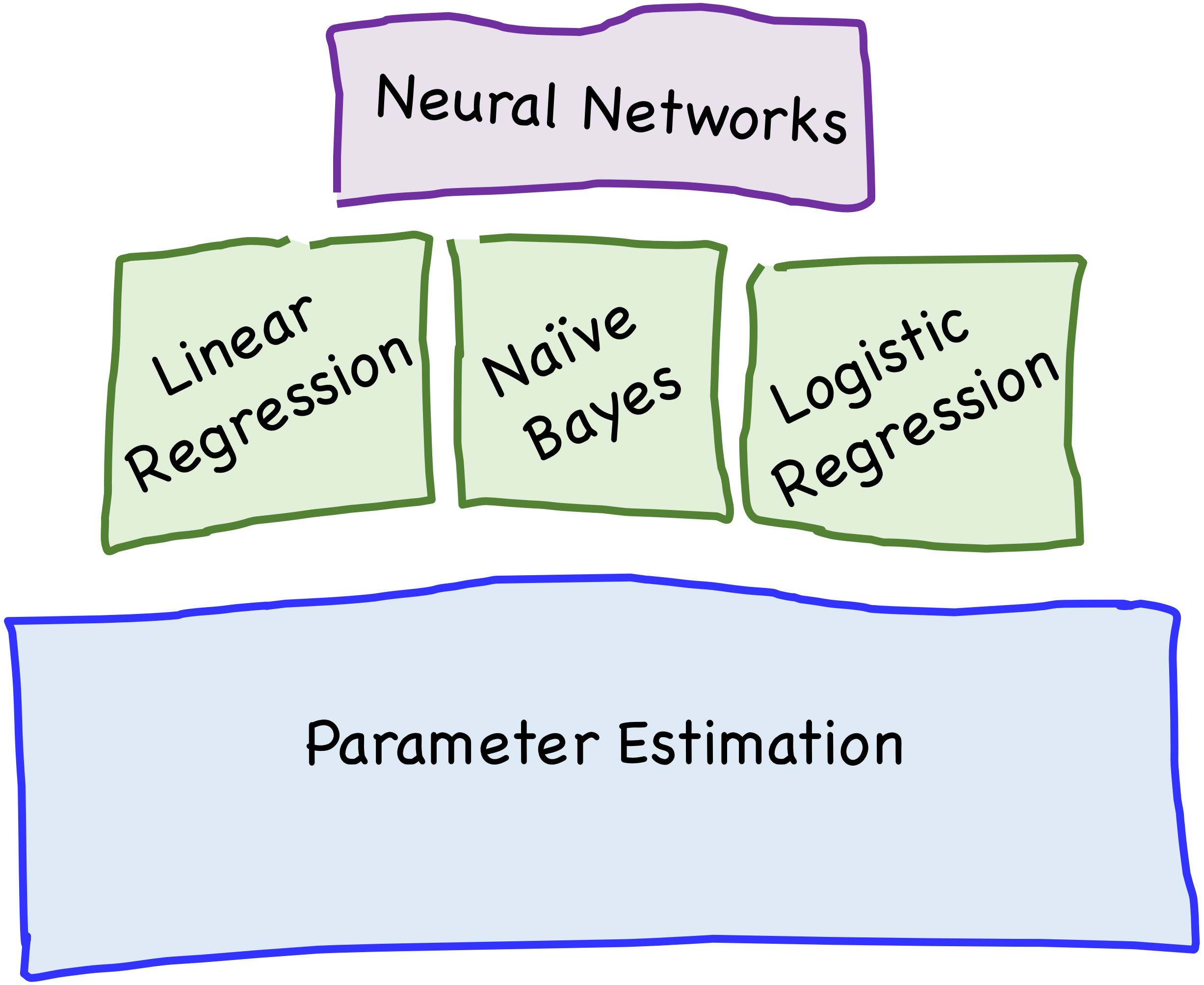
Naïve Bayes

Chris Piech
CS109, Stanford University

Our Path


Neural Networks
Linear Regression
Naïve Bayes
Logistic Regression
Parameter Estimation

MLE vs MAP

Maximum Likelihood Estimation

$$\theta_{\mathrm{MLE}} = \underset{\theta}{\operatorname{argmax}}\ f(X_1, X_2, \ldots, X_n|\theta)$$

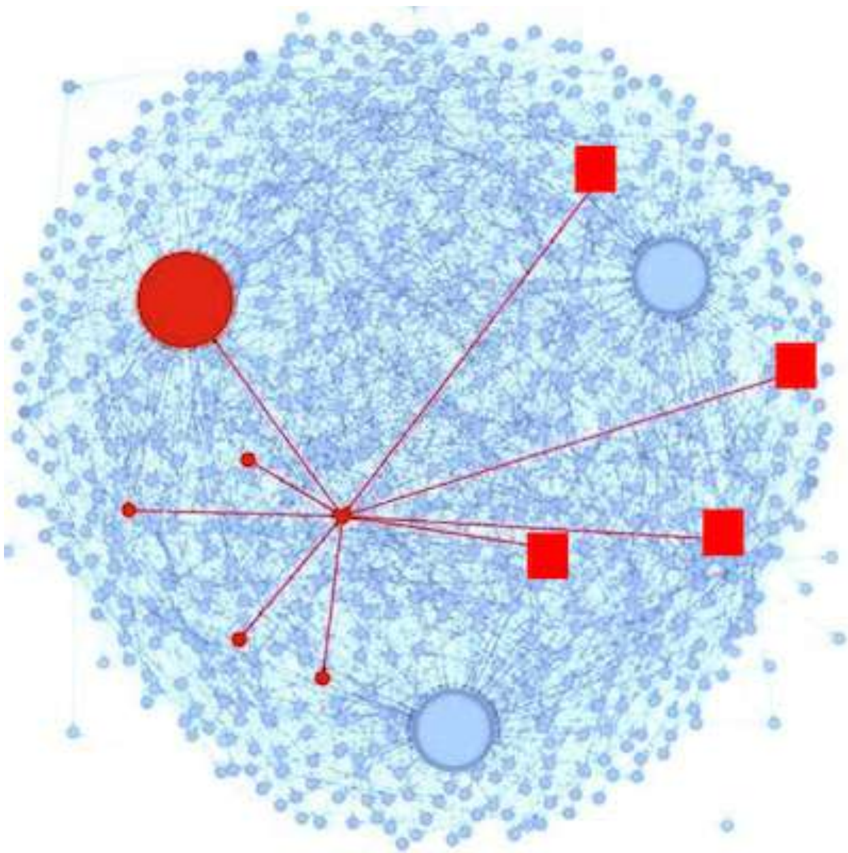
$$= \underset{\theta}{\operatorname{argmax}}\ \sum_i \log f(X_i|\theta)$$

