

Naïve Bayes

CS109, Stanford University

Machine Learning in CS109

Great Idea

Neural Networks

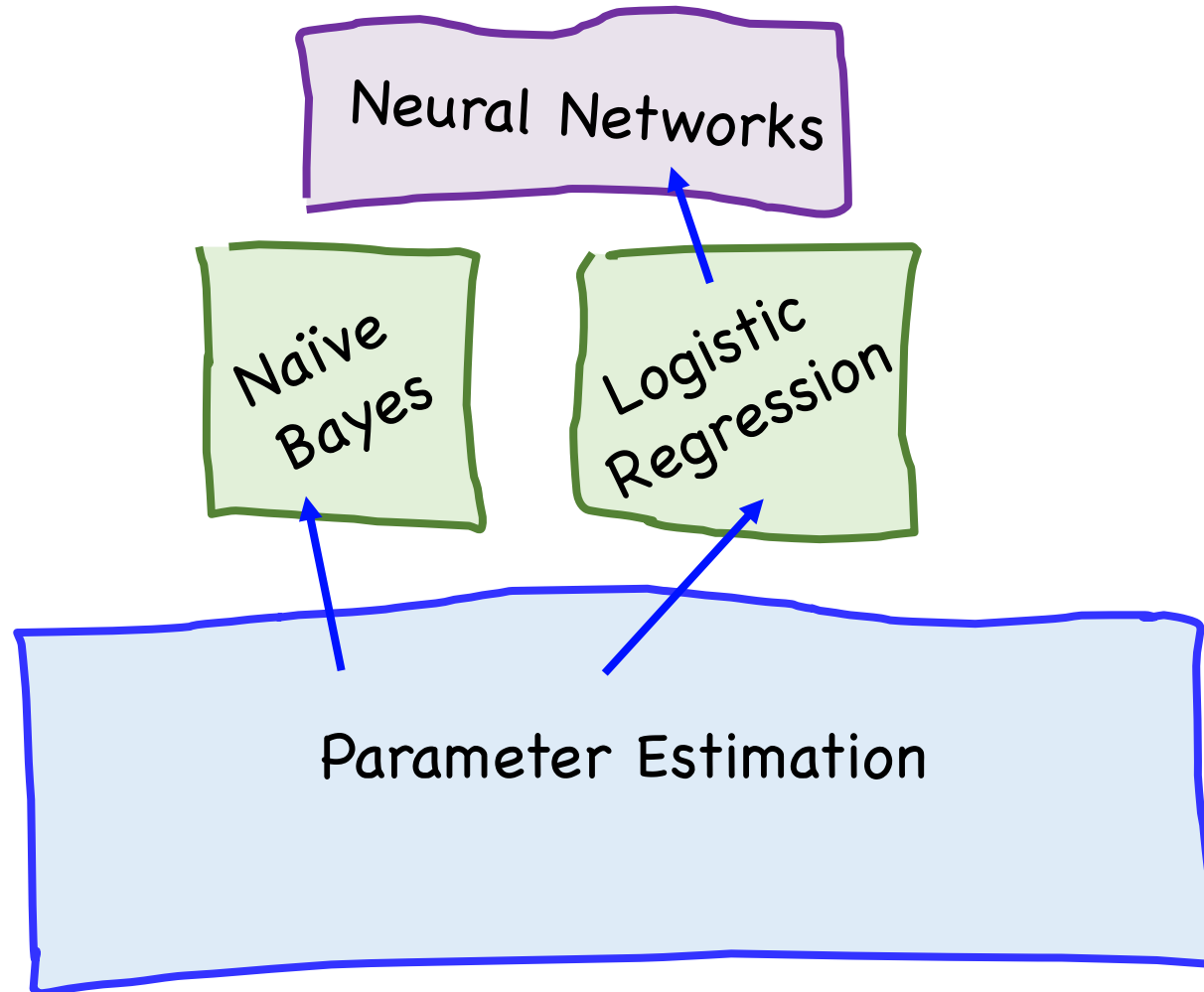
Core Algorithms

Naive Bayes

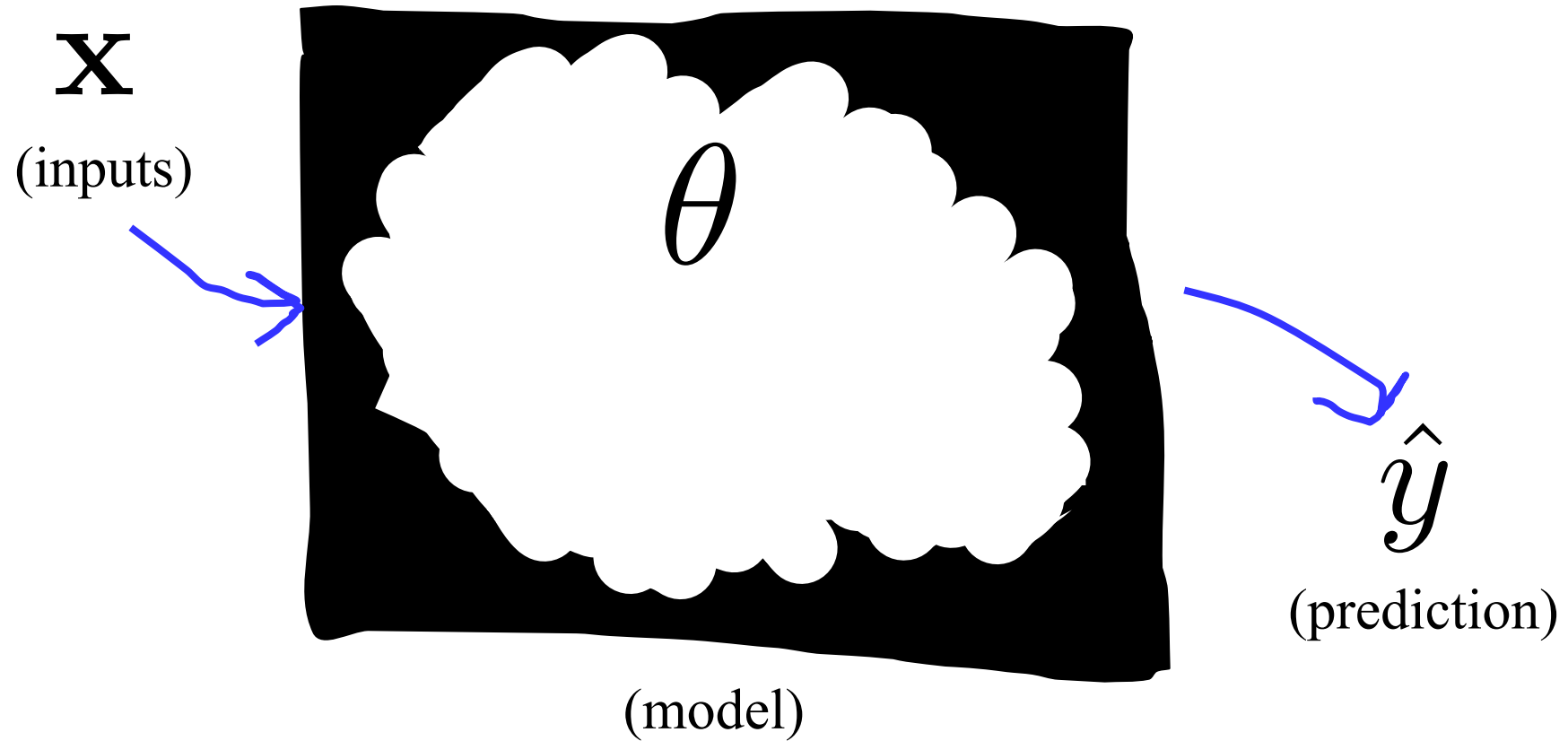
Logistic Regression

Theory

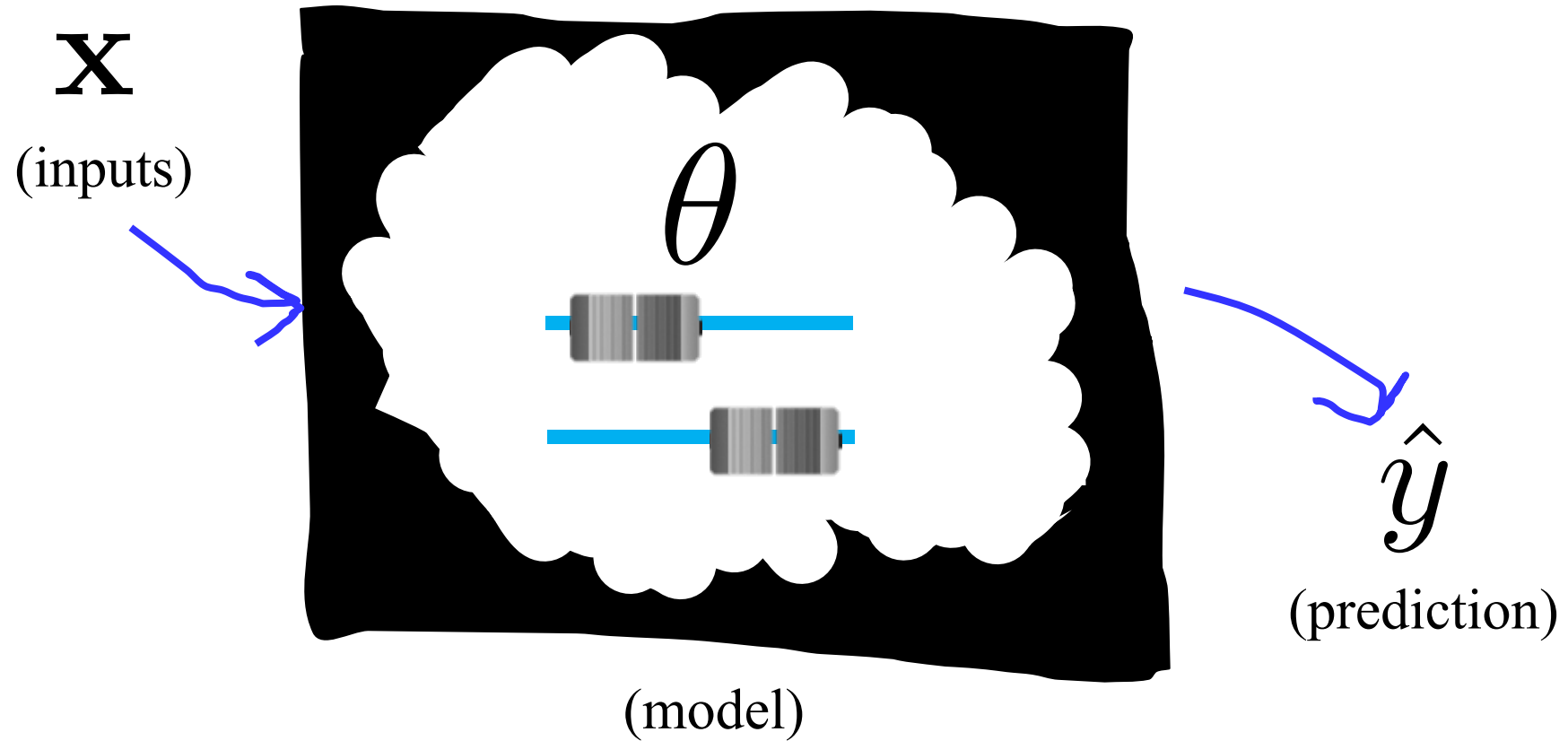
Parameter Estimation



Machine Learning



Machine Learning



MLE vs MAP

Data: $x^{(1)}, \dots, x^{(n)}$

Maximum Likelihood Estimation

$$\begin{aligned}\hat{\theta}_{MLE} &= \operatorname{argmax}_{\theta} f(x^{(1)}, \dots, x^{(n)} | \theta) \\ &= \operatorname{argmax}_{\theta} \left(\sum_i \log f(x^{(i)} | \theta) \right)\end{aligned}$$

