24: Naïve Bayes

David Varodayan

March 4, 2020

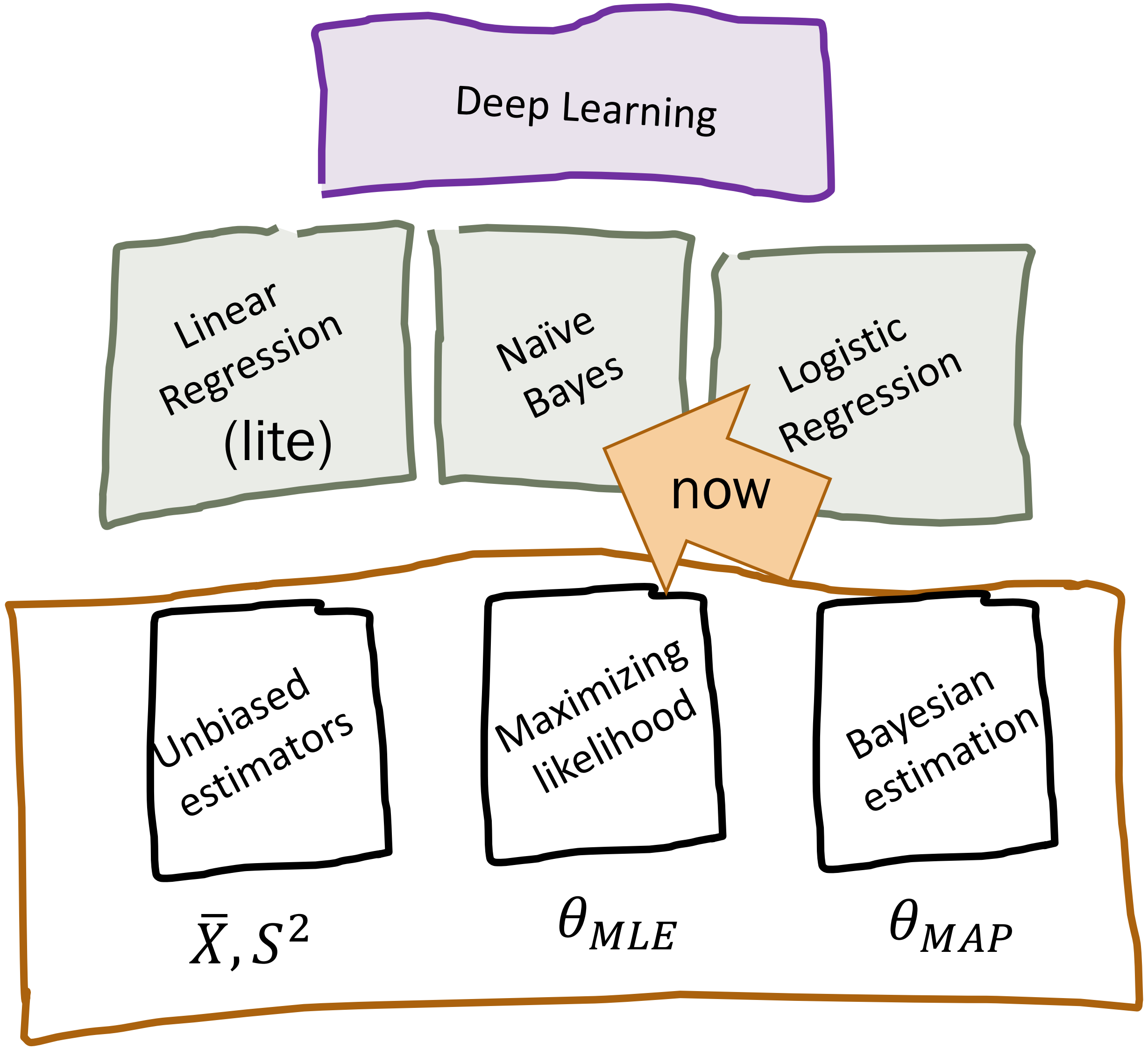
Adapted from slides by Lisa Yan

Today's plan

Machine Learning

- Inefficient classification: Brute force Bayes
- Naïve Bayes

Our path



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