Maximum A Posteriori

$$\theta_{\mathrm{MAP}} = \underset{\theta}{\operatorname{argmax}}\ f(\theta|X_1, X_2, \ldots, X_n)$$

$$= \underset{\theta}{\operatorname{argmax}}\ \Big(\log g(\theta) + \sum_i \log f(X_i|\theta)\Big)$$

Is Peer Grading Accurate Enough?

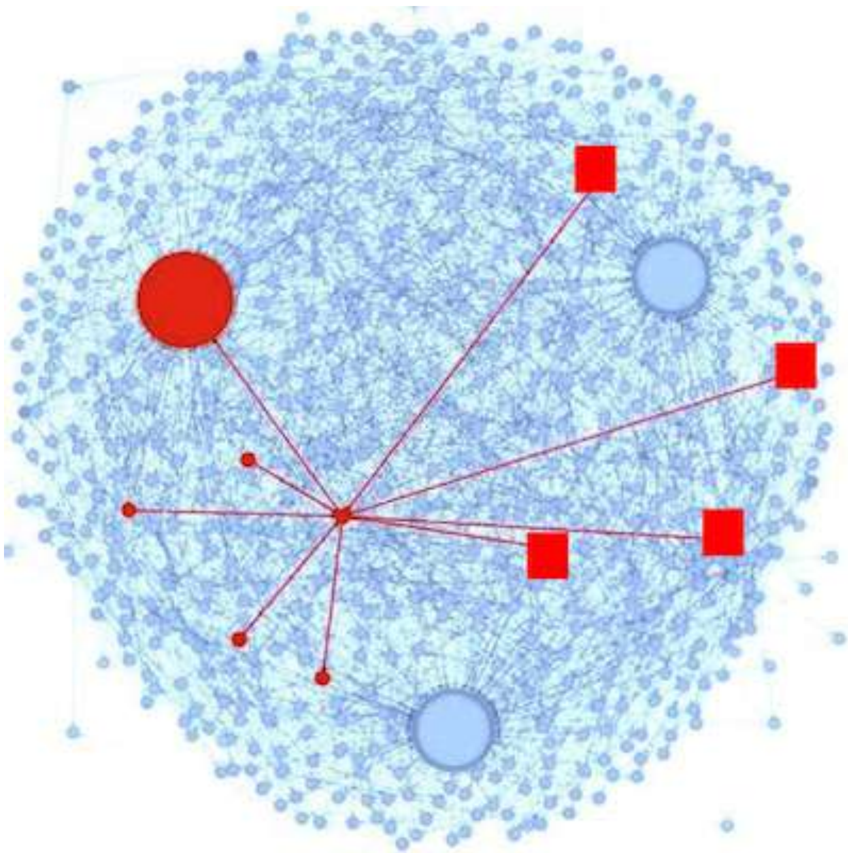


Peer Grading on Coursera HCI.

31,067 peer grades for 3,607 students.

Tuned Models of Peer Assessment. C Piech, J Huang, A Ng, D Koller

Is Peer Grading Accurate Enough?



1. Defined random variables for:

- True grade (s_i) for assignment i
- Observed (z_i^j) score for assign i
- Bias (b_j) for each grader j
- Variance (r_j) for each grader j