Maximum A Posteriori

$$\begin{aligned}\hat{\theta}_{MAP} &= \operatorname{argmax}_{\theta} f(\theta | x^{(1)}, \dots, x^{(n)}) \\ &= \operatorname{argmax}_{\theta} \left(\log(g(\theta)) + \sum_{i=1}^n \log(f(x^{(i)} | \theta)) \right)\end{aligned}$$

Event Shorthand

MAP, without shorthand

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} f(\Theta = \theta | X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)})$$

Our shorthand notation

θ is shorthand for the event: $\Theta = \theta$

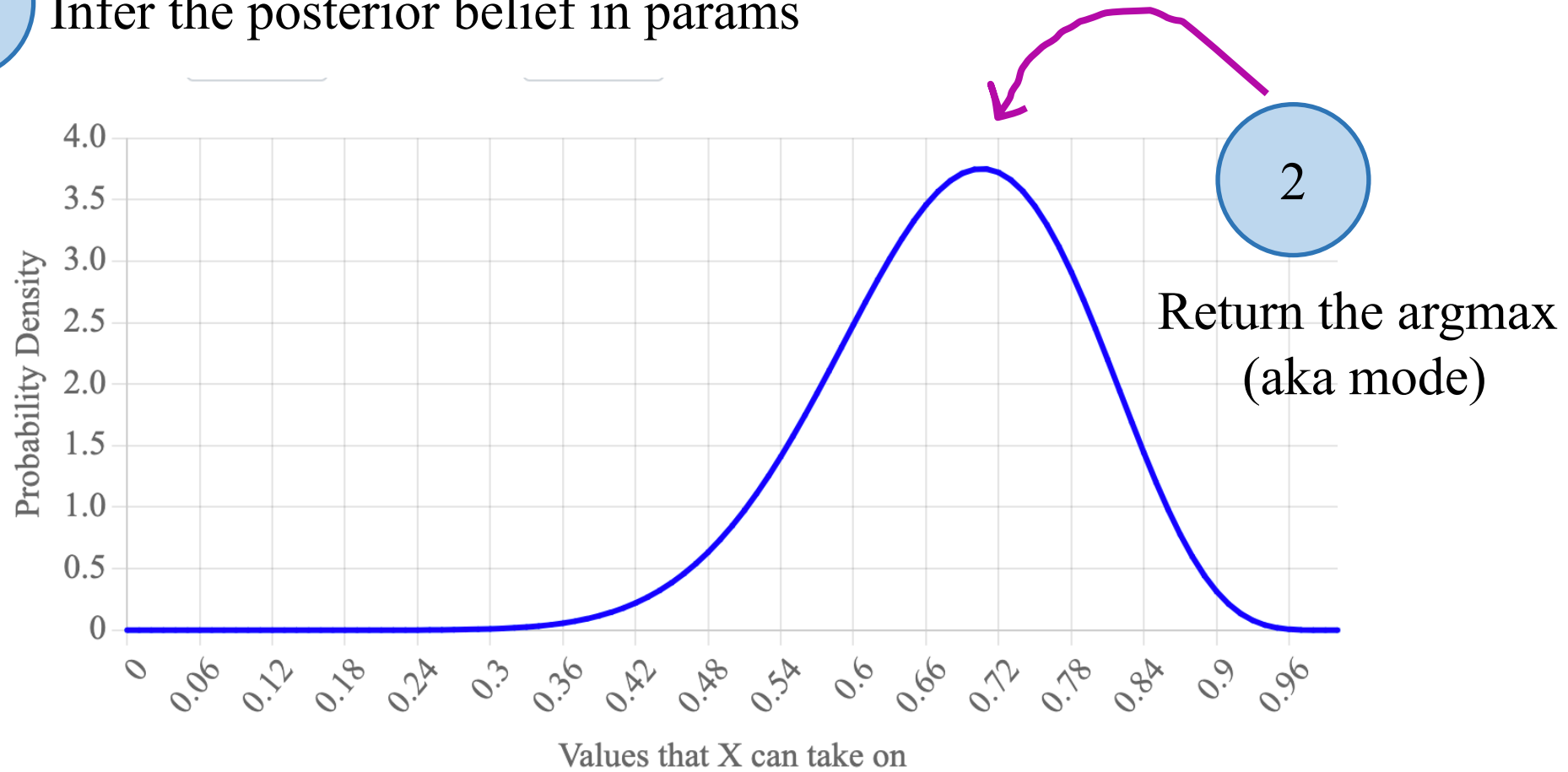
$x^{(i)}$ is shorthand for the event: $X^{(i)} = x^{(i)}$

MAP, now with shorthand

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} f(\theta | x^{(1)}, \dots, x^{(n)})$$

MAP For Bernoulli

1 Infer the posterior belief in params



Beta(a, b) is a conjugate prior for the probability of success in Bernoulli and Binomial distributions.

$$f(x) = \frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1}$$

Prior

Beta(a, b)

Saw $a + b - 2$ imaginary trials: $a - 1$ successes, $b - 1$ failures

Experiment

Observe $n + m$ new trials: n successes, m failures

Posterior

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

MAP:

$$p = \frac{a + n - 1}{a + b + n + m - 2}$$

Quick MAP for Bernoulli with Laplace

Beta(a, b) is a conjugate prior for the probability of success in Bernoulli and Binomial distributions.

$$f(x) = \frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1}$$

Prior

Beta($a = 2, b = 2$)

Saw 2 imaginary trials: 1 successes, 1 failures

Experiment

Observe $n + m$ new trials: n successes, m failures

Posterior

Beta($2 + n, 2 + m$)

MAP:

$$p = \frac{n + 1}{n + m + 2}$$

End Review

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	Conjugate distribution
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 😊

CS109: We'll only focus on MAP for
Bernoulli/Binomial p , Multinomial p_i , and Poisson λ .

Multinomial

Each experiment has M possible outcomes. What is the likelihood of a particular count of each outcome?

*multinomial is parameterized by p_i :
the likelihood of outcome i on any one experiment.*

Dice:

$$M = 6$$

$$p_i = 1/6$$



Multinomial

Each experiment has M possible outcomes. What is the likelihood of a particular count of each outcome?

*multinomial is parameterized by p_i :
the likelihood of outcome i on any one experiment.*

Dice:

$$M = 6$$

$$p_i = 1/6$$



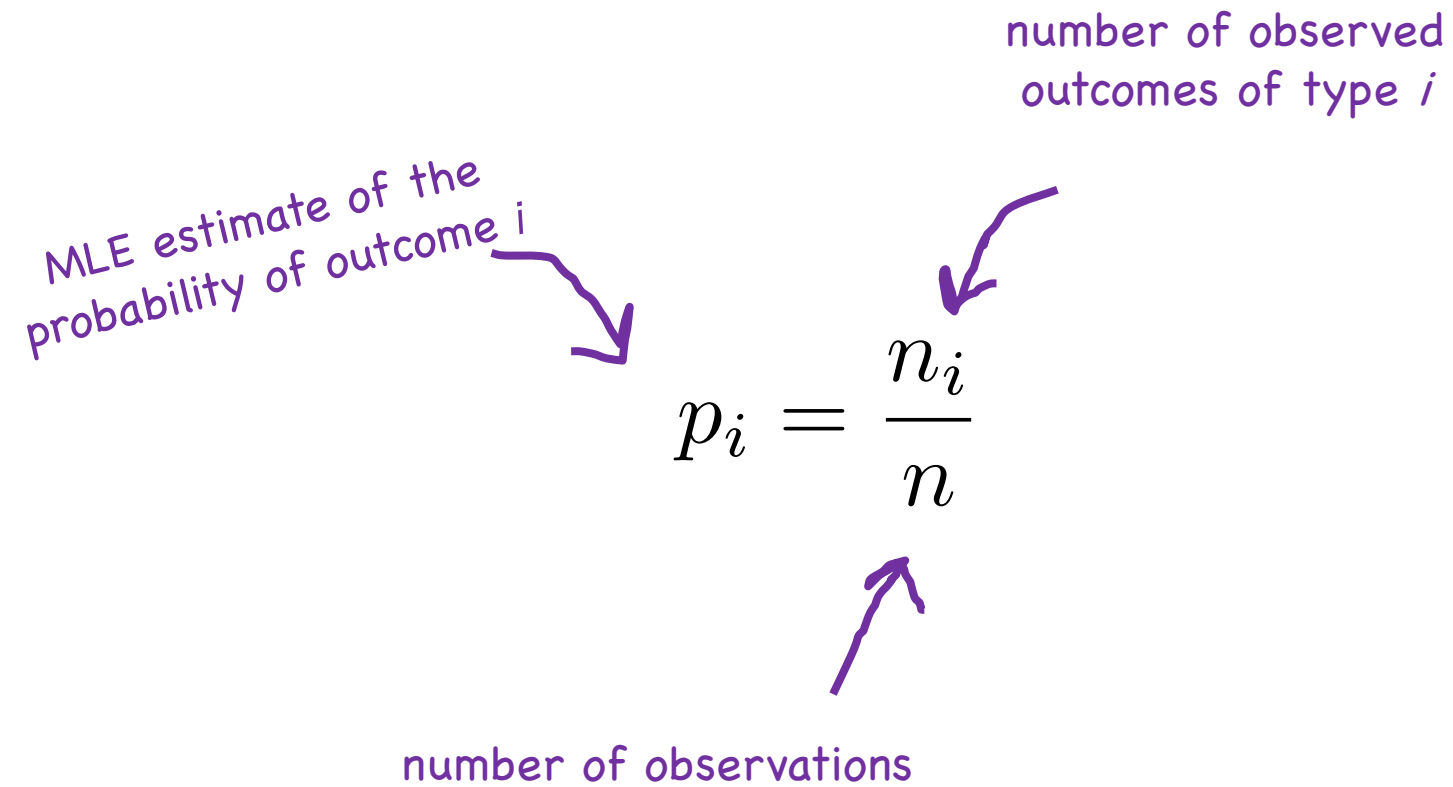
MLE for Multinomial

MLE estimate of the probability of outcome i

number of observed outcomes of type i

$$p_i = \frac{n_i}{n}$$

number of observations



θ_i is p_i
For a multinomial

MAP for Multinomial, Leplace Prior

number of observed outcomes of type i

MAP estimate of the probability of outcome i

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

number of observations

number of outcome types

θ_i is p_i
For a multinomial

Your Happy Laplace

Recall example of 6-sides die rolls:

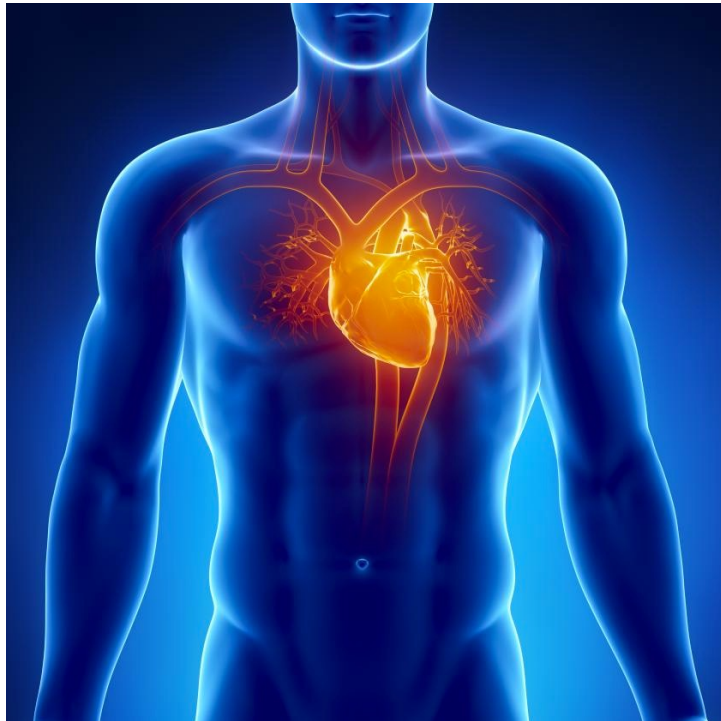
- $X \sim \text{Multinomial}(p_1, p_2, p_3, p_4, p_5, p_6)$
- Roll $n = 12$ times. Result: 3 ones, 2 twos, 0 threes, 3 fours, 1 fives, 3 sixes
 - MLE: $p_1=3/12, p_2=2/12, p_3=0/12, p_4=3/12, p_5=1/12, p_6=3/12$
 - **Laplace estimate:**
$$p_i = \frac{n_i + k}{n + k \cdot m}$$
 - Laplace: $p_1=4/18, p_2=3/18, p_3=1/18, p_4=4/18, p_5=2/18, p_6=4/18$
 - No longer have 0 probability of rolling a three!

Pedagogical Pause

The last estimator has risen...

Example Datasets

Heart



Ancestry



Netflix



Training Data

Training Data: assignments all random variables X and Y

Assume IID data:

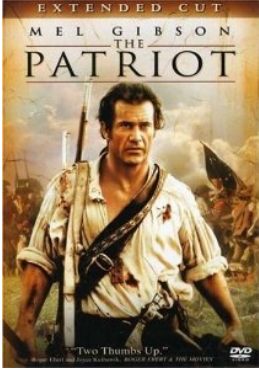

n training datapoints

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

$$m = |\mathbf{x}^{(i)}|$$

Each datapoint has m features and a single output

Target Movie "Like" Classification

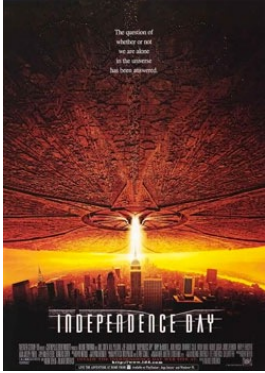
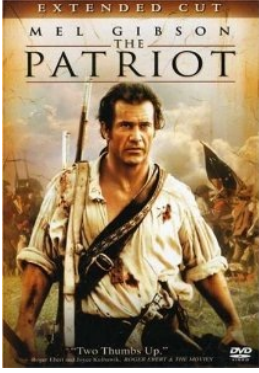
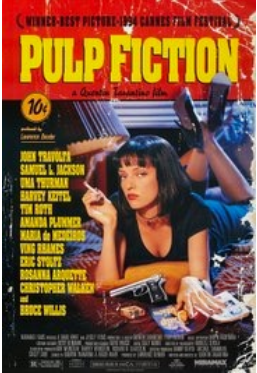

	Movie 1	Movie 2	Movie m	Output
				
User 1	1	0	1	1
User 2	1	1	0	0
		⋮		⋮
User n	0	0	1	1

Single Instance

	Movie 1	Movie 2	Movie m	Output
				
User 1	1	0	1	1
User 2	1	1	0	0
			⋮	⋮
User n	0	0	1	1

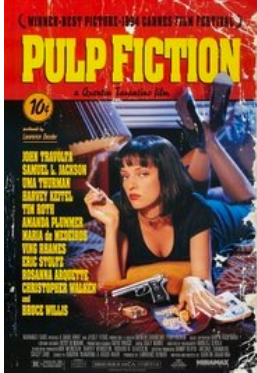
$(\mathbf{x}^{(i)}, y^{(i)})$ such that $1 \leq i \leq n$

Feature Vector

	Movie 1	Movie 2	Movie m	Output
				
User 1	1	0	1	1
User 2	1	1	0	0
		⋮		⋮
User n	0	0	1	1

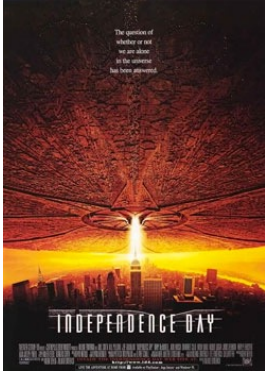
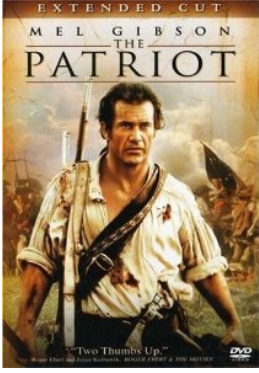
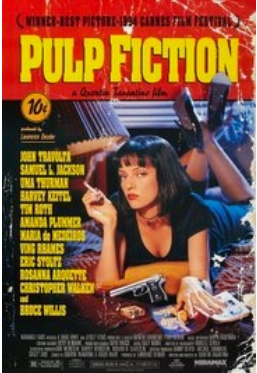
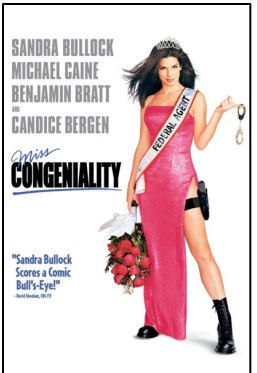
$(\mathbf{x}^{(i)}, y^{(i)})$ such that $1 \leq i \leq n$

Output Value

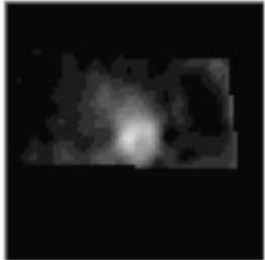
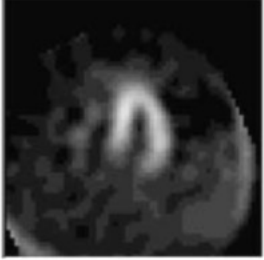
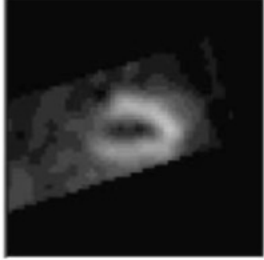

	Movie 1	Movie 2	Movie m	Output
				
User 1	1	0	1	1
User 2	1	1	0	0
		⋮		⋮
User n	0	0	1	1

$(\mathbf{x}^{(i)} \quad y^{(i)})$ such that $1 \leq i \leq n$

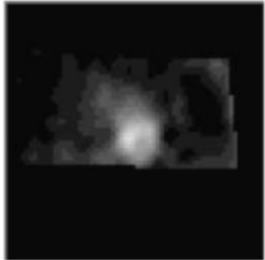
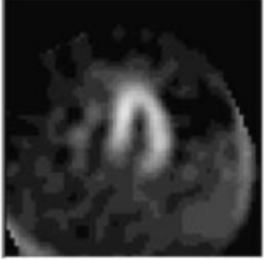
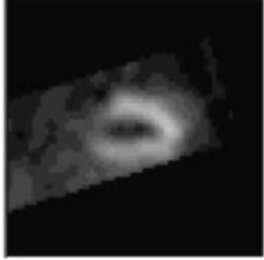

Single Feature Value

	Movie 1	Movie 2	Movie m	Output
				
User 1	1	0	1	1
User 2	1	1	0	0
			⋮	⋮
User n	0	0	1	1
	In general:	$\mathbf{x}_j^{(i)}$	In this case:	$\mathbf{x}_m^{(2)}$

Healthy Heart Classifier

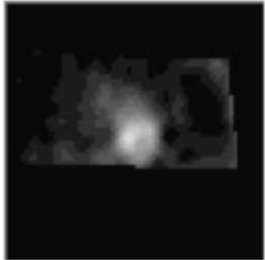
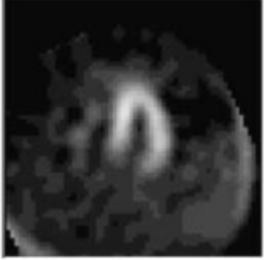
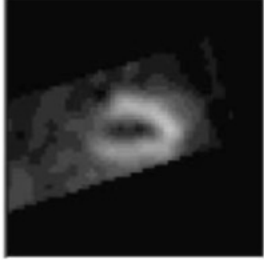

	ROI 1	ROI 2	...	ROI m	Output
			...		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

Healthy Heart Classifier

	ROI 1	ROI 2	ROI m	Output
			... 	
Heart 1	0	1	1	0
Heart 2	1	1	1	0
			⋮	⋮
Heart n	0	0	0	1

