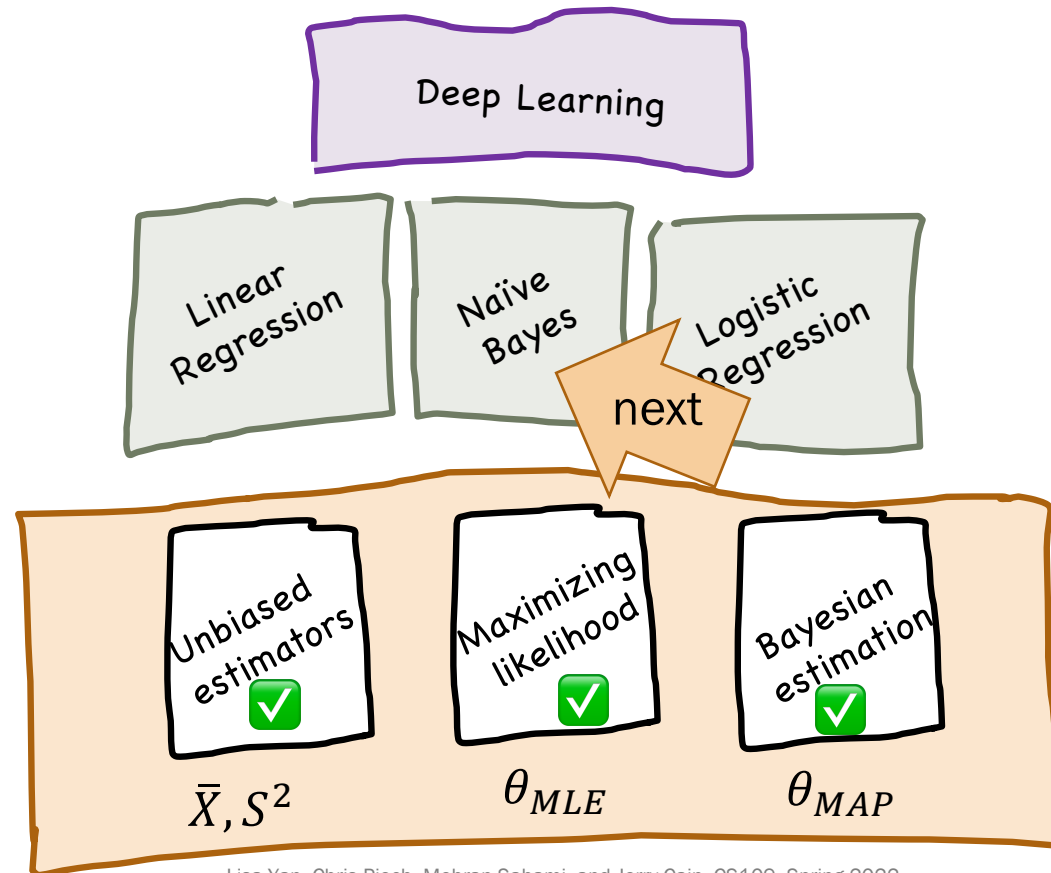
Intro: Machine Learning

Our path from here



Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022

Our path from here



Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022

Machine Learning (formally)

Many different forms of Machine Learning

- We focus on the problem of **prediction** given prior observations.

Machine Learning uses a lot of data.

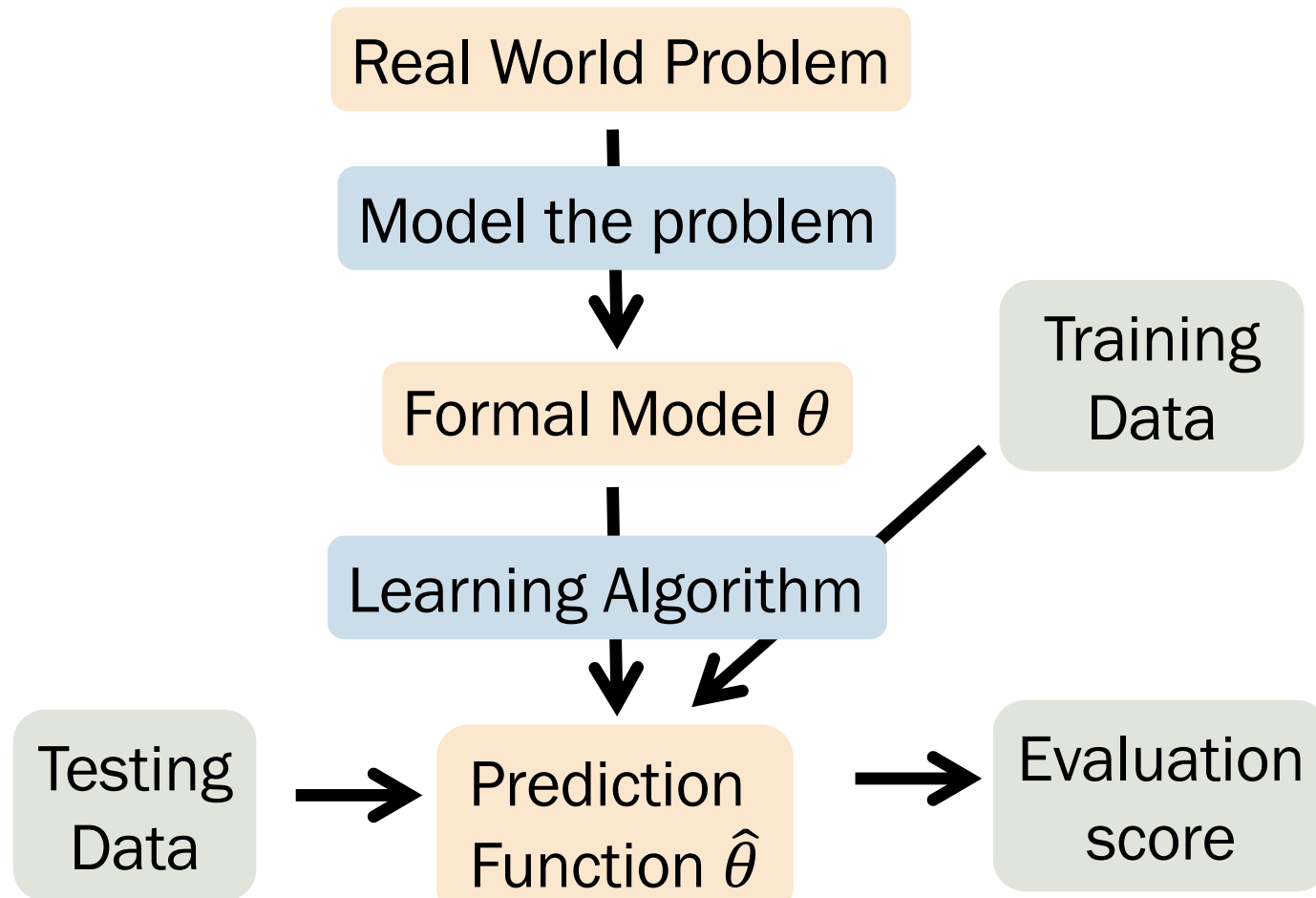


Task: Identify the chair

Data: All the chairs ever

Supervised learning: A category of machine learning where you have labeled data for the problem you are solving.