Multinomial MLE and MAP

Model: Multinomial with m outcomes:
p_i probability of outcome i

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

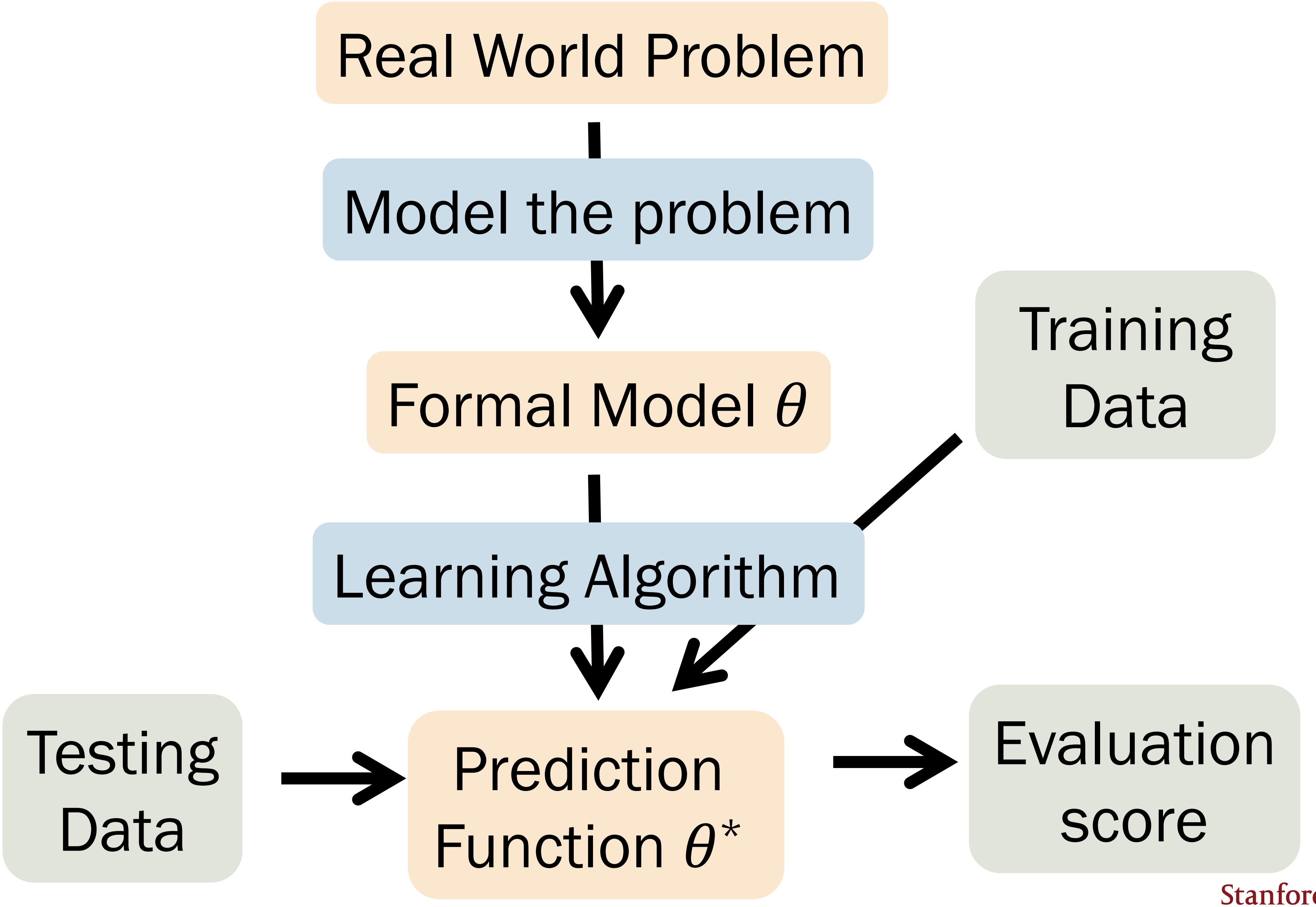
MLE

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

MAP with Laplace smoothing
(Laplace estimate)