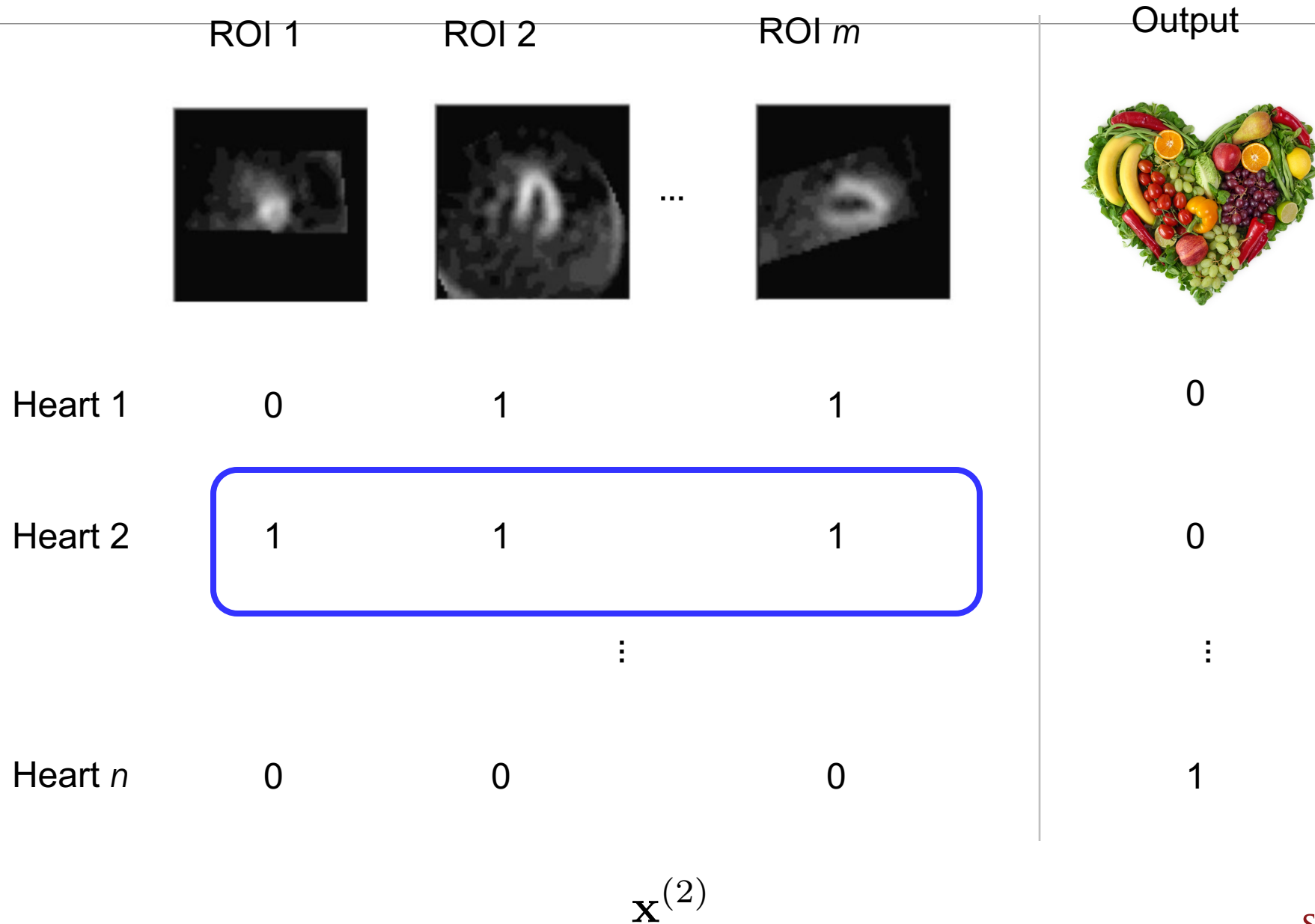
$x_2^{(1)}$

Healthy Heart Classifier

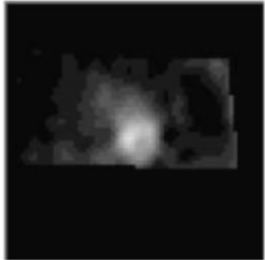
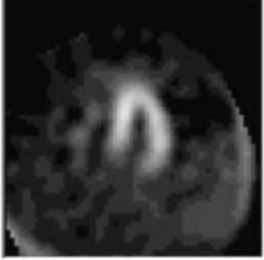
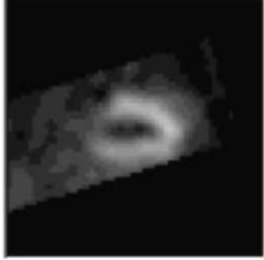

	ROI 1	ROI 2	ROI m	Output
			... 	
Heart 1	0	1	1	0
Heart 2	1	1	1	0
		⋮		⋮
Heart n	0	0	0	1

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

Healthy Heart Classifier

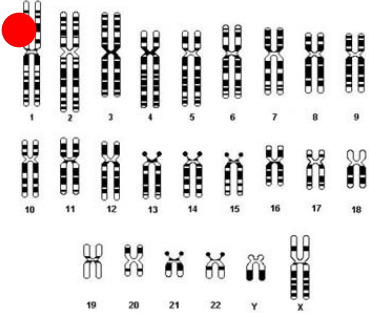
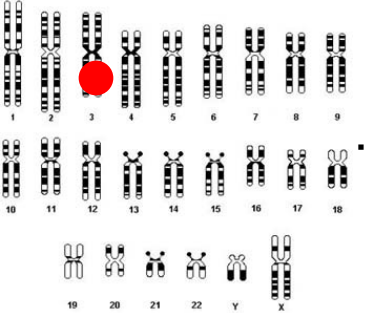
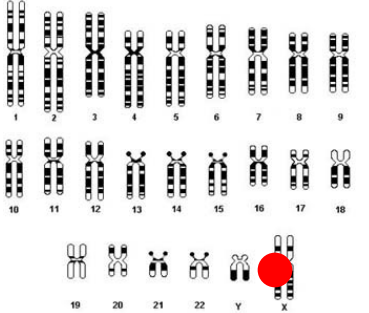



Healthy Heart Classifier





	ROI 1	ROI 2	...	ROI m	Output
			...		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

$y^{(2)}$

Ancestry Classifier

	SNP 1	SNP 2	SNP m	Output
				
User 1	1	0	1	0
User 2	0	0	1	1
...				...
User n	1	1	0	1

Regression: Predicting Real Numbers

	Opposing team ELO	Points in last game	At Home?	Output
				 # Points
Game 1	84	105	1	120
Game 2	90	102	0	95
		⋮		⋮
Game n	74	120	0	115

Training Data

Training Data: assignments all random variables X and Y

Assume IID data:

n training datapoints

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

$$m = |\mathbf{x}^{(i)}|$$

Each datapoint has m features and a single output

ML is ubiquitous

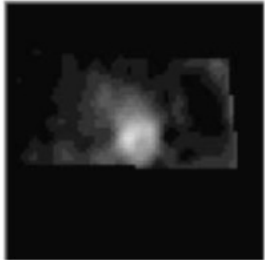
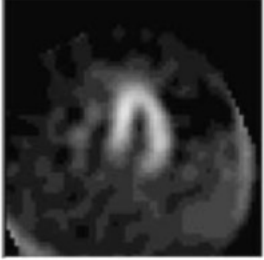
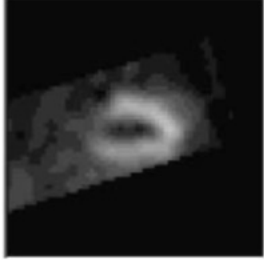

Classification

Classification is Building a Harry Potter Hat

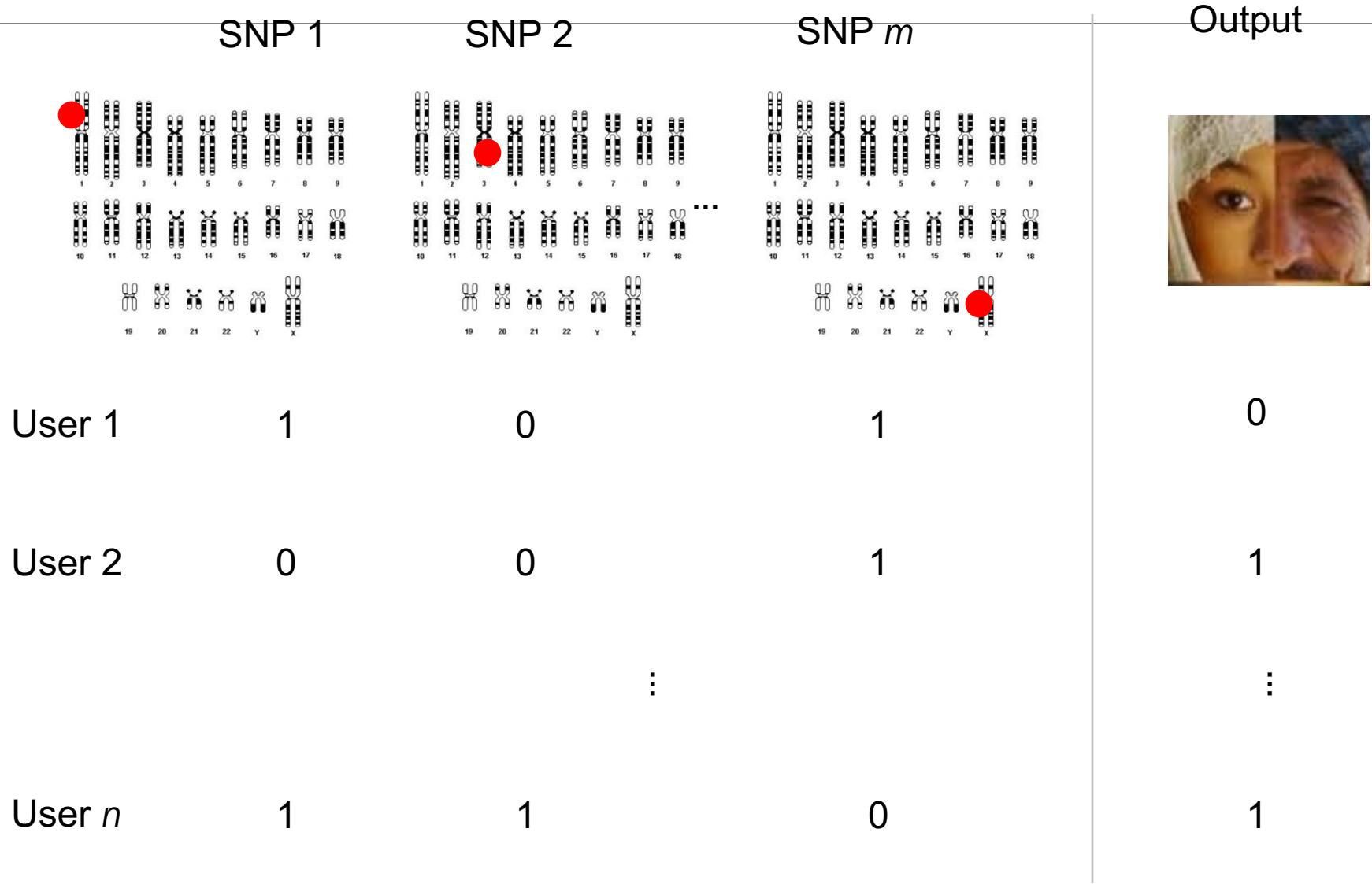


$$\mathbf{x} = [0, 1, \dots, 1]$$

Healthy Heart Classifier

	ROI 1	ROI 2	...	ROI m	Output
			...		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