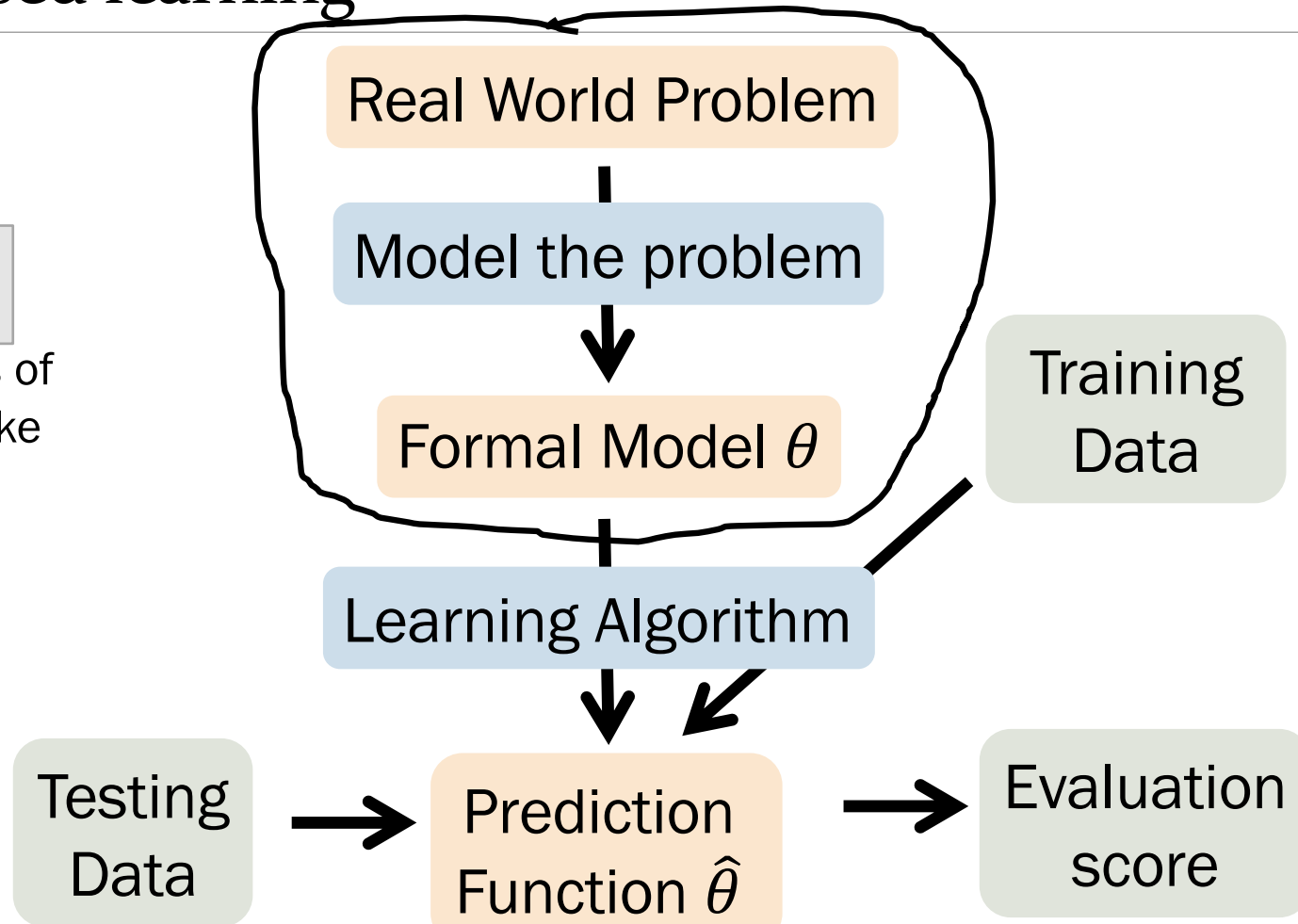
Supervised learning



Supervised learning

Modeling

not the focus of this class (take CS228)