2. Designed a probabilistic model that defined the distributions for all random variables

$$z_i^j \sim \mathcal{N}(\mu = s_i + b_j, \sigma = \sqrt{r_j})$$

$$s_i \sim N(\mu_0, \sigma_0)$$

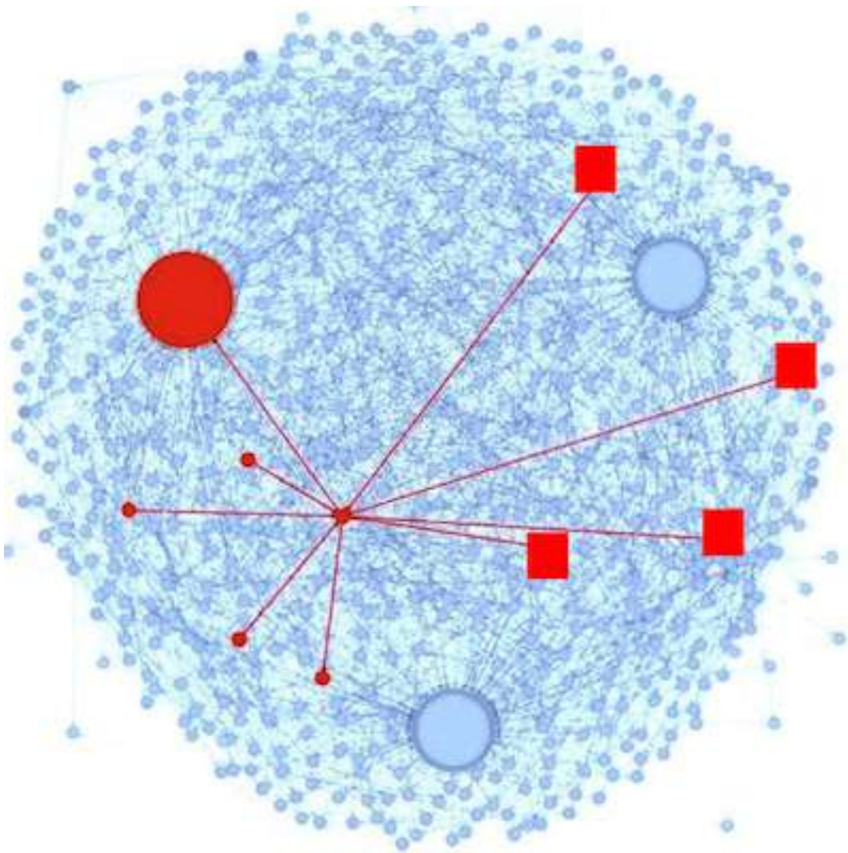
$$b_i \sim N(0, \eta_0)$$

$$r_i \sim \mathrm{InvGamma}(\alpha_0, \theta_0)$$

= hyperparameter

Tuned Models of Peer Assessment. C Piech, J Huang, A Ng, D Koller

Is Peer Grading Accurate Enough?



1. Defined random variables for:
- True grade (s_i) for assignment i
- Observed (z_i^j) score for assign i
- Bias (b_j) for each grader j
- Variance (r_j) for each grader j

2. Designed a probabilistic model that defined the distributions for all random variables

3. Found variable assignments using MAP estimation given the observed data

Inference or Machine Learning

Tuned Models of Peer Assessment. C Piech, J Huang, A Ng, D Koller

Gonna Need Priors

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

Multinomial Parameter Estimation

- Recall example of 6-sides die rolls:
  - X ~ 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
  - Dirichlet prior allows us to pretend we saw each outcome k times before. MAP estimate: $p_i = \frac{X_i + k}{n + mk}$
    - Laplace's "law of succession": idea above with k = 1
    - Laplace estimate: $p_i = \frac{X_i + 1}{n + m}$
    - 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!

The last estimator has risen…

Machine Learning

Machine Learning: Formally

- Many different forms of "Machine Learning"
  - We focus on the problem of *prediction*
- Want to make a prediction based on observations
  - Vector **X** of *m* observed variables: $<X_1, X_2, \ldots, X_m>$
    - $X_1, X_2, \ldots, X_m$ are called "input features/variables"
  - Based on observed **X**, want to predict unseen variable Y
    - Y called "output feature/variable" (or the "dependent variable")
  - Seek to "learn" a function $g(\boldsymbol{X})$ to predict Y: $\hat{Y} = g(\boldsymbol{X})$
    - When Y is discrete, prediction of Y is called "classification"
    - When Y is continuous, prediction of Y is called "regression"

Training Data

Assume IID data:

N training datapoints

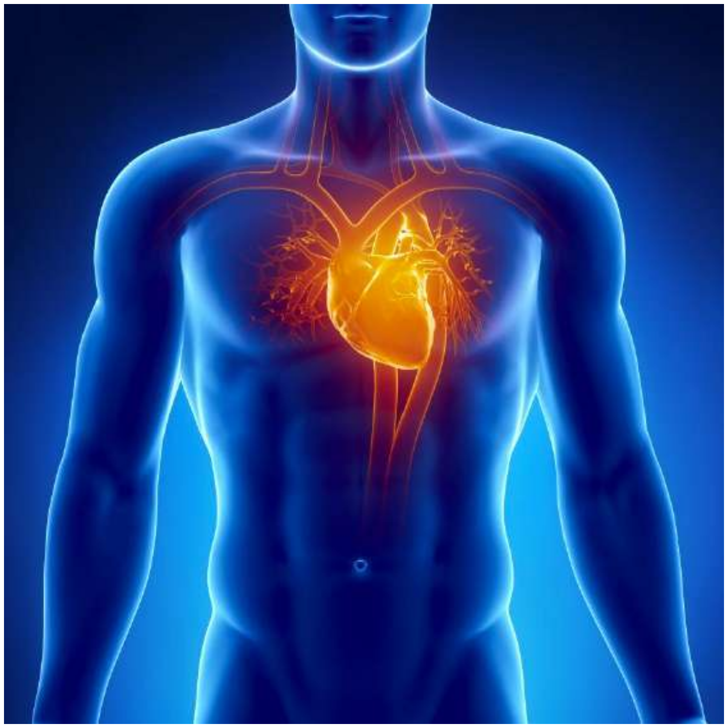
$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(n)}, y^{(n)})$$