Ancestry Classifier



NETFLIX

And Learn

Target Movie “Like” Classification

Feature 1



Output



User 1

1

1

User 2

1

0

⋮

User n

0

1

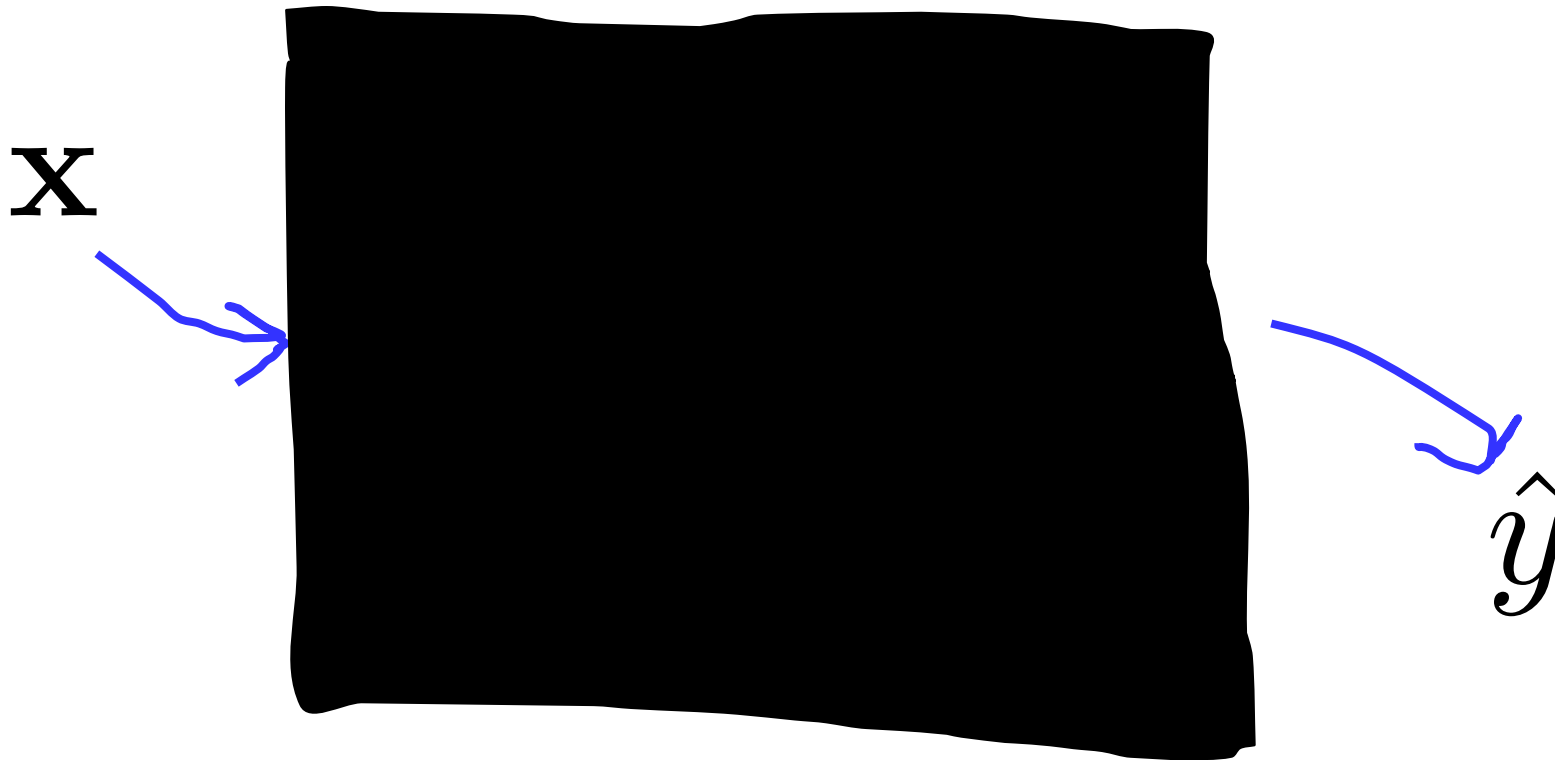
$$x_j^{(i)} \in \{0, 1\}$$

$$y^{(i)} \in \{0, 1\}$$

How could we predict the class label:
will the user like life is beautiful?

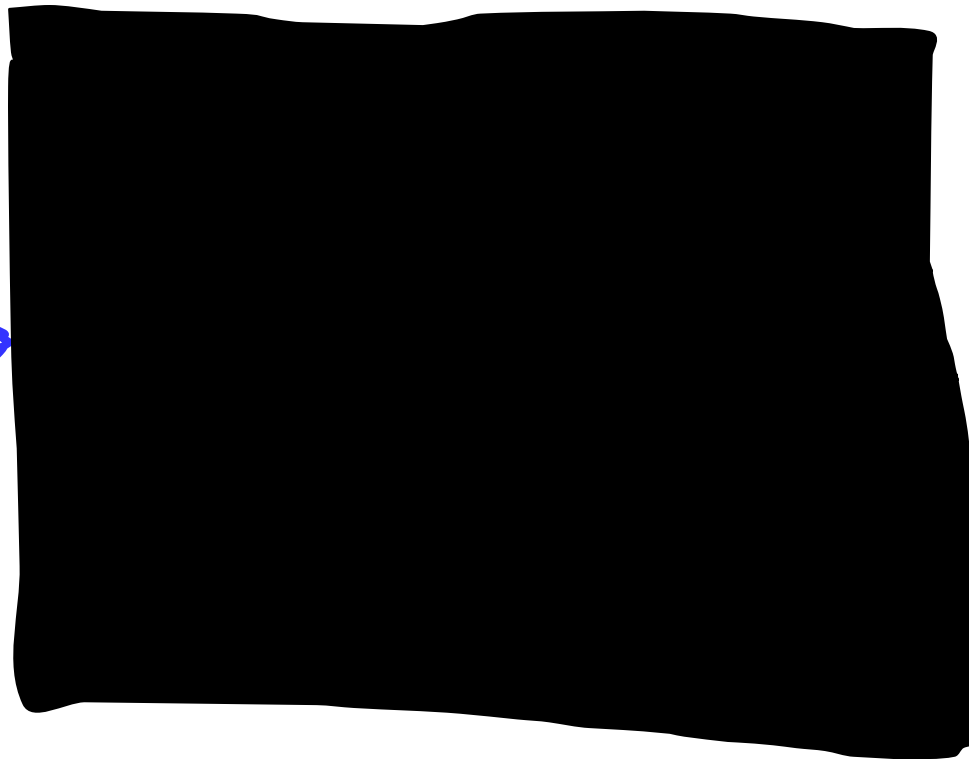
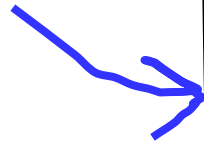
Fake Algorithm: Brute Bayes Classifier

Brute Force Bayes



Brute Force Bayes

\mathbf{x}
[0, 1, 1, 0]

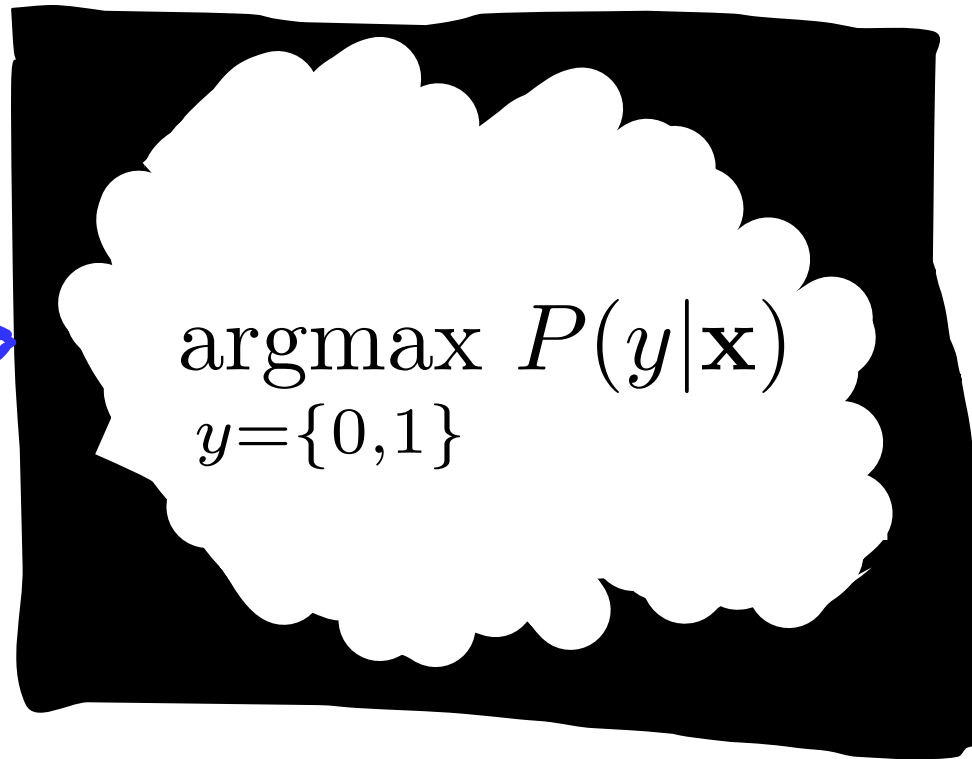


\hat{y}

A blue arrow pointing from the right side of the black box to the predicted output \hat{y} .

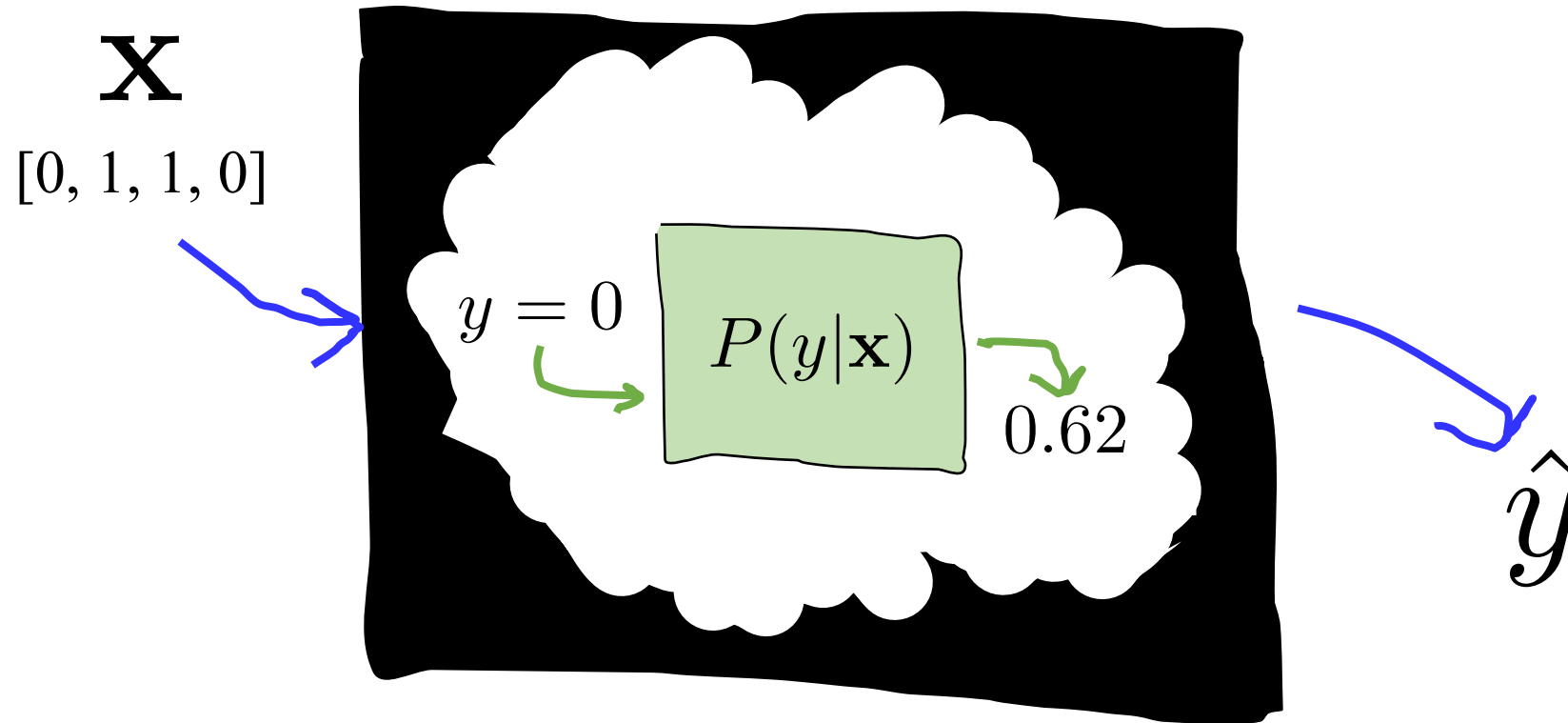
Brute Force Bayes

\mathbf{x}
[0, 1, 1, 0]