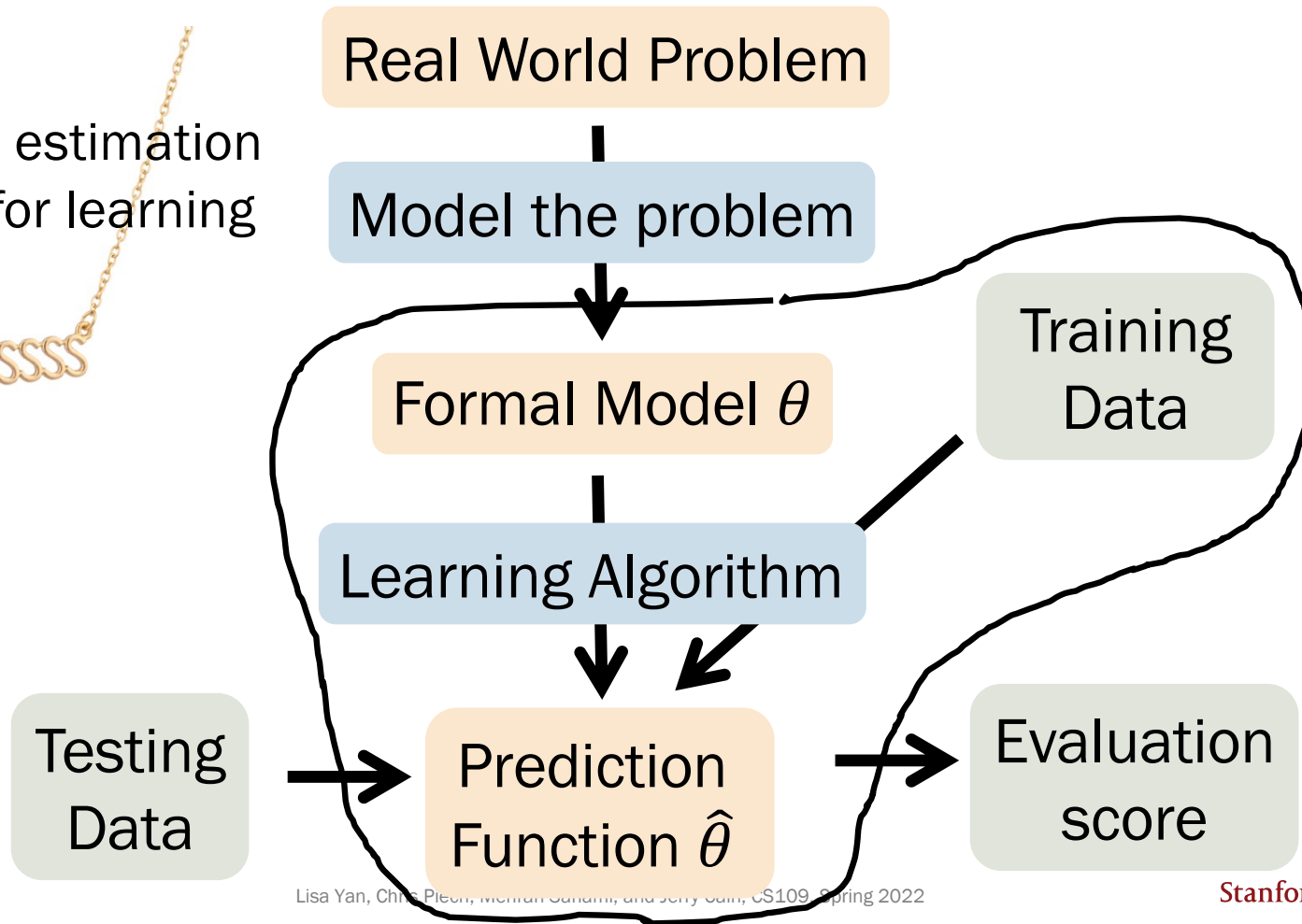


Supervised learning

Parameter estimation
is a basis for learning
from data.

YASSSSS

Training



Model and dataset

Many different forms of machine learning

- We focus on a specific type of problem: **prediction** from observations.

Goal

Based on observed \mathbf{X} , predict some unknown Y

- **Features**

Vector \mathbf{X} of m observations (new term: **feature vector**)

$$\mathbf{X} = (X_1, X_2, \dots, X_m)$$

- **Output**

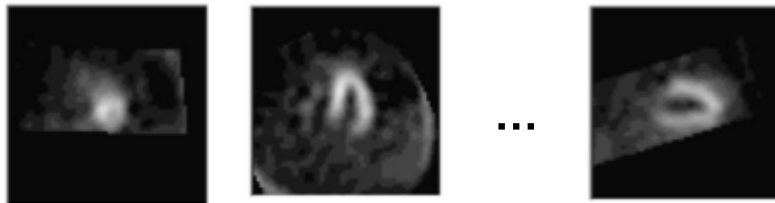
Variable Y (also called **class label** if discrete)

Model

$$\hat{Y} = g(\mathbf{X}), \text{ a function on } \mathbf{X}$$

Training data

$$\mathbf{X} = (X_1, X_2, X_3, \dots, X_{300})$$



Feature 1 Feature 2 ... Feature 300



Output

| | | | | | |
|-------------|---|---|-----|---|---|
| Patient 1 | 1 | 0 | ... | 1 | 1 |
| Patient 2 | 1 | 1 | ... | 0 | 0 |
| ... | | | ⋮ | | ⋮ |
| Patient n | 0 | 0 | ... | 1 | 1 |

Training data notation

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$$

n datapoints, assumed to be iid

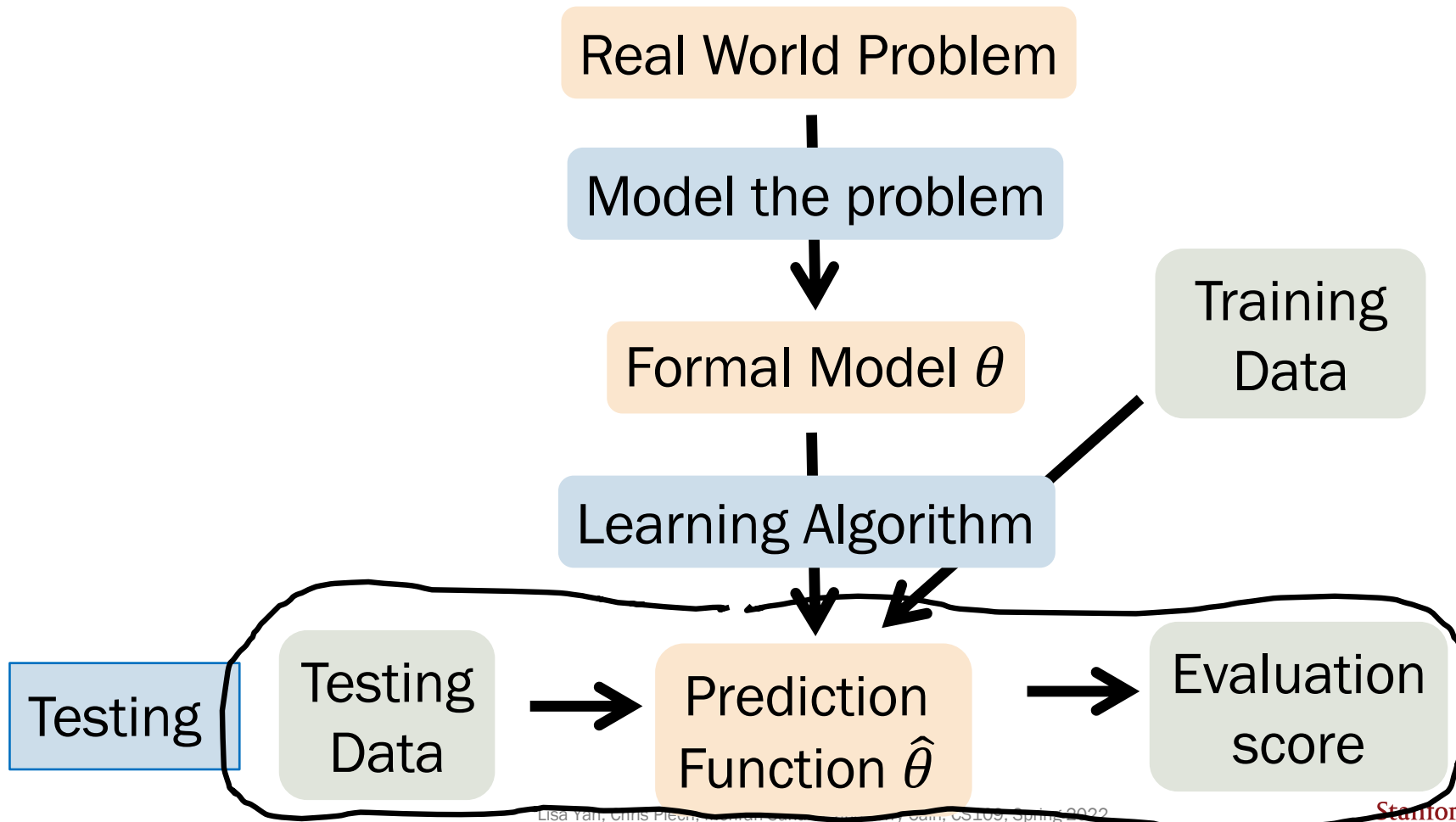
i -th datapoint $(\mathbf{x}^{(i)}, y^{(i)})$:

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

Training Goal:

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

Supervised learning



Testing data notation

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$$

n_{test} other datapoints, assumed to be iid

i -th datapoint $(\mathbf{x}^{(i)}, y^{(i)})$:

- Has the same structure as your training data

Testing Goal:

Leveraging the model $\hat{Y} = g(\mathbf{X})$ that you trained, see how well you can predict Y on known data

Two tasks we will focus on

Many different forms of “Machine Learning”

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

Goal Based on observed \mathbf{X} , predict some unknown Y

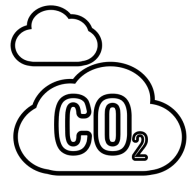
- **Features** Vector \mathbf{X} of m observations (new term: feature vector)
 $\mathbf{X} = (X_1, X_2, \dots, X_m)$
- **Output** Variable Y (also called **class label** if discrete)

Model $\hat{Y} = g(\mathbf{X})$, a function on \mathbf{X}

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

Regression: Predicting real numbers

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



CO2 levels

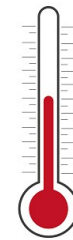


Sea level

...



Feature m



Global Land-Ocean temperature

Output

Year 1

338.8

0

...

1

Year 2

340.0

1

...

0

...

⋮

Year n

340.76

0

...

1

0.26

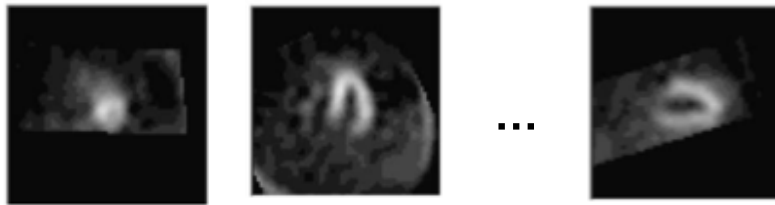
0.32

⋮

0.14

Classification: Predicting class labels

$$\mathbf{X} = (X_1, X_2, X_3, \dots, X_{300})$$

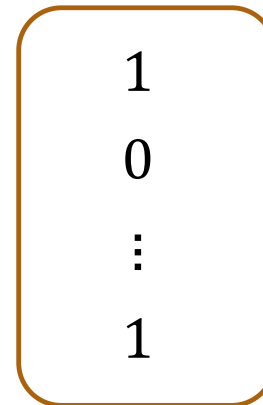


Feature 1 Feature 2 ... Feature 300



Output

| | | | | |
|-------------|---|---|-----|---|
| Patient 1 | 1 | 0 | ... | 1 |
| Patient 2 | 1 | 1 | ... | 0 |
| ... | | | ⋮ | |
| Patient n | 0 | 0 | ... | 1 |

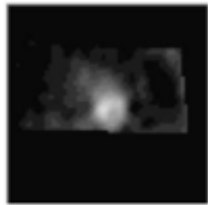




Brute Force Bayes

Classification: Having a healthy heart

$$\mathbf{X} = (X_1)$$



Feature 1



Output

| | | |
|-------------|---|---|
| Patient 1 | 1 | 0 |
| Patient 2 | 1 | 1 |
| | ⋮ | ⋮ |
| Patient n | 0 | 1 |

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

How can we predict whether heart is healthy (1) or unhealthy (0)?

The following strategy is **not used in practice** but helps us understand how to approach classification.

Classification: Brute Force Bayes

$$\hat{Y} = g(\mathbf{X})$$

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

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

Proposed model: Choose the Y that is
more or most likely given \mathbf{X}

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

(Bayes' Theorem)

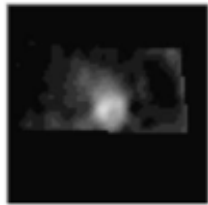
$$= \arg \max_{y=\{0,1\}} P(\mathbf{X}|Y)P(Y)$$

($1/P(\mathbf{X})$ is constant w.r.t. y)

If we estimate $P(\mathbf{X}|Y)$ and $P(Y)$, we can classify datapoints!

Training: Estimate parameters

$$\mathbf{X} = (X_1)$$



Feature 1



Output

Patient 1 1
 Patient 2 1
 ⋮
 Patient n 0

0
 1
 ⋮
 1

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

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

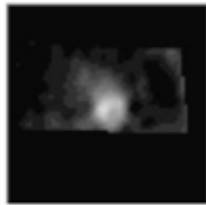
$$\hat{Y} = \arg \max_{y=\{0,1\}} \hat{P}(\mathbf{X}|Y) \hat{P}(Y)$$

| | $\hat{P}(\mathbf{X} Y = 0)$ | $\hat{P}(\mathbf{X} Y = 1)$ |
|--------------|-----------------------------|-----------------------------|
| $X_1 = 0$ | θ_1 | θ_3 |
| $X_1 = 1$ | θ_2 | θ_4 |
| $\hat{P}(Y)$ | | |
| $Y = 0$ | θ_5 | |
| $Y = 1$ | θ_6 | |

Training Goal:

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

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



| Count: | <u># datapoints</u> |
|--------------------|---------------------|
| $X_1 = 0, Y = 0$: | 4 |
| $X_1 = 1, Y = 0$: | 6 |
| $X_1 = 0, Y = 1$: | 0 |
| $X_1 = 1, Y = 1$: | 100 |
| Total: | 110 |

Patient n 0

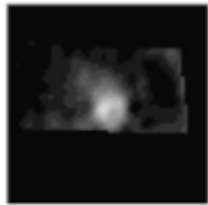
1

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

$\mathbf{X}|Y = 0$ and $\mathbf{X}|Y = 1$
are each multinomials with 2 outcomes!