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

Each datapoint has m features and a single output

Example Datasets

Heart



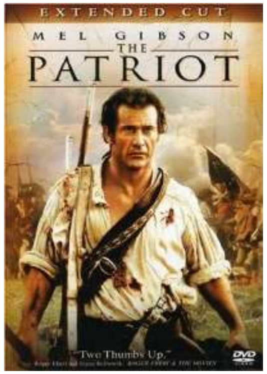
Ancestry

Netflix

Target Movie "Like" Classification

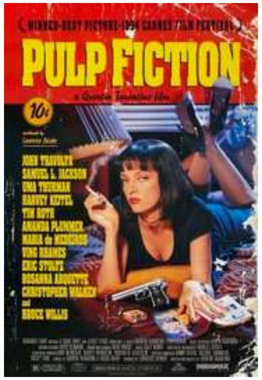
Movie 1

Movie 2

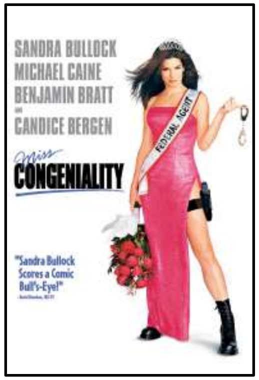


…

Movie m

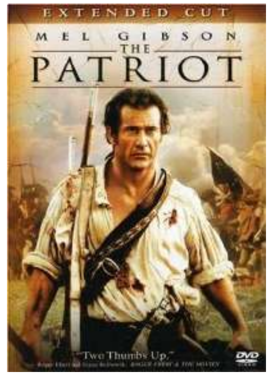


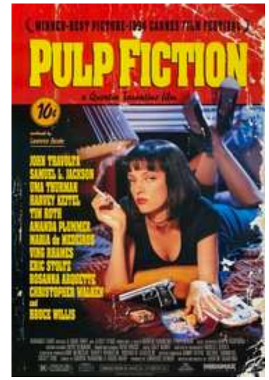
Output



	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

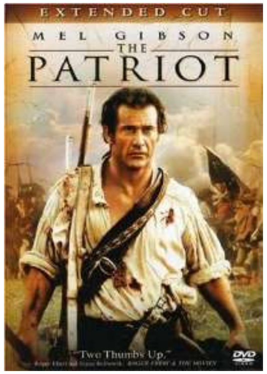


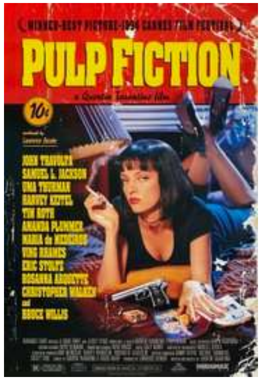


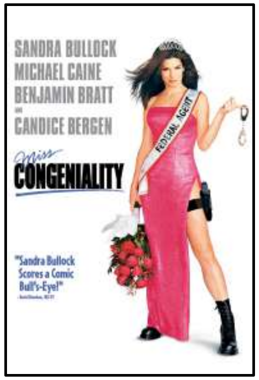
	Movie 1	Movie 2	…	Movie m	Output
User 1	1	0		1	1
User 2	1	1		0	0
			$\vdots$		$\vdots$
User n	0	0		1	1

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

Feature Vector



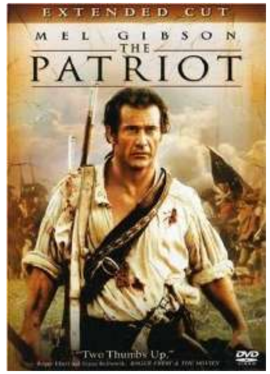


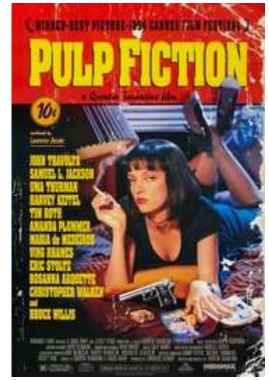


	Movie 1	Movie 2	$\cdots$	Movie m	Output
User 1	1	0		1	1
User 2	1	1		0	0
			$\vdots$		$\vdots$
User n	0	0		1	1

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

Output Value

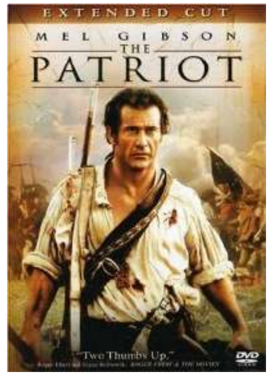


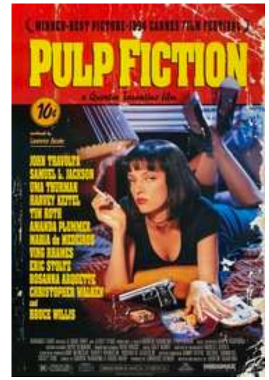


	Movie 1	Movie 2	$\cdots$	Movie m	Output
User 1	1	0		1	1
User 2	1	1		0	$\boxed{0}$
			$\vdots$		$\vdots$
User n	0	0		1	1

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

Single Feature Value

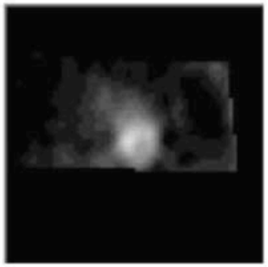




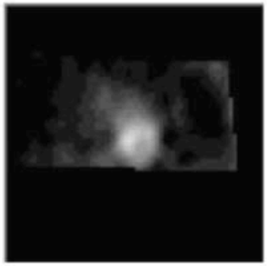
	Movie 1	Movie 2	$\cdots$	Movie m	Output
User 1	1	0		1	1
User 2	1	1		0	0
$\vdots$			$\vdots$		$\vdots$
User n	0	0		1	1

In general: $\mathbf{x}_j^{(i)}$ In this case: $\mathbf{x}_m^{(2)}$

Healthy Heart Classifier

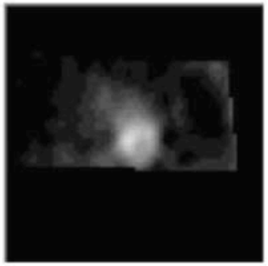
	ROI 1	ROI 2		ROI m	Output
			…		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

Healthy Heart Classifier

	ROI 1	ROI 2		ROI m	Output
			…		
Heart 1	0	$\boxed{1}$		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

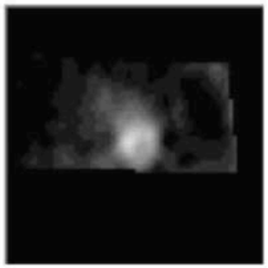
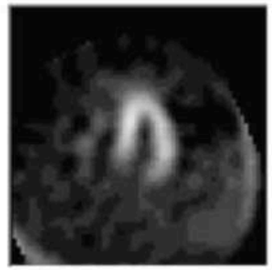
$$x_2^{(1)}$$

Healthy Heart Classifier

	ROI 1	ROI 2		ROI m	Output
			$\cdots$		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			$\vdots$		$\vdots$
Heart n	0	0		0	1

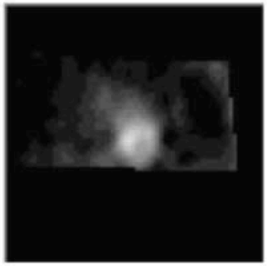
$$(\mathbf{x}^{(2)}, y^{(2)})$$

Healthy Heart Classifier

	ROI 1	ROI 2		ROI m	Output
			$\cdots$		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
		$\vdots$			$\vdots$
Heart n	0	0		0	1