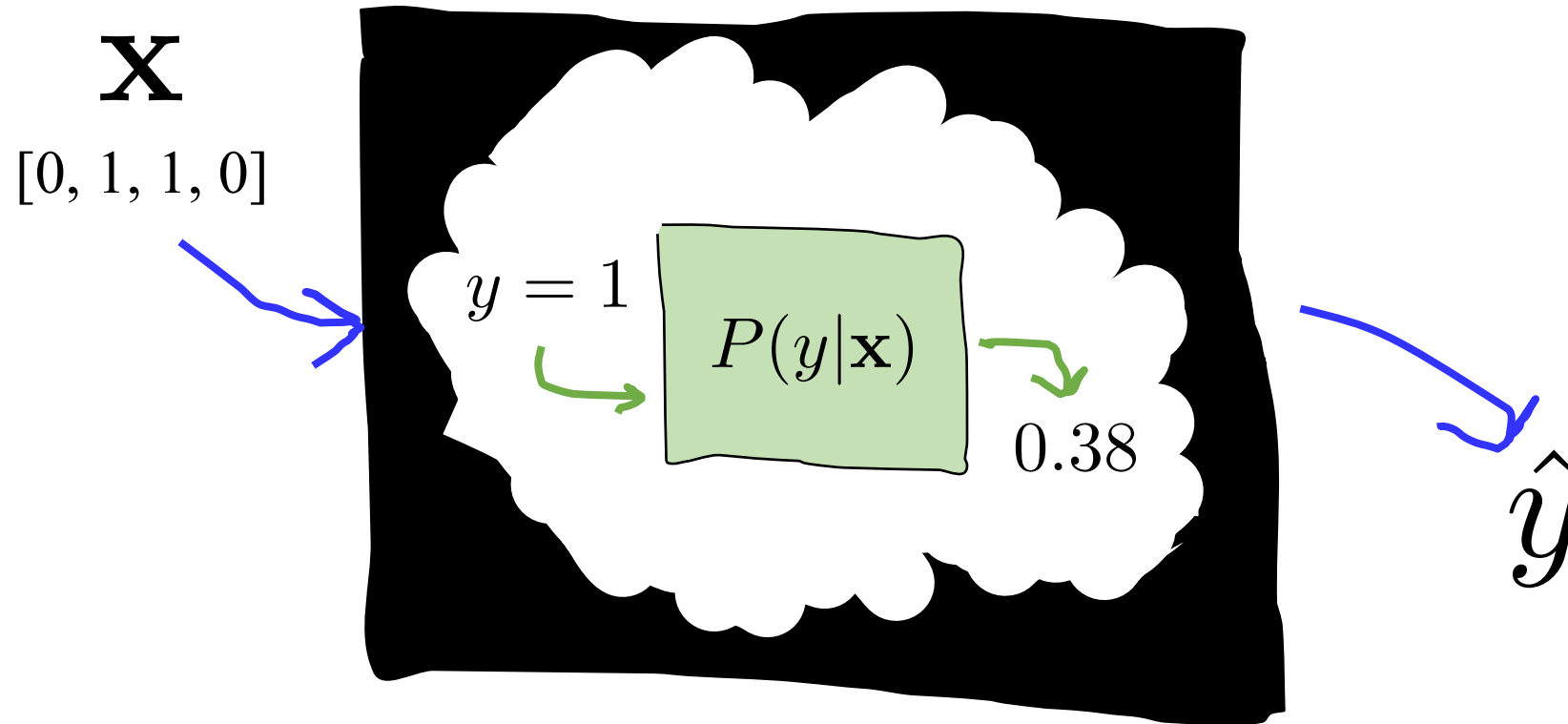


\hat{y}

Brute Force Bayes

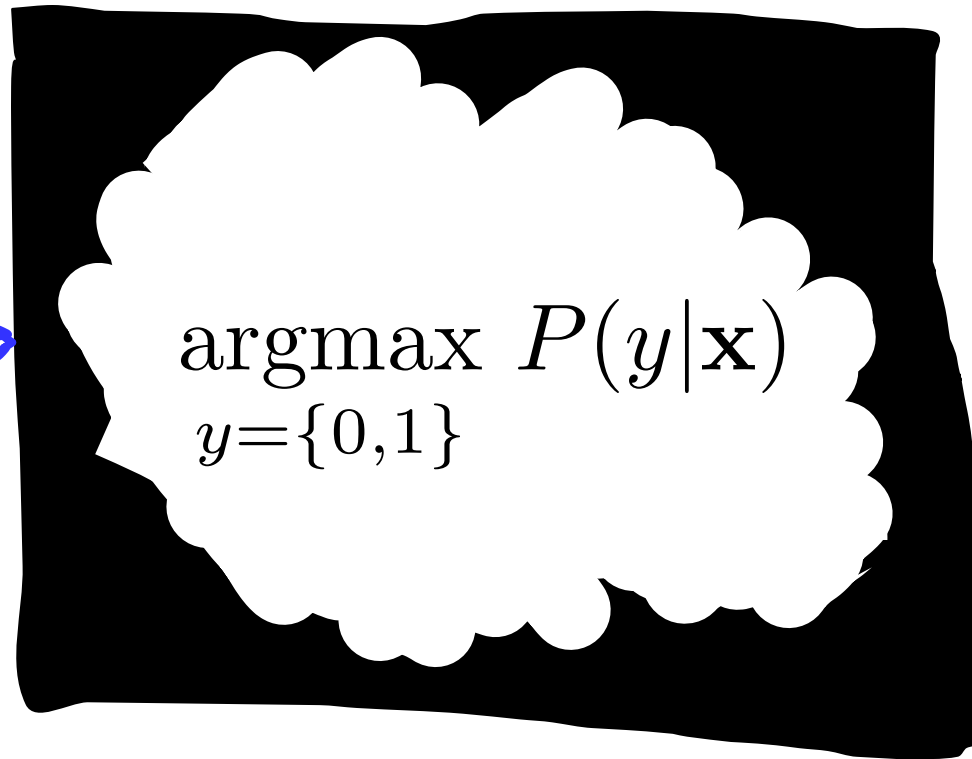


Brute Force Bayes



Brute Force Bayes

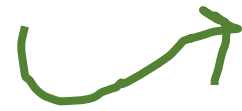
\mathbf{x}
[0, 1, 1, 0]



\hat{y}

Brute Force Bayes

Prediction: will they like
L.I.B.?



$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x})$$

If $y = 1$, they like L.I.B.?



Whether or not they liked
Independence day



Simply chose the class label that is the most likely given the data

This is for one user

Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x})$$

Simply chose the class label that is the most likely given the data

This is for one user

Brute Force Bayes

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

Simply chose the class label that is the most likely given the data

This is for one user

* Note how similar this is to Hamilton example 😊

What are the Parameters?

Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} \underline{P(\mathbf{x}|y)} \underline{P(y)}$$



Conditional probability table



Y = 0

$x_1 = 0$	θ_0
$x_1 = 1$	θ_1

Y = 1

$x_1 = 0$	θ_2
$x_1 = 1$	θ_3



Y = 0	θ_4
Y = 1	θ_5

Learn these during training

Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} \underline{P(\mathbf{x}|y)} \underline{P(y)}$$



Conditional probability table



$x_1 \backslash Y$	0	1
0	θ_0	θ_2
1	θ_1	θ_3

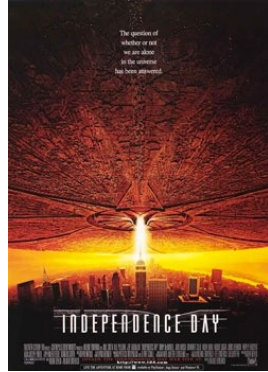


$Y = 0$	θ_4
$Y = 1$	θ_5

Learn these during training

Training

x_1



y



$P(\mathbf{x}|y)$

User 1

1

1

User 2

0

0

⋮

User n

0

1

$x_1 \backslash y$	0	1
0	θ_0	θ_2
1	θ_1	θ_3


What is $P(x_1 | Y = 0)$?

What is $P(x_1 | Y = 1)$?


Both multinomials with two outcomes

MLE Estimate

x_1 y $P(x|y)$



INDEPENDENCE DAY



LIFE IS BEAUTIFUL

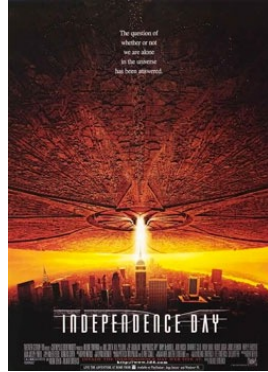
User 1	1	1
User 2	0	0
⋮		
User n	0	1

$x_1 \backslash y$	0	1
0	0.0	0.4
1	1.0	0.6

MLE: Just count

MAP Estimate

x_1



y



$P(\mathbf{x}|y)$

User 1

1

1

User 2

0

0

⋮

User n

0

1

$x_1 \backslash y$	0	1
0	0.01	0.42
1	0.99	0.58

MAP: Just count and add imaginary trials

Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$

$x_1 \backslash Y$	0	1
0	0.01	0.42
1	0.99	0.58

$Y=0$	0.21
$Y=1$	0.79

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0)$$

vs

$$P(x_1 = 1|y = 1)P(y = 1)$$

Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$

$x_1 \backslash Y$	0	1
0	0.01	0.42
1	0.99	0.58

$Y=0$	0.21
$Y=1$	0.79

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0)$$

0.208

vs

$$P(x_1 = 1|y = 1)P(y = 1)$$

Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$

$x_1 \backslash Y$	0	1
0	0.01	0.42
1	0.99	0.58

$Y=0$	0.21
$Y=1$	0.79

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0) \quad 0.208$$

vs

$$P(x_1 = 1|y = 1)P(y = 1) \quad 0.458$$


That was pretty good!