Use MLE or Laplace (MAP) estimate $\hat{P}(\mathbf{X}|Y)$ and $\hat{P}(Y)$ as parameters.

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



| Count: | # datapoints |
|--------------------|--------------|
| $X_1 = 0, Y = 0$: | 4 |
| $X_1 = 1, Y = 0$: | 6 |
| $X_1 = 0, Y = 1$: | 0 |
| $X_1 = 1, Y = 1$: | 100 |
| Total: | 110 |

Patient n 0

1

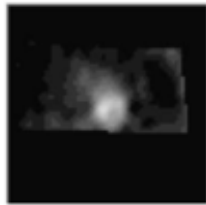
| | $\hat{P}(X Y = 0)$ | $\hat{P}(X Y = 1)$ |
|-----------|--------------------|--------------------|
| $X_1 = 0$ | 0.4 | 0.0 |
| $X_1 = 1$ | 0.6 | 1.0 |

MLE

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

Just count!

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



| Count: | # datapoints |
|--------------------|--------------|
| $X_1 = 0, Y = 0$: | 4 |
| $X_1 = 1, Y = 0$: | 6 |
| $X_1 = 0, Y = 1$: | 0 |
| $X_1 = 1, Y = 1$: | 100 |
| Total: | 110 |

Pa

Pa

Patient n 0

1

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

MLE

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

Just count!

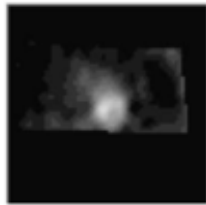
MAP

Laplace of $\hat{P}(X_1 = x|Y = y) = ?$

Just count + add imaginary trials!



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



| Count: | # datapoints |
|--------------------|--------------|
| $X_1 = 0, Y = 0$: | 4 |
| $X_1 = 1, Y = 0$: | 6 |
| $X_1 = 0, Y = 1$: | 0 |
| $X_1 = 1, Y = 1$: | 100 |
| Total: | 110 |

Pa
Pa

Patient n 0

1



MLE of $\hat{P}(X_1 = x|Y = y) = \frac{\#(X_1 = x, Y = y)}{\#(Y = y)}$
Just count!



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

Laplace of $\hat{P}(X_1 = x|Y = y) = \frac{\#(X_1 = x, Y = y) + 1}{\#(Y = y) + 2}$
Just count + add imaginary trials!

Testing

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

| (MAP) | $\hat{P}(\mathbf{X} Y = 0)$ | $\hat{P}(\mathbf{X} Y = 1)$ | (MLE) | $\hat{P}(Y)$ |
|-----------|-----------------------------|-----------------------------|---------|--------------|
| $X_1 = 0$ | 0.42 | 0.01 | $Y = 0$ | 0.09 |
| $X_1 = 1$ | 0.58 | 0.99 | $Y = 1$ | 0.91 |

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

$$\hat{P}(X_1 = 1|Y = 0) \hat{P}(Y = 0) = 0.58 \cdot 0.09 \approx 0.052$$

$$\hat{P}(X_1 = 1|Y = 1) \hat{P}(Y = 1) = 0.99 \cdot 0.91 \approx 0.901$$

- A. $0.052 < 0.5 \Rightarrow \hat{Y} = 1$
- B. $0.901 > 0.5 \Rightarrow \hat{Y} = 1$
- C. $0.052 < 0.901 \Rightarrow \hat{Y} = 1$

Sanity check: Why don't these sum to 1?

Testing

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

| (MAP) | $\hat{P}(\mathbf{X} Y = 0)$ | $\hat{P}(\mathbf{X} Y = 1)$ | (MLE) | $\hat{P}(Y)$ |
|-----------|-----------------------------|-----------------------------|---------|--------------|
| $X_1 = 0$ | 0.42 | 0.01 | $Y = 0$ | 0.09 |
| $X_1 = 1$ | 0.58 | 0.99 | $Y = 1$ | 0.91 |

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

$$\hat{P}(X_1 = 1|Y = 0) \hat{P}(Y = 0) = 0.58 \cdot 0.09 \approx 0.052$$

$$\hat{P}(X_1 = 1|Y = 1) \hat{P}(Y = 1) = 0.99 \cdot 0.91 \approx 0.901$$

- A. $0.052 < 0.5 \Rightarrow \hat{Y} = 1$
- B. $0.901 > 0.5 \Rightarrow \hat{Y} = 1$
- C. $0.052 < 0.901 \Rightarrow \hat{Y} = 1$**

Sanity check: Why don't these sum to 1?

Brute Force Bayes classifier

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

$(\hat{P}(Y))$ is an estimate of $P(Y)$,
 $\hat{P}(\mathbf{X}|Y)$ is an estimate of $P(\mathbf{X}|Y)$

Training

Estimate these probabilities—i.e., learn these parameters using MLE or Laplace (MAP)

$$\begin{aligned} &\hat{P}(X_1, X_2, \dots, X_m | Y = 1) \\ &\hat{P}(X_1, X_2, \dots, X_m | Y = 0) \\ &\hat{P}(Y = 1) \quad \hat{P}(Y = 0) \end{aligned}$$

Testing

Given an observation $\mathbf{X} = (X_1, X_2, \dots, X_m)$, predict

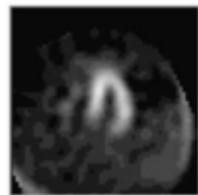
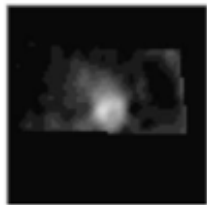
$$\hat{Y} = \arg \max_{y=\{0,1\}} \left(\hat{P}(X_1, X_2, \dots, X_m | Y) \hat{P}(Y) \right)$$



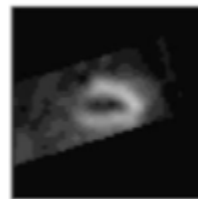
Naïve Bayes

Brute Force Bayes: $m = 300$ (# features)

$$\mathbf{X} = (X_1, X_2, X_3, \dots, X_{300})$$



...