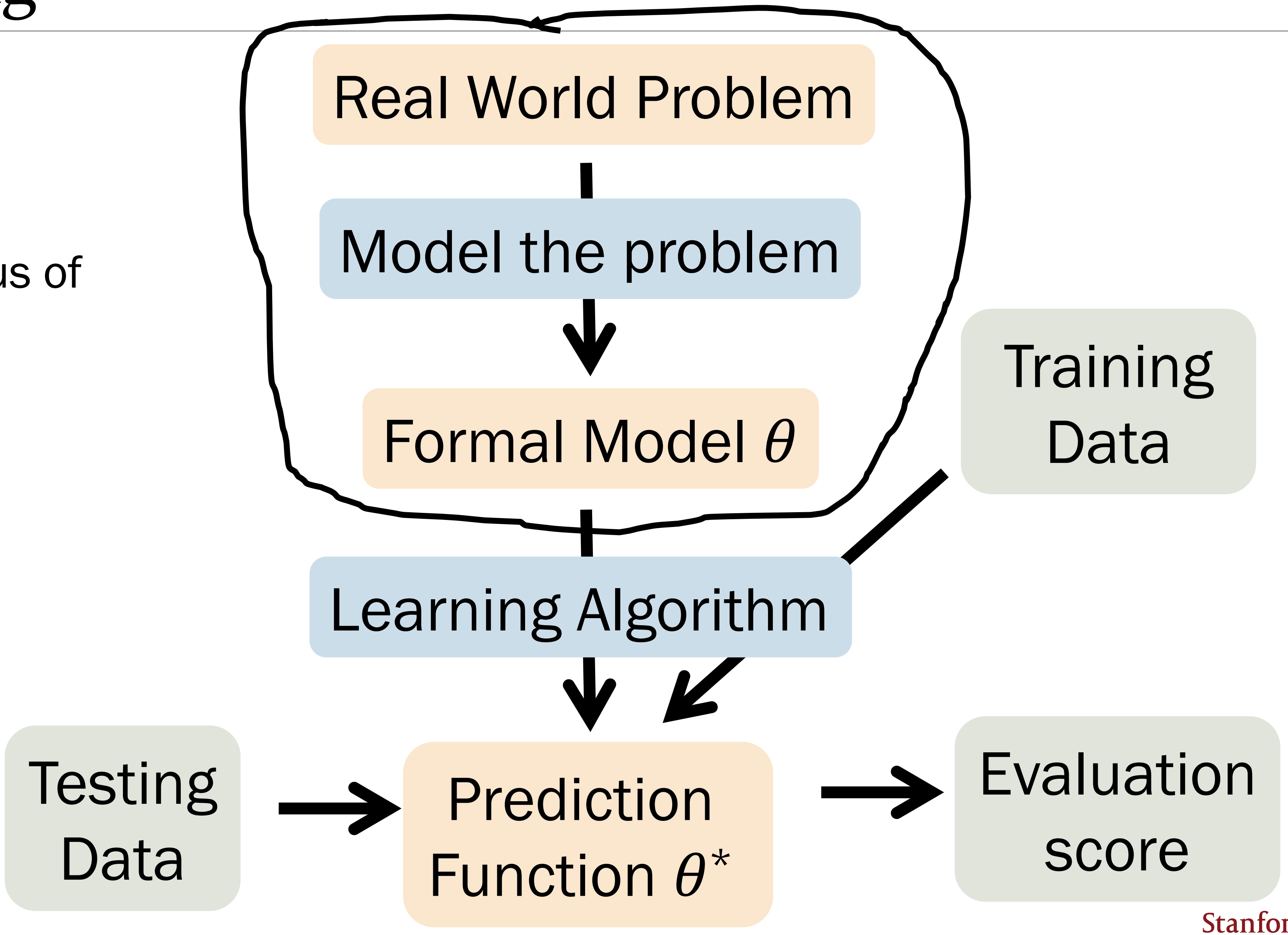
$$p_i = \frac{n_i + 1}{\sum_{i=1}^{m} n_i + m}$$