Brute Force Bayes $m = 2$

	x_1	x_2	y
			
User 1	1	0	1
User 2	1	0	0
			⋮
User n	0	1	1

Brute Force Bayes $m = 2$

Simply chose the class label that is the most likely given the data

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

$$P(x_1, x_2|y)$$

Brute Force Bayes

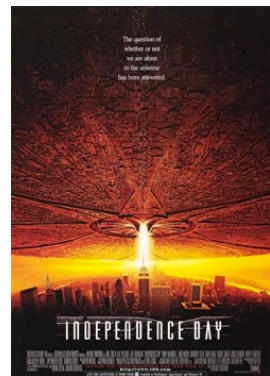
$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$

		Y = 0		Y = 1	
		X ₁ = 0	X ₁ = 1	X ₁ = 0	X ₁ = 1
X ₂	X ₁				
0	0	θ_0	θ_1	θ_4	θ_5
1	0	θ_2	θ_3	θ_6	θ_7
	1				

X₁



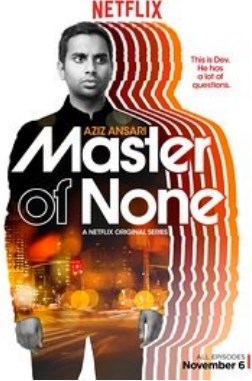

X₂

y




Fine

Brute Force Bayes $m = 3$

	X_1	X_2	X_3	y
				
User 1	1	0	1	1
User 2	1	0	1	0
				⋮
User n	0	1	1	1

Brute Force Bayes $m = 3$

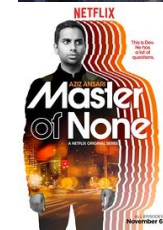
Simply chose the class label that is the most likely given the data

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

$$P(x_1, x_2, x_3|y)$$

Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$

		Y = 0		Y = 1			
		X ₂ \ X ₁	0	1	X ₂ \ X ₁	0	1
X ₃ = 0	0		θ_0	θ_1		θ_8	θ_9
	1		θ_2	θ_3		θ_{10}	θ_{11}
		X ₂ = 0		X ₂ = 1			
		X ₂ \ X ₁	0	1	X ₂ \ X ₁	0	1
X ₃ = 1	0		θ_4	θ_5		θ_{12}	θ_{13}
	1		θ_6	θ_7		θ_{14}	θ_{15}



And if $m=100$?

Brute Force Bayes $m = 100$

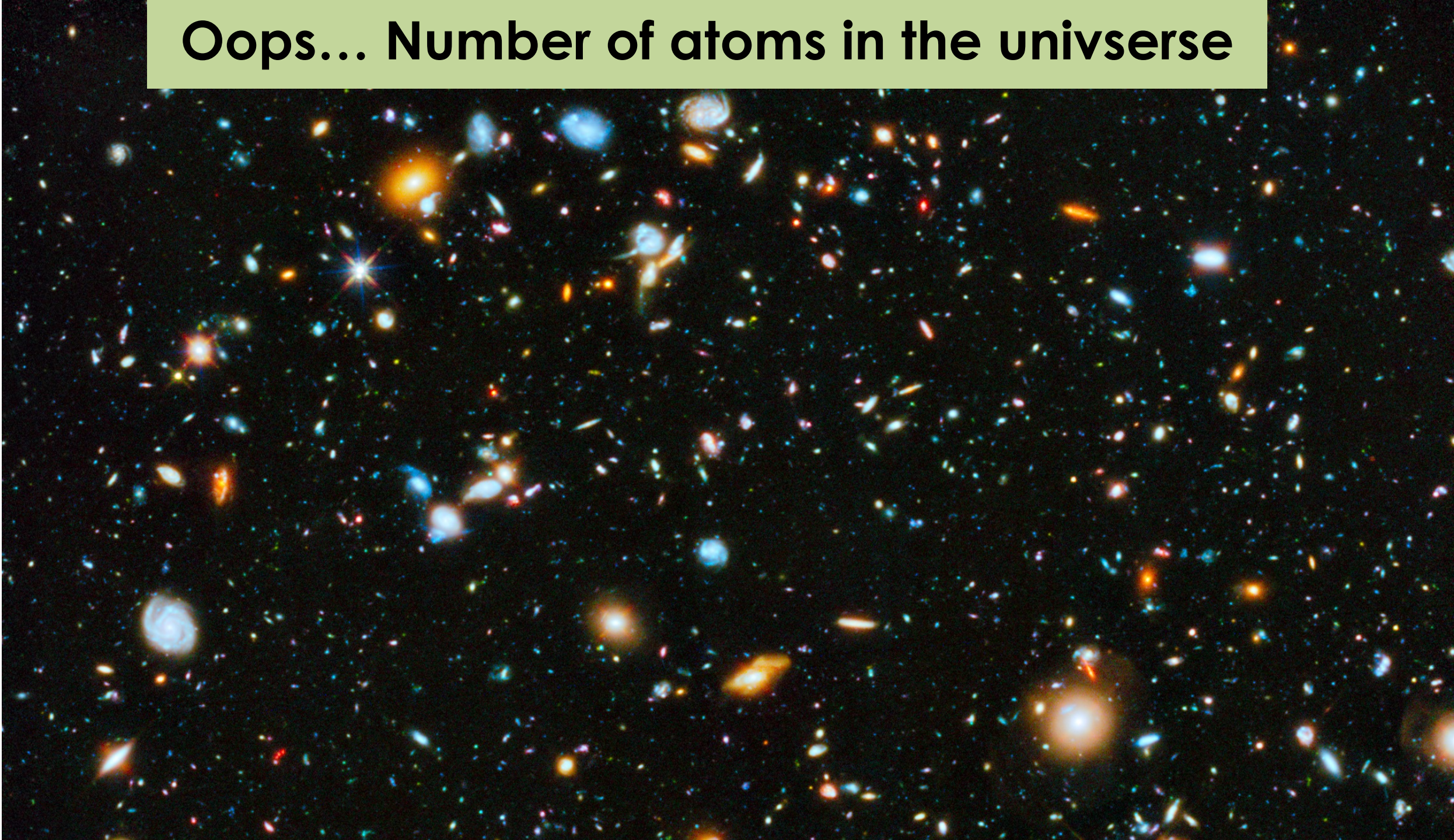
Simply chose the class label that is the most likely given the data

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$



$$P(x_1, x_2, x_3, \dots, x_{100}|y)$$

Oops... Number of atoms in the universe



What is the big O for # parameters?
 $m = \# \text{ features.}$

Big O of Brute Force Joint

What is the big O for # parameters?
 $m = \# \text{ features.}$

$$O(2^m)$$

*Assuming each feature is
binary...*

Not going to cut it!

What is the problem here?

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

$$P(\mathbf{x}|y) = P(x_1, x_2, \dots, x_m|y)$$

Naïve Bayes Assumption

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

$$P(\mathbf{x}|y) = P(x_1, x_2, \dots, x_m|y)$$

$$= \prod_i P(x_i|y)$$

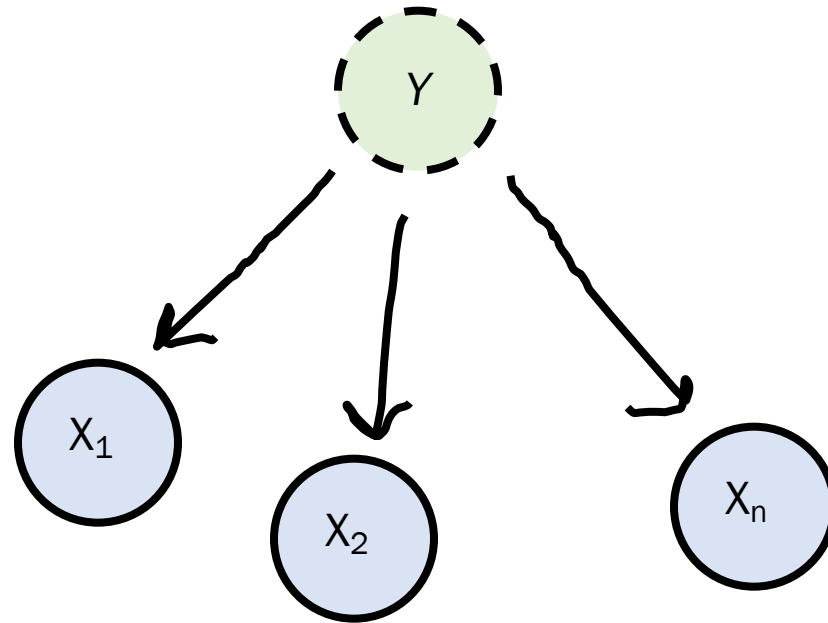
*The Naïve Bayes
assumption*



Naïve Bayes Assumption:

$$P(\mathbf{x}|y) = \prod_i P(x_i|y)$$

Naïve Bayes as a Model



Class Label

$Y \sim \text{Bern}$

Data distribution

$X_i|Y \sim \text{Bern}$

$$P(\mathbf{x}, y) = P(y) \prod_i P(x_i|y)$$

Naïve Bayes Classifier

Naïve Bayes

Our prediction
for y

Is a function of \mathbf{x}

That chooses the best
value of y given \mathbf{x}

$$\hat{y} = g(\mathbf{x}) = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(y|\mathbf{x})$$

$$= \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(\mathbf{x}|y) \hat{P}(y)$$

Bayes rule!

$$= \operatorname{argmax}_y \left(\prod_{i=1}^n \hat{P}(x_i|y) \right) \hat{P}(y)$$

Naïve Bayes
Assumption

$$= \operatorname{argmax}_y \log \hat{P}(y) + \sum_{i=1}^m \log \hat{P}(x_i|y)$$

This log version is useful for numerical
stability

Computing Probabilities from Data

Various probabilities you will need to compute for Naive Bayesian Classifier (using MLE here):

$$\hat{p}(X_i = 1|Y = 0) = \frac{(\# \text{ training examples where } X_i = 1 \text{ and } Y = 0)}{(\# \text{ training examples where } Y = 0)}$$

$$\hat{p}(Y = 1) = \frac{(\# \text{ training examples where } Y = 1)}{(\# \text{ training examples})}$$

Computing Probabilities from Data With Laplace

Various probabilities you will need to compute for Naive Bayesian Classifier (using MAP with Laplace):

$$\hat{p}(X_i = 1|Y = 0) = \frac{(\# \text{ training examples where } X_i = 1 \text{ and } Y = 0) + 1}{(\# \text{ training examples where } Y = 0) + 2}$$

$$\hat{p}(Y = 1) = \frac{(\# \text{ training examples where } Y = 1) + 1}{(\# \text{ training examples}) + 2}$$

Naïve Bayes Example

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

- Use training data to estimate params: $\hat{P}(x_i|y)$ $\hat{P}(y)$

$X_1 \backslash Y$	0	1	MLE estimates		$X_2 \backslash Y$	0	1	MLE estimates		Y	#	MLE est.
0	3	10	0.23	0.77	0	5	8	0.38	0.62	0	13	0.43
1	4	13	0.24	0.76	1	7	10	0.41	0.59	1	17	0.57