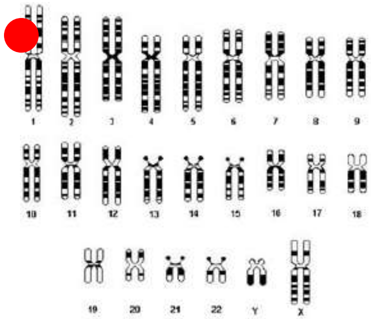
$\mathbf{x}^{(2)}$

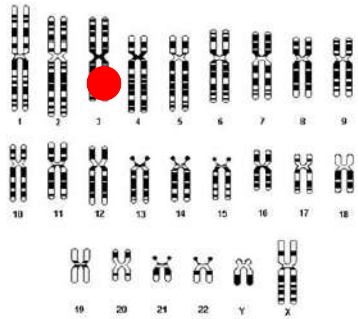
Healthy Heart Classifier

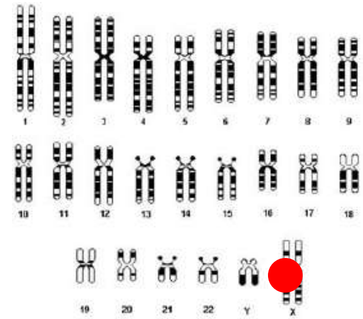
	ROI 1	ROI 2		ROI m	Output
			$\cdots$		
Heart 1	0	1		1	0
Heart 2	1	1		1	$\boxed{0}$
			$\vdots$		$\vdots$
Heart n	0	0		0	1

$y^{(2)}$

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

How Many Points will Warriors Score

	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

Assume IID data:

N training datapoints

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

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

Each datapoint has m features and a single output

Machine learning problems are expressed abstractly.

We can write machine learning algorithms generally.

A (Very Short) List of Applications

- Machine learning widely used in many contexts
  - Stock price prediction
    - Using economic indicators, predict if stock will go up/down
  - Computational biology and medical diagnosis
    - Predicting gene expression based on DNA
    - Determine likelihood for cancer using clinical/demographic data
  - Credit card fraud and telephone fraud detection
    - Based on past purchases/phone calls is a new one fraudulent?
      - Saves companies *billions(!)* of dollars annually
  - Spam E-mail detection (gmail, hotmail, many others)

That list is ridiculously short ☺

Linear Regression

A Grounding Example: Linear Regression

Problem: Predict real value Y based on observing variable X

Model: Linear weight for every feature

$$
\begin{aligned}
\hat{Y} &= \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_{n-1} X_{n-1} + \theta_n 1 \\
&= \theta^T \mathbf{X}
\end{aligned}
$$

Training: Chose the best thetas to describe your data

$$
\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} \; -\sum_{i=1}^{n} (Y^{(i)} - \theta^T \mathbf{x}^{(i)})^2
$$

Predicting Warriors

X_1 = Opposing team ELO

X_2 = Points in last game

X_3 = Curry playing?

X_4 = Playing at home?

Y = Warriors points

Predicting CO_2

X_1 = Temperature

X_2 = Elevation

X_3 = CO_2 level yesterday

X_4 = GDP of region

X_5 = Acres of forest growth

Y = CO_2 levels

Regression Data

Assume IID data:

N training datapoints

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

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

Each datapoint has m features and a single output

How Many Points will Warriors Score

	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

How Did We Get Linear Regression?

N training pairs: $(\boldsymbol{x}^{(1)}, y^{(1)})$, $(\boldsymbol{x}^{(2)}, y^{(2)})$, …, $(\boldsymbol{x}^{(N)}, y^{(N)})$

1. Linear Regression Model:

$$\begin{aligned} Y &= \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_{n-1} X_{n-1} + \theta_n 1 + Z \\ &= \theta^T \mathbf{X} + Z \end{aligned}$$

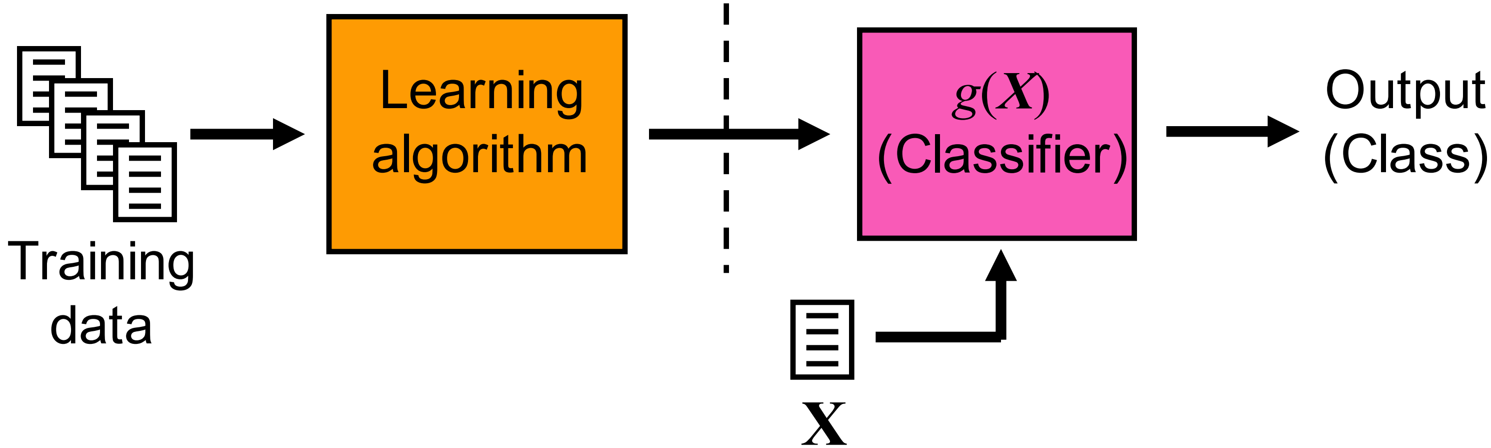
$$Z \sim N(0, \sigma^2)$$

2. Find the LL function and chose thetas which maximize it

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} \; -\sum_{i=1}^{n} (Y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$$