Supervised Learning

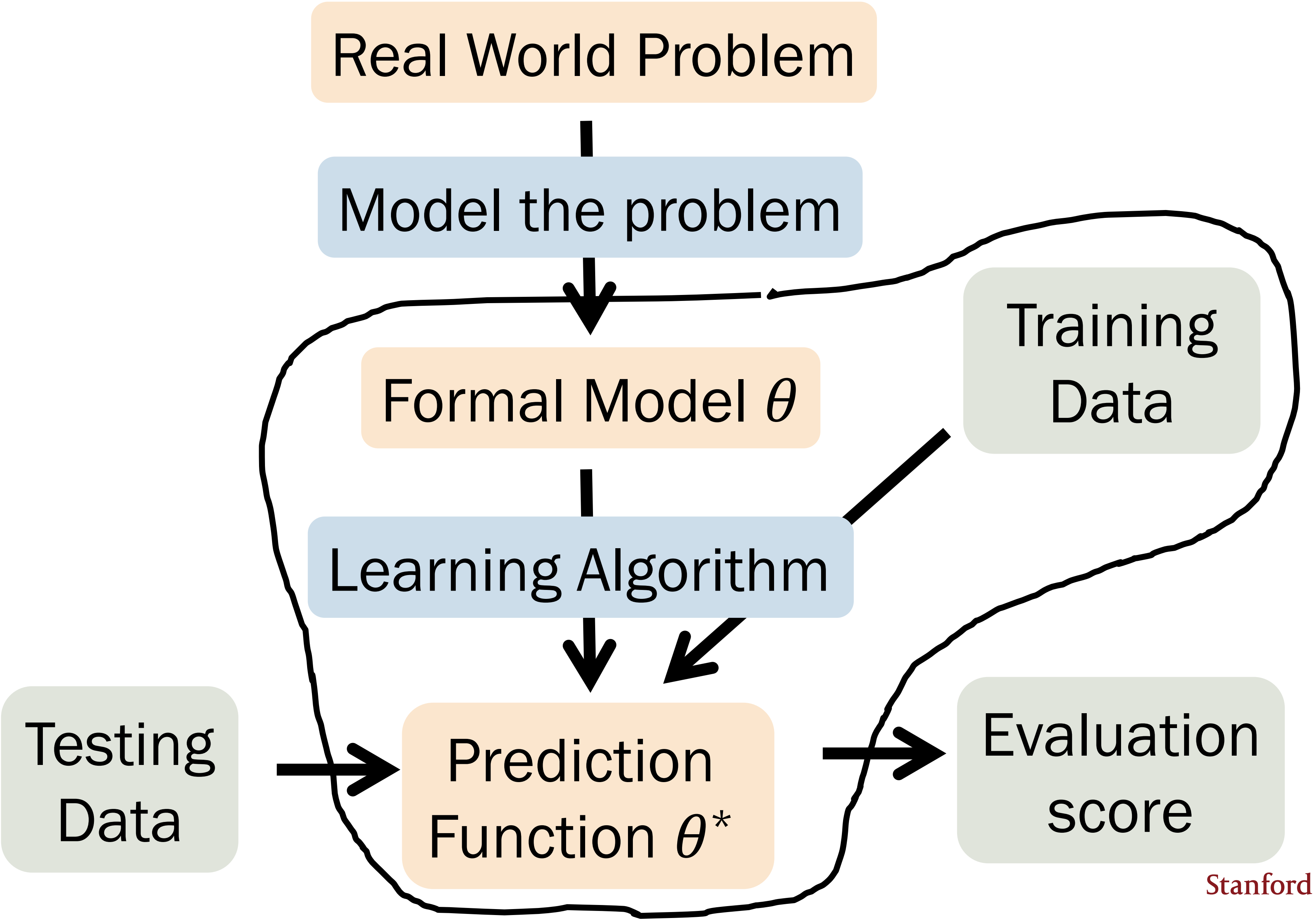


Modeling

(not the focus of this class)