- Say someone likes **Star Wars ($X_1 = 1$)**, but not **Harry Potter ($X_2 = 0$)**
- Will they like “Lord of the Rings”? Need to predict Y:

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(\mathbf{x}|y)\hat{P}(y) = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(x_1|y)\hat{P}(x_2|y)\hat{P}(y)$$

Naïve Bayes Example

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

- Use training data to estimate params: $\hat{P}(x_i|y)$ $\hat{P}(y)$

Y \ X ₁	X ₁		MLE estimates		Y \ X ₂	X ₂		MLE estimates		Y	#	MLE est.
	0	1				0	1					
0	3	10	0.23	0.77	0	5	8	0.38	0.62	0	13	0.43
1	4	13	0.24	0.76	1	7	10	0.41	0.59	1	17	0.57

- Say someone likes **Star Wars ($X_1 = 1$)**, but not **Harry Potter ($X_2 = 0$)**
- Will they like “Lord of the Rings”? Need to predict Y :

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(X_1 = x_1 | Y = y) \hat{P}(X_2 = x_2 | Y = y) \hat{P}(Y = y)$$

Naïve Bayes Example

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

- Use training data to estimate params: $\hat{P}(x_i|y)$ $\hat{P}(y)$

$Y \backslash X_1$	0	1	MLE estimates		$Y \backslash X_2$	0	1	MLE estimates		Y	#	MLE est.
0	3	10	0.23	0.77	0	5	8	0.38	0.62	0	13	0.43
1	4	13	0.24	0.76	1	7	10	0.41	0.59	1	17	0.57

- Say someone likes **Star Wars ($X_1 = 1$)**, but not **Harry Potter ($X_2 = 0$)**
- Will they like “Lord of the Rings”? Need to predict Y :

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(X_1 = 1|Y = y) \hat{P}(X_2 = 0|Y = y) \hat{P}(Y = y)$$

MAP Naïve Bayes

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

○ Use training data to estimate PMFs:

$$\hat{P}(x_i|y) \quad \hat{P}(y)$$

$Y \backslash X_1$	0	1	MAP estimates	$Y \backslash X_2$	0	1	MAP estimates	Y	#	MAP est.
0	3	10		0	5	8		0	13	
1	4	13		1	7	10		1	17	

What prior?

MAP Naïve Bayes

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

○ Use training data to estimate PMFs:

$$\hat{P}(x_i|y) \quad \hat{P}(y)$$

$X_1 \backslash Y$	0	1	MAP estimates		$X_2 \backslash Y$	0	1	MAP estimates		Y	#	MAP est.
0	3	10	0.27	0.73	0	5	8			0	13	
1	4	13			1	7	10			1	17	

Laplace!

$$p_i = \frac{n_i + 1}{n + m} \qquad p_i = \frac{n_i + 1}{n + 2}$$

MAP Naïve Bayes

Predict Y based on observing variables X_1 and X_2

- X_1 and X_2 are both indicator variables
 - X_1 denotes “likes Star Wars”, X_2 denotes “likes Harry Potter”
- Y is indicator variable: “likes Lord of the Rings”

○ Use training data to estimate PMFs:

$$\hat{P}(x_i|y) \quad \hat{P}(y)$$

$X_1 \backslash Y$	0	1	MAP estimates		$X_2 \backslash Y$	0	1	MAP estimates		Y	#	MAP est.
0	3	10	0.27	0.73	0	5	8	0.4	0.6	0	13	0.45
1	4	13	0.26	0.74	1	7	10	0.42	0.58	1	17	0.55

Laplace!

$$p_i = \frac{n_i + 1}{n + m} \qquad p_i = \frac{n_i + 1}{n + 2}$$



Training Naïve Bayes, is estimating parameters for a multinomial (or bernoulli).

Thus training is just counting.

What is Bayes Doing in my Mail Server

This is spam:

<p>From: Abey Chavez [tristramu@deletedomains.com] To: sahami@robotics.stanford.edu Cc: Subject: For excellent metabolism</p> <p>Sent: Sat 5/22/03</p> <p>Canadian ** Pharmacy #1 Internet Inline Drugstore</p> <table><tr><td>Viagra Our price \$1.15</td><td>Cialis Our price \$1.99</td><td>Viagra Professional Our price \$3.73</td></tr><tr><td>Cialis Professional Our price \$4.17</td><td>Viagra Super Active Our price \$2.82</td><td>Cialis Super Active Our price \$3.66</td></tr><tr><td>Levitra Our price \$2.93</td><td>Viagra Soft Tabs Our price \$1.64</td><td>Cialis Soft Tabs Our price \$3.51</td></tr></table> <p>And more...</p> <p>Click here</p>	Viagra Our price \$1.15	Cialis Our price \$1.99	Viagra Professional Our price \$3.73	Cialis Professional Our price \$4.17	Viagra Super Active Our price \$2.82	Cialis Super Active Our price \$3.66	Levitra Our price \$2.93	Viagra Soft Tabs Our price \$1.64	Cialis Soft Tabs Our price \$3.51	<h2>Let's get Bayesian on your spam:</h2> <p>Content analysis details: (49.5 hits, 7.0 required)</p> <ul style="list-style-type: none">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] <p>8.0 BAYES_99 BODY: Bayesian spam probability is 99 to 100% [score: 1.0000]</p>
Viagra Our price \$1.15	Cialis Our price \$1.99	Viagra Professional Our price \$3.73								
Cialis Professional Our price \$4.17	Viagra Super Active Our price \$2.82	Cialis Super Active Our price \$3.66								
Levitra Our price \$2.93	Viagra Soft Tabs Our price \$1.64	Cialis Soft Tabs Our price \$3.51								

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

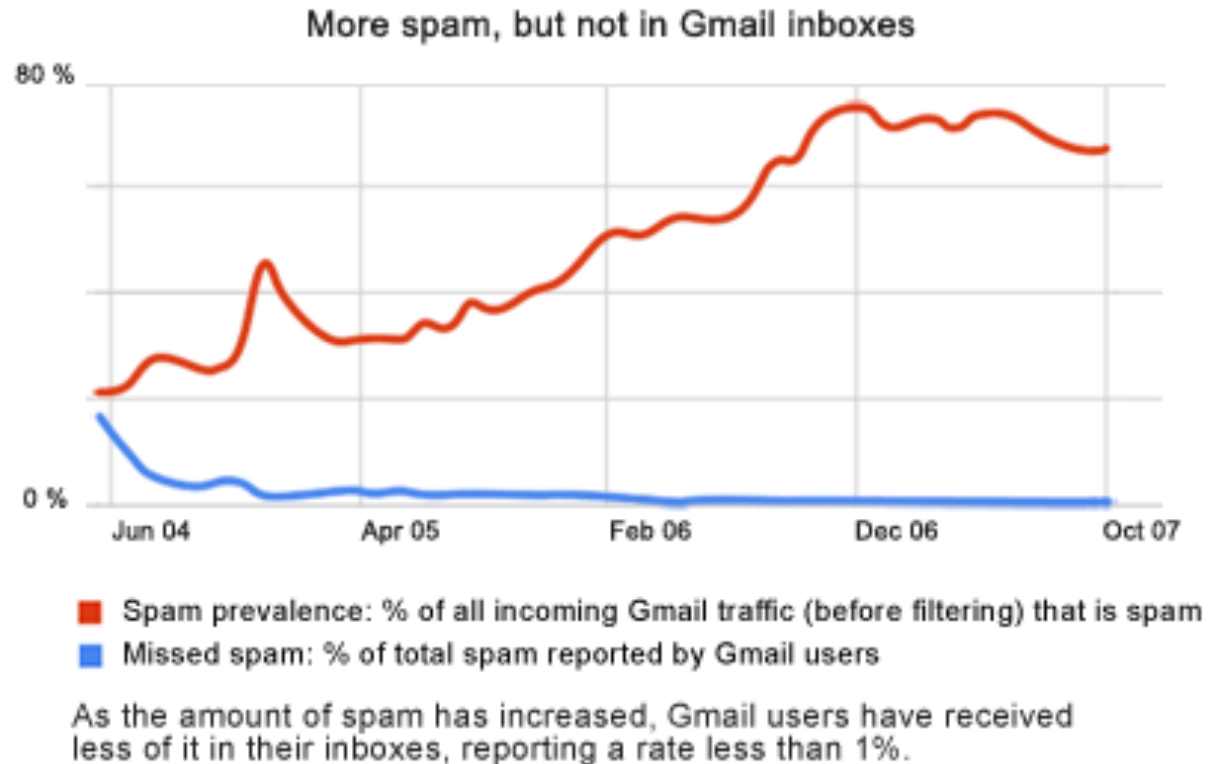
Abstract

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

Spam, Spam... Go Away!

The constant battle with spam



“And machine-learning algorithms developed to merge and rank large sets of Google search results allow us to combine hundreds of factors to classify spam.”

Email Classification

Want to predict if an email is spam or not

- Start with the input data
 - Consider a lexicon of m words (Note: in English $m \approx 100,000$)
 - Define m indicator variables $\mathbf{X} = \langle X_1, X_2, \dots, X_m \rangle$
 - Each variable X_i denotes if word i appeared in a document or not
 - Note: m is huge, so make “Naive Bayes” assumption
- Define output classes Y to be: {spam, non-spam}
- Given training set of N previous emails
 - For each email message, we have a training instance: $\langle X_1, X_2, \dots, X_m \rangle$ noting for each word, if it appeared in email $\mathbf{X} =$
 - Each email message is also marked as spam or not (value of Y)

Training the Classifier

Given N training pairs:

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

Learning

- Estimate probabilities $P(y)$ and $P(x_i | y)$ for all i
 - Many words are likely to not appear at all in given set of email
- Laplace estimate: $\hat{p}(X_i = 1 | Y = spam)_{Laplace} = \frac{(\# \text{ spam emails with word } i) + 1}{\text{total } \# \text{ spam emails} + 2}$

Classification

- For a new email, generate $\mathbf{X} = \langle X_1, X_2, \dots, X_m \rangle$
- Classify as spam or not using: $\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(\mathbf{x}|y) \hat{P}(y)$
- Employ Naive Bayes assumption: $P(\mathbf{x}|y) = \prod_i P(x_i|y)$



Training Naïve Bayes, is
estimating parameters for
Bernoulli's.

Thus it is just counting.

How Does This Do?

After training, can test with another set of data

- “Testing” set also has known values for Y, so we can see how often we were right/wrong in predictions for Y
- Spam data
 - Email data set: 1789 emails (1578 spam, 211 non-spam)
 - First, 1538 email messages (by time) used for training
 - Next 251 messages used to test learned classifier
- Criteria:
 - Precision = # *correctly* predicted class Y / # predicted class Y
 - Recall = # *correctly* predicted class Y / # real class Y messages

	Spam		Non-spam	
	Precision	Recall	Precision	Recall
Words only	97.1%	94.3%	87.7%	93.4%
Words + add'l features	100%	98.3%	96.2%	100%

Naïve Bayes Classification