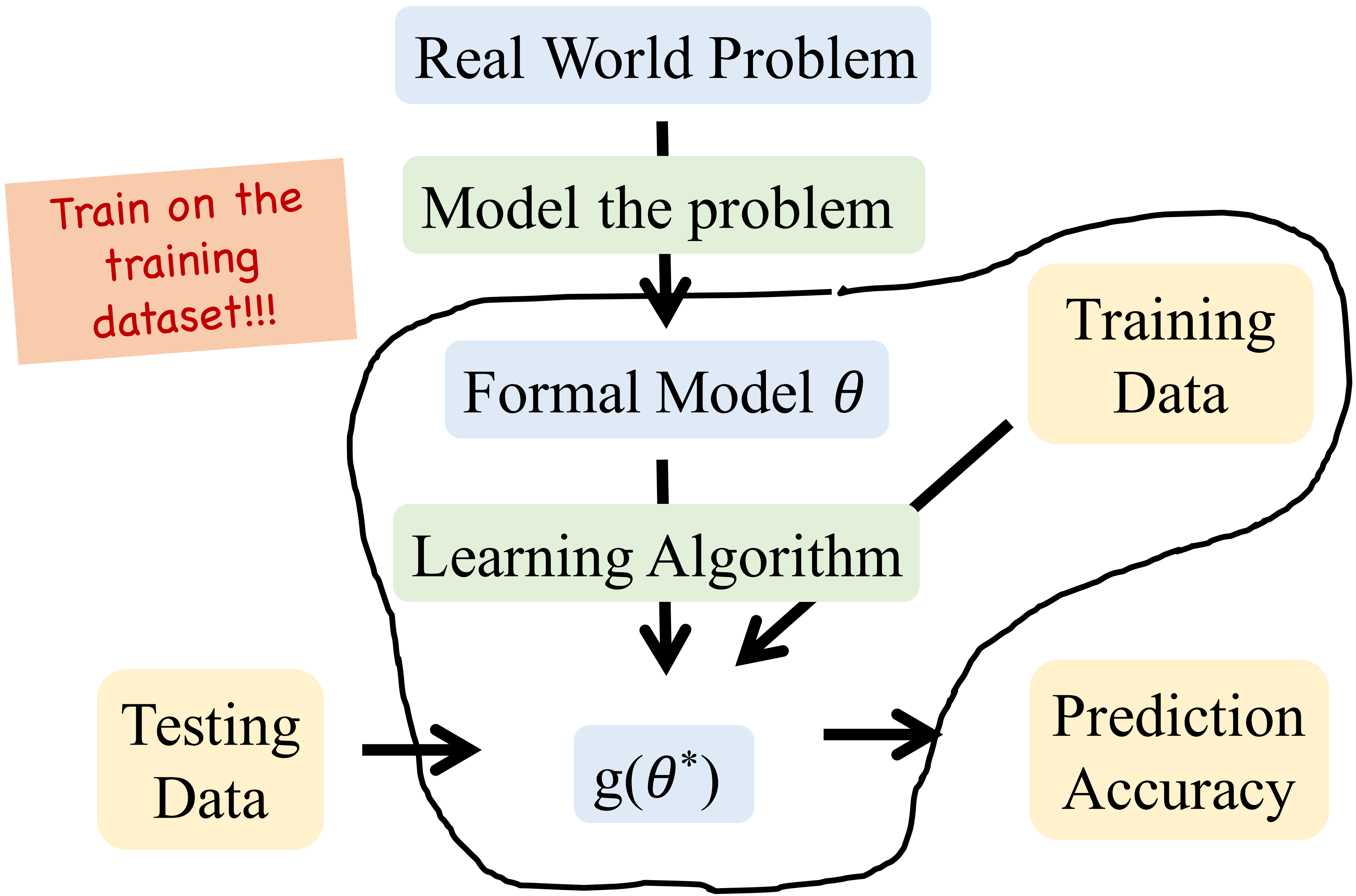
3. Use an optimizer to calculate each theta.

The Machine Learning Process



- <u>Training data</u>: set of *N* pre-classified data instances
  - *N* training pairs: $(\boldsymbol{x}^{(1)},y^{(1)}),\ (\boldsymbol{x}^{(2)},y^{(2)}),\ \ldots,\ (\boldsymbol{x}^{(n)},\ y^{(n)})$
    - Use superscripts to denote *i*-th training instance
- <u>Learning algorithm</u>: method for determining $g(\boldsymbol{X})$
  - Given a new input observation of $\boldsymbol{x} = x_1, x_2, \ldots, x_m$
  - Use $g(\boldsymbol{x})$ to compute a corresponding output (prediction)

Training



Real World Problem
Model the problem
Formal Model θ
Learning Algorithm
Training Data
Train on the training dataset!!!
Testing Data
$g(\theta^*)$
Prediction Accuracy

Predicting Warriors

Y = Warriors points

$$\hat{Y} = \theta_1 X_1 + \theta_2 X_2 + \dots \theta_{n-1} X_{n-1} + \theta_n 1$$
$$= \theta^T \mathbf{X}$$

X_1 = Opposing team ELO

X_2 = Points in last game

X_3 = Curry playing?

X_4 = Playing at home?