Feature 1

Feature 2

Feature 300

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: $m = 300$ (# features)

$$\mathbf{X} = (X_1, X_2, X_3, \dots, X_{300})$$



| | Count: | <u># datapoints</u> |
|-------------|--|---------------------|
| | $X_1 = 0, X_2 = 0, \dots, X_{299} = 0, X_{300} = 0, Y = 0:$ | 0 |
| | $X_1 = 0, X_2 = 0, \dots, X_{299} = 0, X_{300} = 1, Y = 0:$ | 0 |
| | $X_1 = 0, X_2 = 0, \dots, X_{299} = 1, X_{300} = 0, Y = 0:$ | 1 |
| Pat | ... | |
| Pat | $X_1 = 0, X_2 = 0, \dots, X_{299} = 0, X_{300} = 0, Y = 1:$ | 2 |
| | $X_1 = 0, X_2 = 0, \dots, X_{299} = 0, X_{300} = 1, Y = 1:$ | 1 |
| | $X_1 = 0, X_2 = 0, \dots, X_{299} = 1, X_{300} = 0, Y = 1:$ | 1 |
| Patient n | 0 0 ... 1 | 1 |

This won't be too bad, right?

Brute Force Bayes: $m = 300$ (# features)

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

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

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

Learn parameters
through MLE or MAP

- $\hat{P}(Y = 1 | \mathbf{x})$: estimated probability a heart is healthy given \mathbf{x}
- $\mathbf{X} = (X_1, X_2, \dots, X_{300})$: whether 300 regions of interest (ROI) are healthy (1) or unhealthy (0)

How many parameters do we have to learn?

- | | $\hat{P}(\mathbf{X} Y)$ | $\hat{P}(Y)$ | |
|----|-------------------------|--------------|------------------|
| A. | $2 \cdot 2$ | $+ 2$ | $= 6$ |
| B. | $2 \cdot 300$ | $+ 2$ | $= 602$ |
| C. | $2 \cdot 2^{300}$ | $+ 2$ | $= \text{a lot}$ |



Brute Force Bayes: $m = 300$ (# features)

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

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

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

Learn parameters
through MLE or MAP

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

- $\hat{P}(Y = 1 | \mathbf{x})$: estimated probability a heart is healthy given \mathbf{x}
- $\mathbf{X} = (X_1, X_2, \dots, X_{300})$: whether 300 regions of interest (ROI) are healthy (1) or unhealthy (0)

How many parameters do we have to learn?

- | | $\hat{P}(\mathbf{X} Y)$ | $\hat{P}(Y)$ | |
|----|-------------------------|--------------|------------------|
| A. | $2 \cdot 2$ | $+ 2$ | $= 6$ |
| B. | $2 \cdot 300$ | $+ 2$ | $= 602$ |
| C. | $2 \cdot 2^{300}$ | $+ 2$ | $= \text{a lot}$ |

The problem with our current classifier

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


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

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

(Bayes' Theorem)

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

($1/P(\mathbf{X})$ is constant w.r.t. y)


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

Estimating this joint conditional distribution is intractable.

What if we could make a simplifying assumption—even if incredibly naïve—to make our parameter estimation effort computationally tractable?

The Naïve Bayes assumption

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

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

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

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

Assumption:

X_1, \dots, X_m are all **conditionally independent** given Y .

Naïve Bayes
Assumption

Naïve Bayes Classifier

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

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

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

Training

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

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

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

Use MLE or
Laplace (MAP)

Testing

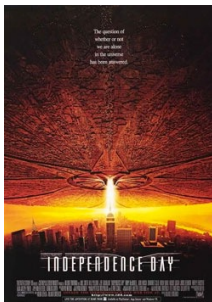
$$\hat{Y} = \arg \max_{y=\{0,1\}} \left(\log \hat{P}(Y) + \sum_{j=1}^m \log \hat{P}(X_j|Y) \right)$$

NETFLIX

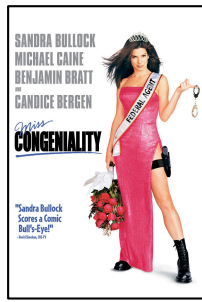
and Learn

Classification terminology check

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

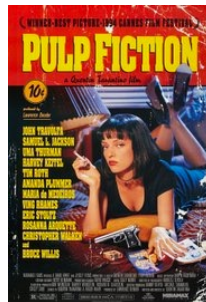


Movie 1



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 |

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

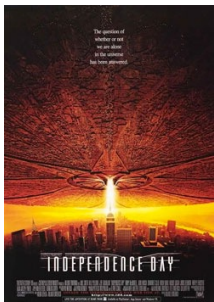
1: like movie

0: dislike movie

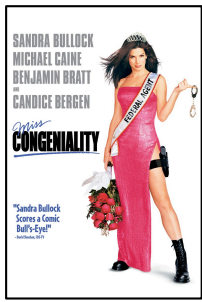


Classification terminology check

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

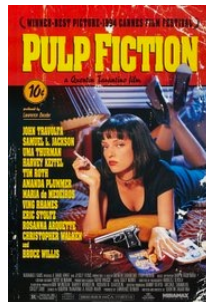


Movie 1



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 |

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

1: like movie

0: dislike movie

i : i -th user

j : movie j

Predicting user TV preferences

Will a user like the Pokémon TV series?

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



$X_1 = 1$:
"likes Star Wars"



$X_2 = 1$:
"likes Harry Potter"

Output Y indicator:



$Y = 1$:
"likes Pokémon"

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

Predicting user TV preferences

Which probabilities do you need to estimate?
How many are there?

- Brute Force Bayes
(strawman, without NB assumption)
- Naïve Bayes

During training, how to estimate the prob
 $\hat{P}(X_1 = 1, X_2 = 1|Y = 0)$ with MLE? with Laplace?

- Brute Force Bayes
- Naïve Bayes

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

Naïve Bayes Assumption $P(X|Y) = \prod_{j=1}^m P(X_j|Y)$



Predicting user TV preferences

Which probabilities do you need to estimate?

How many are there?

- Brute Force Bayes
(strawman, without NB assumption)
- Naïve Bayes

During training, how to estimate the prob

$\hat{P}(X_1 = 1, X_2 = 1 | Y = 0)$ with MLE? with Laplace?