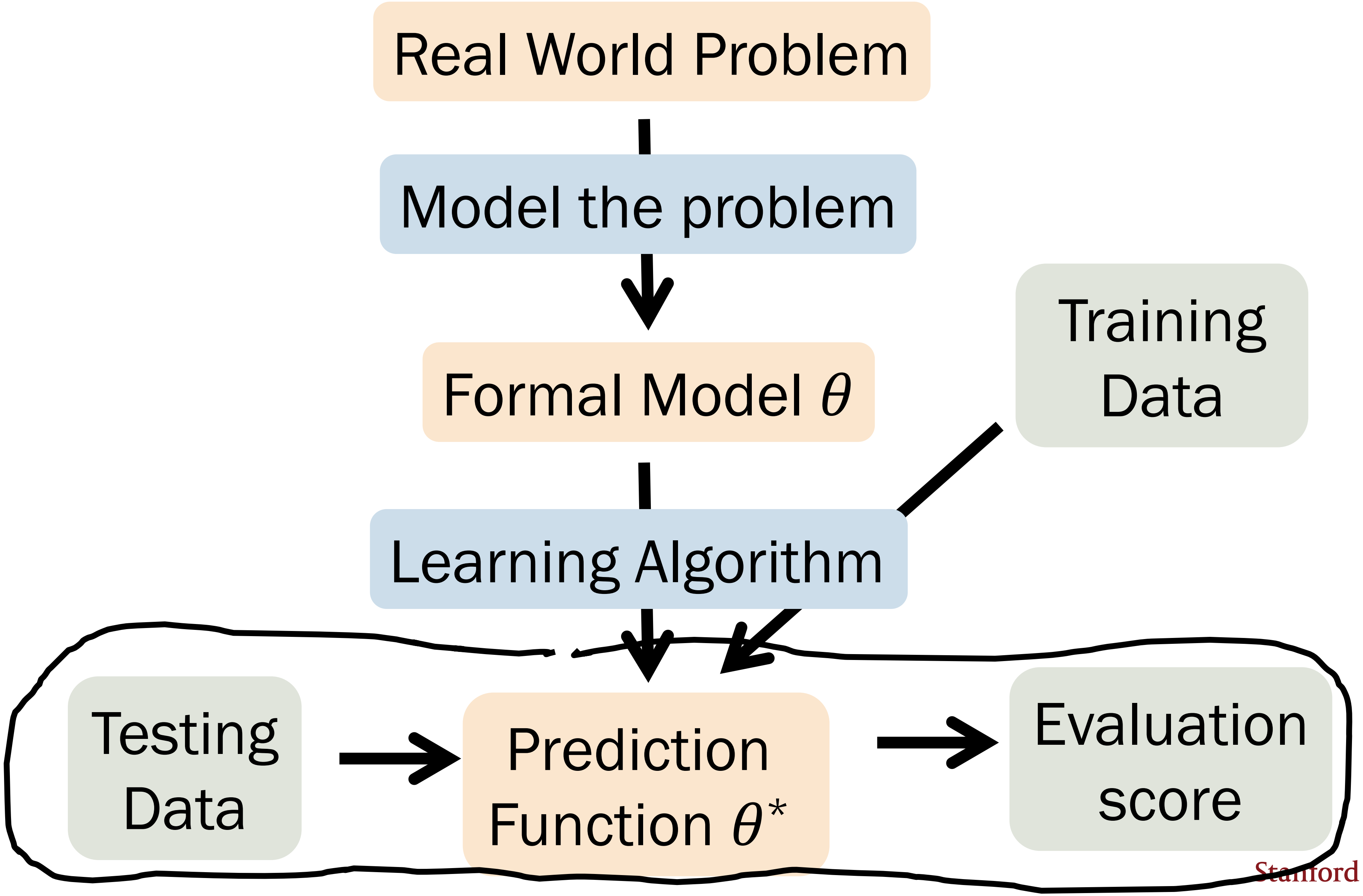


Training



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

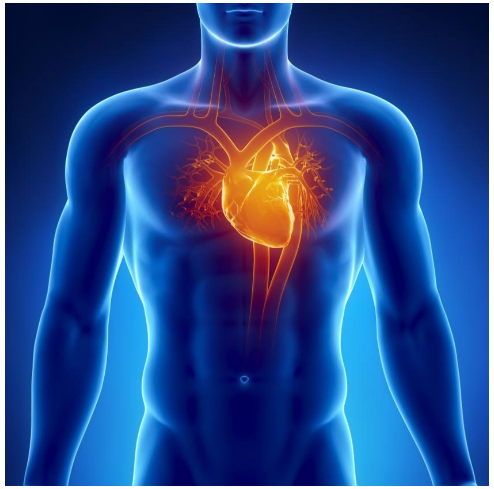
- m features: $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}\right)$
- A single output $y^{(i)}$
- Independent of all other datapoints

Training Goal:

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

Example datasets

Heart



Ancestry

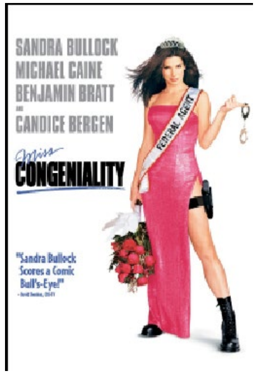
Netflix

Classification terminology check

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

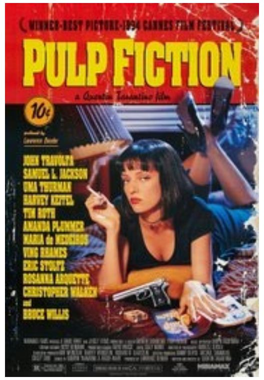
Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$

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

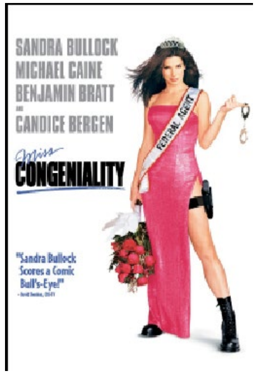
Classification terminology check

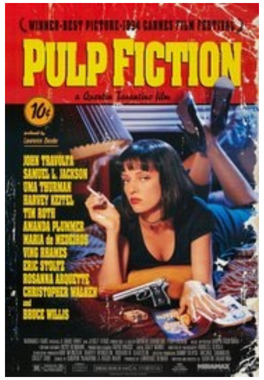
Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$

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