θ_1 = -2.3

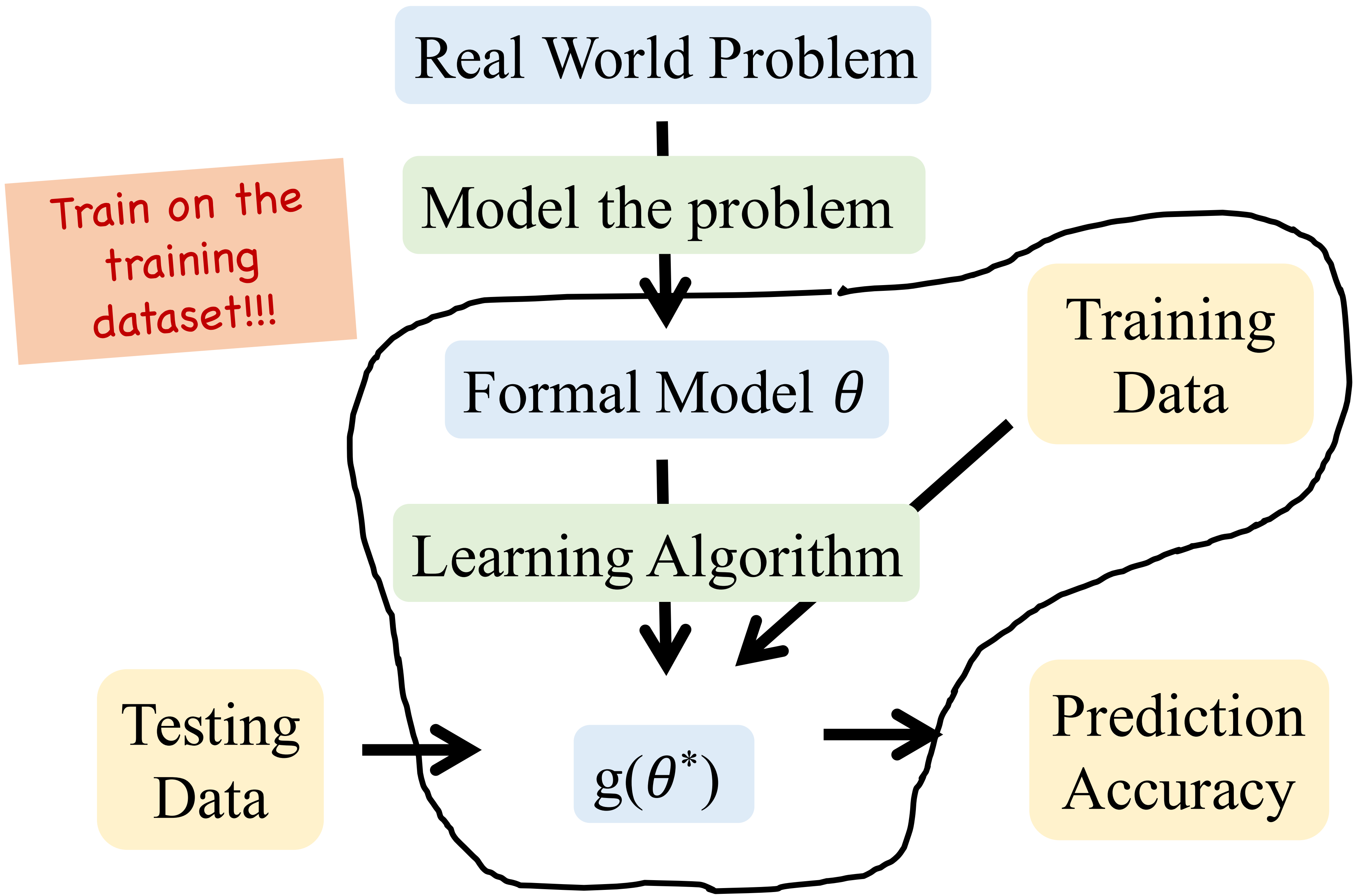
θ_2 = +1.2

θ_3 = +10.2

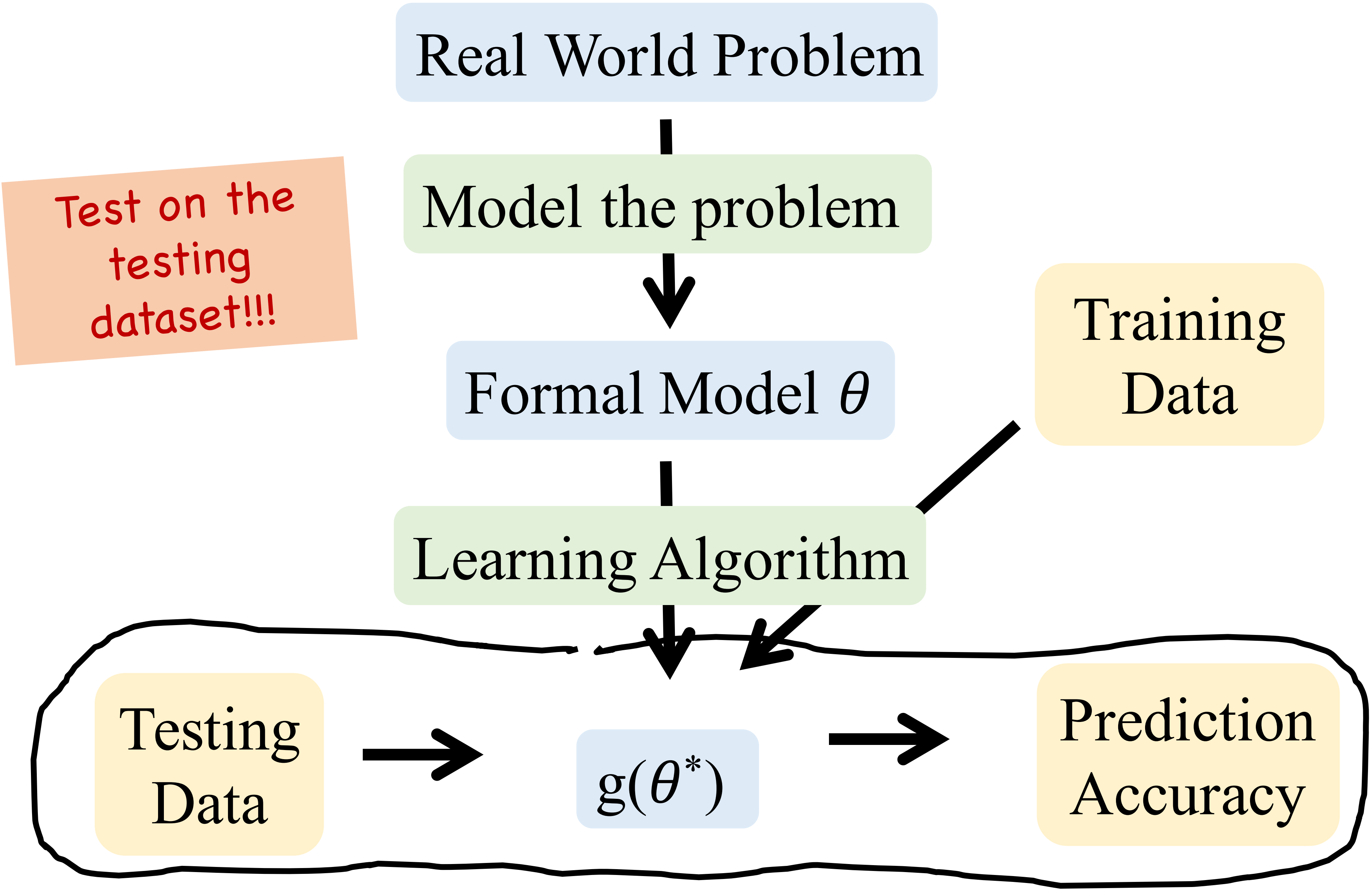
θ_4 = +3.3

θ_5 = +95.4

Training


Train on the training dataset!!!
Real World Problem
Model the problem
Formal Model θ
Training Data
Learning Algorithm
Testing Data
g(θ*)
Prediction Accuracy

Testing

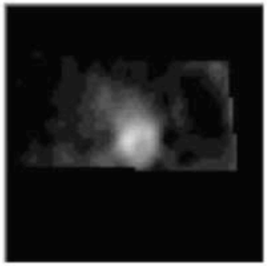


Test on the testing dataset!!!
Real World Problem
Model the problem
Formal Model θ
Training Data
Learning Algorithm
Testing Data
$g(\theta^*)$
Prediction Accuracy

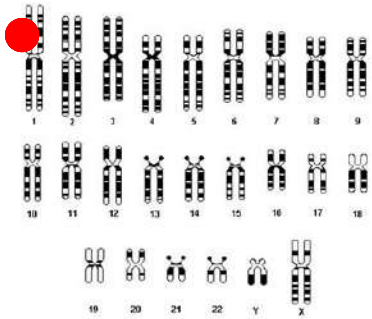
That is linear regression ☺

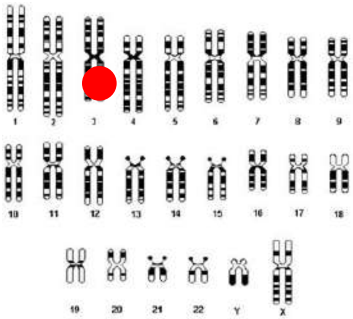
Classification

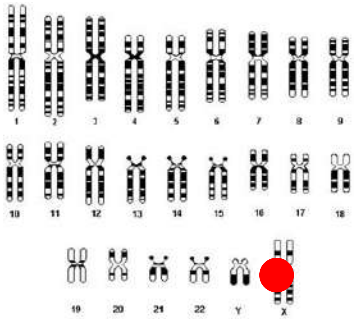
Healthy Heart Classifier

	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







	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

NETFLIX

And Learn

Target Movie "Like" Classification

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

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

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

Fake Algorithm:
Brute Bayes Classifier

Brute Force Bayes

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

Whether or not they liked Independence day

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

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

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

This is for one user

Brute Force Bayes

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

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

This is for one user

Brute Force Bayes