- Brute Force Bayes
- Naïve Bayes

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

Naïve Bayes
Assumption

$$P(X|Y) = \prod_{j=1}^m P(X_j|Y)$$

(Strawman Brute Force) Multinomial MLE and MAP

Model: Multinomial, m outcomes:
 p_j probability of outcome j

Observe: $n_j = \#$ of trials with outcome j
Total of $\sum_{j=1}^m n_j$ trials

MLE

$$\hat{p}_j = \frac{n_j}{\sum_{j=1}^m n_j}$$

Laplace estimate

(MAP w/Laplace smoothing)

$$\hat{p}_j = \frac{n_j + 1}{\sum_{j=1}^m n_j + m}$$

| X_1 | X_2 | Y |
|-------|-------|-----|
| 0 | 1 | 1 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| ... | ... | ... |
| 1 | 1 | 1 |

training data

$$\hat{P}(X_1 = 1 \ X_2 = 1 | Y = 0)$$

(Naïve Bayes) Multinomial MLE and MAP

Model: Multinomial, m outcomes:

p_j probability of outcome j

MLE

$$\hat{p}_j = \frac{n_j}{\sum_{j=1}^m n_j}$$

Observe: $n_j = \#$ of trials with outcome j

Total of $\sum_{j=1}^m n_j$ trials

Laplace estimate

(MAP w/Laplace smoothing)

$$\hat{p}_j = \frac{n_j + 1}{\sum_{j=1}^m n_j + m}$$

| X_1 | X_2 | Y |
|-------|-------|-----|
| 0 | 1 | 1 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| ... | ... | ... |
| 1 | 1 | 1 |

training data

$$\hat{P}(X_1 = 1 \ X_2 = 1 | Y = 0)$$

Ex 1. Naïve Bayes Classifier (**MLE**)

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

Training

$\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 1), \hat{P}(Y = 1), \hat{P}(Y = 0)$ Use **MLE** or Laplace (MAP)

Testing

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

Training: Naïve Bayes for TV shows (MLE)

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

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

Predict Y : “likes Pokémon”

| $Y \backslash X_1$ | X_1 | | $Y \backslash X_2$ | X_2 | |
|--------------------|-------|----|--------------------|-------|----|
| | 0 | 1 | | 0 | 1 |
| 0 | 3 | 10 | 0 | 5 | 8 |
| 1 | 4 | 13 | 1 | 7 | 10 |

Training data counts

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

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



Training: Naïve Bayes for TV shows (MLE)

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

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

Predict Y : “likes Pokémon”

| $Y \backslash X_1$ | X_1 | | $Y \backslash X_2$ | X_2 | |
|--------------------|-------|----|--------------------|-------|----|
| | 0 | 1 | | 0 | 1 |
| 0 | 3 | 10 | 0 | 5 | 8 |
| 1 | 4 | 13 | 1 | 7 | 10 |

Training data counts

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

| $Y \backslash X_1$ | X_1 | |
|--------------------|-------|---|
| | 0 | 1 |
| 0 | | |
| 1 | | |

Training: Naïve Bayes for TV shows (MLE)

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

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

Predict Y : “likes Pokémon”

| | | X_1 | | X_2 | | Y | |
|-----|---|-------|----|-------|----|-----|----|
| | | 0 | 1 | 0 | 1 | | |
| Y | 0 | 3 | 10 | 5 | 8 | 0 | 13 |
| | 1 | 4 | 13 | 7 | 10 | 1 | 17 |

Training data counts

| X_1 | | X_2 | | Y | | | |
|-------|---|-------|------|---------------------|----------------------|---|----------------------|
| | | | | | | 0 | 1 |
| Y | 0 | 0.23 | 0.77 | $5/13 \approx 0.38$ | $8/13 \approx 0.62$ | 0 | $13/30 \approx 0.43$ |
| | 1 | 0.24 | 0.76 | $7/17 \approx 0.41$ | $10/17 \approx 0.59$ | 1 | $17/30 \approx 0.57$ |

(from last slide)

Training : Naïve Bayes for TV shows (MLE)

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

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

Predict Y : “likes Pokémon”

| $Y \backslash X_1$ | 0 | 1 | $Y \backslash X_2$ | 0 | 1 | Y | |
|--------------------|------|------|--------------------|------|------|-----|------|
| 0 | 0.23 | 0.77 | 0 | 0.38 | 0.62 | 0 | 0.43 |
| 1 | 0.24 | 0.76 | 1 | 0.41 | 0.59 | 1 | 0.57 |

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

Ex 1. Naïve Bayes Classifier (**MLE**)

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

Training

$\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 1), \hat{P}(Y = 1), \hat{P}(Y = 0)$ Use **MLE** or Laplace (MAP)

Testing

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

Testing: Naïve Bayes for TV shows (MLE)

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

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

Predict Y : “likes Pokémon”

| $Y \backslash X_1$ | 0 | 1 | $Y \backslash X_2$ | 0 | 1 | Y | |
|--------------------|------|------|--------------------|------|------|-----|------|
| 0 | 0.23 | 0.77 | 0 | 0.38 | 0.62 | 0 | 0.43 |
| 1 | 0.24 | 0.76 | 1 | 0.41 | 0.59 | 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 :

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

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

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

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

Ex 2. Naïve Bayes Classifier (MAP)

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

Training

$\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 1), \hat{P}(Y = 1), \hat{P}(Y = 0)$ Use MLE or **Laplace (MAP)**

Testing

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

(note the same as before)

Training: Naïve Bayes for TV shows (MAP)

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

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

Predict Y : “likes Pokémon”

| $Y \backslash X_j$ | X_1 | | X_2 | |
|--------------------|-------|----|-------|----|
| | 0 | 1 | 0 | 1 |
| 0 | 3 | 10 | 5 | 8 |
| 1 | 4 | 13 | 7 | 10 |

Training data counts

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

- A. $\frac{\#(X_j=x, Y=y)}{\#(Y=y)}$
- B. $\frac{\#(X_j=x, Y=y)+1}{\#(Y=y)+2}$
- C. $\frac{\#(X_j=x, Y=y)+1}{\#(Y=y)+4}$
- D. other

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



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. $\mathbf{X} = (X_1, X_2)$:

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

Predict Y : “likes Pokémon”

| | | X_1 | | X_2 | | Y | |
|-----|---|-------|----|-------|----|-----|----|
| | | 0 | 1 | 0 | 1 | | |
| Y | 0 | 3 | 10 | 5 | 8 | 0 | 13 |
| | 1 | 4 | 13 | 7 | 10 | 1 | 17 |