i: i-th user
j: movie j

1: like movie
0: dislike movie



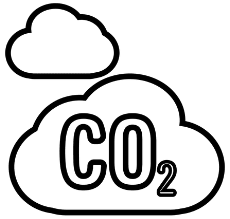


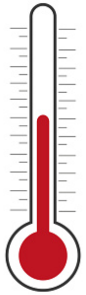
	Movie 1	Movie 2	…	Movie m	Output
User 1 (1.)	1	0	…	1	(2.) 1
User 2 (3.)	1	1	…	0	0
…			⋮		⋮
User n	0	(4.) 0	…	1	1

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

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

$\hat{Y} = 1$

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

Announcements

Problem Set 6

Due: Wednesday 3/11
Covers: Up to Lecture 25
Extra Python Office Hours: Saturday 3/7, 3-5PM

Regrades

Pset 1 to 5 and Midterm regrades to close on 3/11 at 1pm

Autograded Coding Problems

Run your code in the command line or install Pycharm following directions on Pset 6 webpage

Late Day Reminder

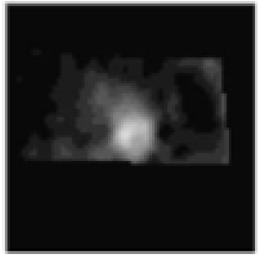
No late days permitted past last day of the quarter, 3/13

Today's plan

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	0

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

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

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

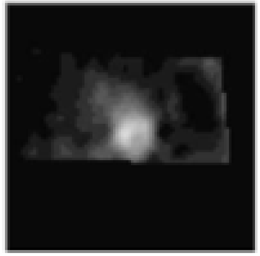
	$\hat{P}(\boldsymbol{X}\|Y=1)$
$X_1 = 0$	θ_3
$X_1 = 1$	θ_4

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



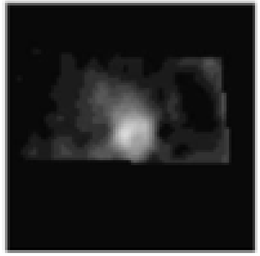
	Feature 1	Output
Patient 1	1	0
Patient 2	1	1
	⋮	⋮
Patient n	0	0

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



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

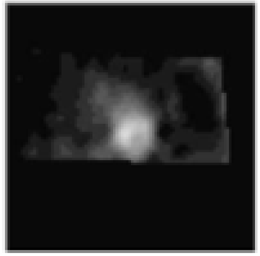
MLE

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

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



Feature 1

Output

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

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

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

(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

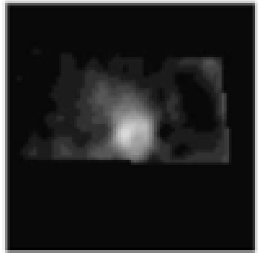
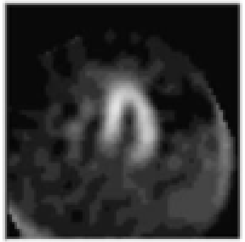
(MAP)	$\hat{P}(Y)$
$Y = 0$	0.10
$Y = 1$	0.90

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

The problem with our Brute force Bayes classifier

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

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

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

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

Estimating this joint conditional distribution will require too many parameters.

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

Today's plan

Machine Learning

- Inefficient classification: Brute force Bayes
- Naïve Bayes

The Naïve Bayes assumption

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

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

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

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

(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 $i = 1, \ldots, m$:

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

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

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)

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)

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

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

Y \ 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)

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

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

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 $\hat{Y} = 1$

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 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 \ 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	0.27	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

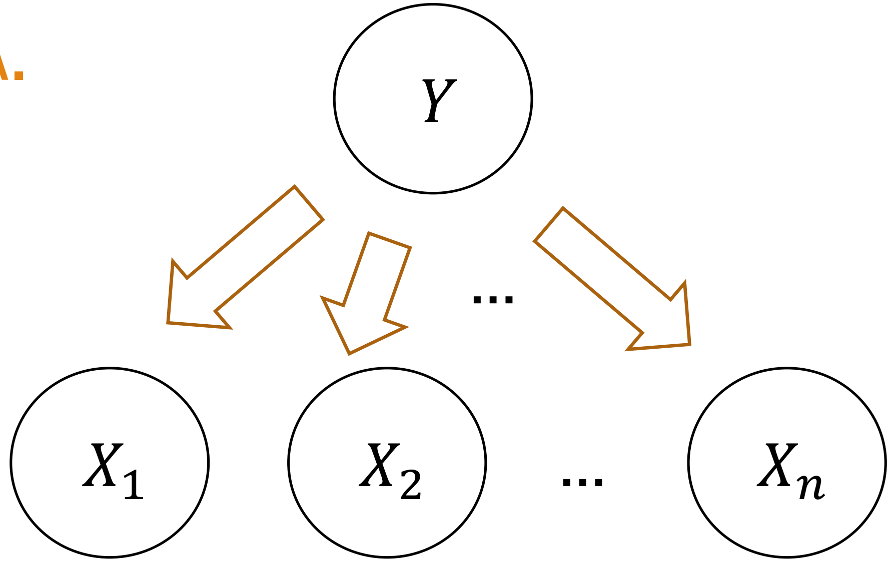
$$\hat{Y} = \arg\max_{y=\{0,1\}} \left(\prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$$