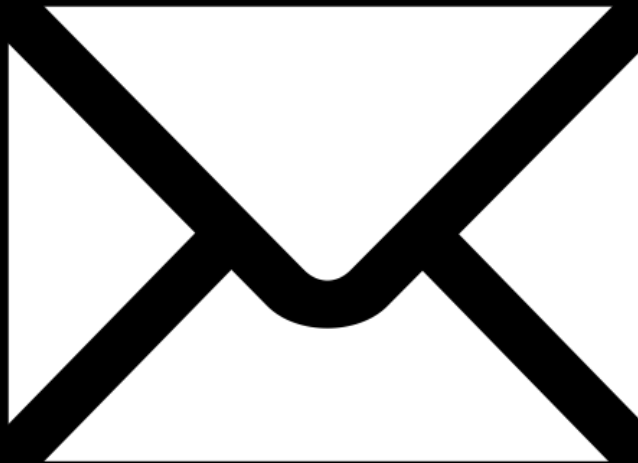
Training data counts

| | | X_1 | |
|-----|---|-------|------|
| | | 0 | 1 |
| Y | 0 | 0.27 | 0.73 |
| | 1 | 0.26 | 0.74 |

| | | X_2 | |
|-----|---|-------|------|
| | | 0 | 1 |
| Y | 0 | 0.40 | 0.60 |
| | 1 | 0.42 | 0.58 |

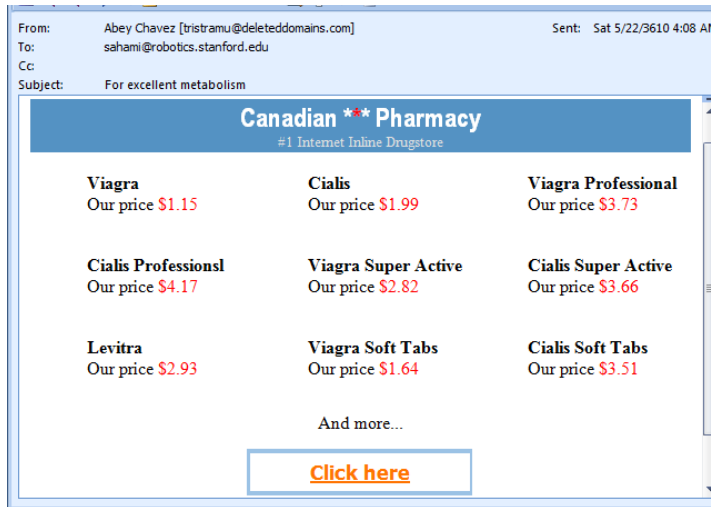
In practice:

- We use Laplace for $\hat{P}(X_j|Y)$ in case some events $X_j = x_j$ don't appear
- We don't use Laplace for $\hat{P}(Y)$, because all class labels should appear reasonably often



and Learn
naively

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 blacklist [URIs: recragas.cn]
- 5.0 URIBL_JP_SURBL
Contains an URL listed in the JP SURBL blacklist [URIs: recragas.cn]
- 5.0 URIBL_OB_SURBL
Contains an URL listed in the OB SURBL blacklist [URIs: recragas.cn]
- 5.0 URIBL_SC_SURBL
Contains an URL listed in the SC SURBL blacklist [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 junk. Lastly, junk mail not only wastes user time, but

Ex 3. Naïve Bayes Classifier (m, n large)

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

Training

$\forall i: \hat{P}(X_i|Y)$

What changes are necessary?

$\hat{P}(X_i|Y=0)$, Use MLE or
 $\hat{P}(X_i|Y=1)$, Laplace (MAP)

Testing

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

Email classification

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

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

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

$X_j = 1$ if word j appeared in document

Output $Y = 1$ if email is spam

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

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

Training: Naïve Bayes Email classification

Train set n previous emails $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$
 $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)})$ for each word, whether it appears in email i
 $y^{(i)} = 1$ if spam, 0 if not spam

Note: m is huge.

Which estimator should we use for $\hat{P}(X_j|Y)$?

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



Training: Naïve Bayes Email classification

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

$\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)})$ for each word, whether it appears in email i

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

Note: m is huge.

Which estimator should we use for $\hat{P}(X_j|Y)$?

- 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!

Ex 3. Naïve Bayes Classifier (m, n large)

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

Training

$\forall j$: $\hat{P}(X_j = 1|Y = 0)$, $\hat{P}(X_j = 0|Y = 0)$, Use MLE or
 $\hat{P}(X_j = 1|Y = 1)$, $\hat{P}(X_j = 0|Y = 1)$, **Laplace (MAP)**
 $\hat{P}(Y = 1)$, $\hat{P}(Y = 0)$

Testing

$\hat{Y} = \arg \max_{y=\{0,1\}} \left(\prod_{j=1}^m \right)$ Laplace (MAP) estimates avoid estimating 0 probabilities for events that don't occur in your training data.

Testing: Naïve Bayes Email classification

For a new email:

- Generate $\mathbf{X} = (X_1, X_2, \dots, X_m)$
- Classify as spam or not using Naïve Bayes assumption

Note: m is huge.

Suppose train set size n also huge (many labeled emails).

Can we still use the below prediction?

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

Testing: Naïve Bayes Email classification

For a new email:

- Generate $\mathbf{X} = (X_1, X_2, \dots, X_m)$
- Classify as spam or not using Naïve Bayes assumption

Note: m is huge.

Suppose train set size n also huge (many labeled emails).

Can we still use the below prediction?

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

Will probably lead to underflow!

Ex 3. Naïve Bayes Classifier (m, n large)

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

Training

$$\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 1), \hat{P}(Y = 1), \hat{P}(Y = 0)$$

Use sums of log-probabilities for numerical stability.

Testing

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

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 workflow:

- 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:

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

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

| | 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% |

What are precision and recall?

Accuracy ($\frac{\# \text{ correct}}{\# \text{ total}}$) sometimes just doesn't cut it.

| | | |
|-------------------|--|----------------------------|
| Precision: | Of the emails you predicted as spam, how many are <i>truly</i> spam? | Measure of false positives |
| Recall: | Of the emails that are truly spam, how many did you predict? | Measure of false negatives |

More on Wikipedia (https://en.wikipedia.org/wiki/Precision_and_recall)