Naïve Bayes Assumption

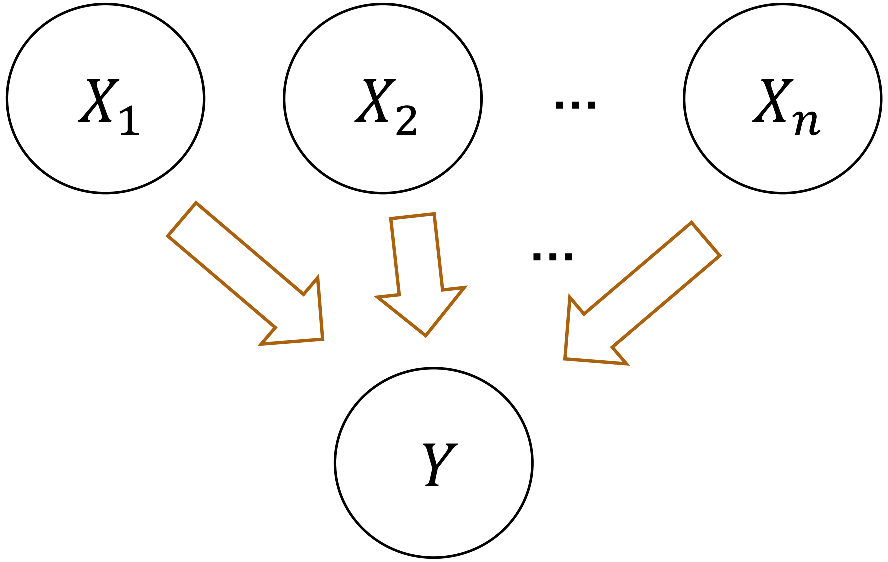
$$P(\mathbf{X}|Y) = \prod_{i=1}^{m} P(X_i|Y) \quad \Rightarrow \quad P(\mathbf{X}, Y) = P(Y) \prod_{i=1}^{m} P(X_i|Y)$$

Which Bayesian Network encodes this conditional independence?

A.



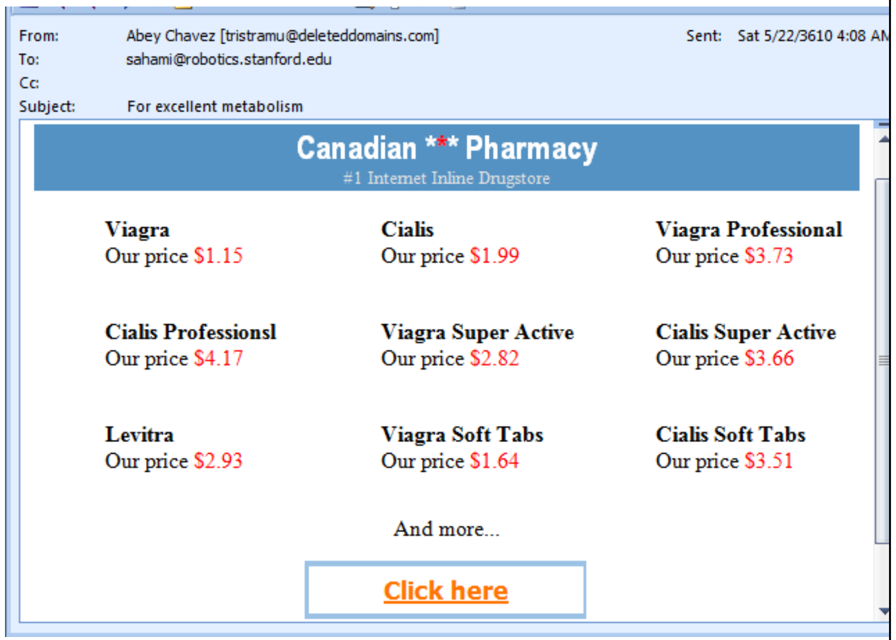
B.



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

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ 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 $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ 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

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

$\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ 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

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

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%