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

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

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

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

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

Joint probability table

Y \ X_1	0	1
0	θ_0	θ_1
1	θ_2	θ_3

y
θ_4
θ_5

Learn these during training

Training

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

$Y \backslash X_1$	0	1
0	θ_0	θ_1
1	θ_2	θ_3

Let (x_1, y) be one giant multinomial

MLE Estimate

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

Y \ X_1	0	1
0	0.2	0.0
1	0.3	0.5

MLE: Just count

MAP Estimate

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

Y \ X_1	0	1
0	0.20	0.01
1	0.30	0.49

Add Laplace smoothing

Testing

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

Y \ X_1	0	1
0	0.20	0.01
1	0.30	0.49

y
0.21
0.79

Test user: Likes independence day

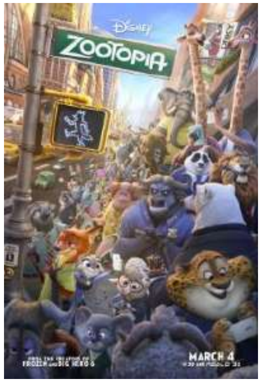
$$P(x_1 = 1|y = 0)P(y = 0)$$

vs

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

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

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

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

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

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

Brute Force Bayes

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

X_2 = 0

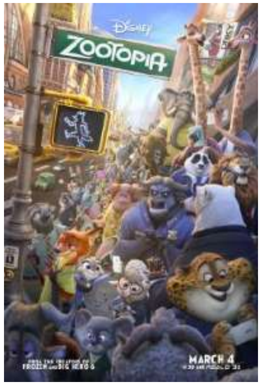
Y \ X_1	0	1
0	θ_0	θ_1
1	θ_2	θ_3

X_2 = 1

Y \ X_1	0	1
0	θ_4	θ_5
1	θ_6	θ_7

x_1

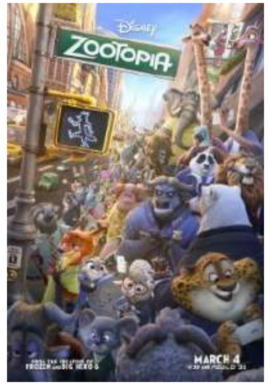
x_2

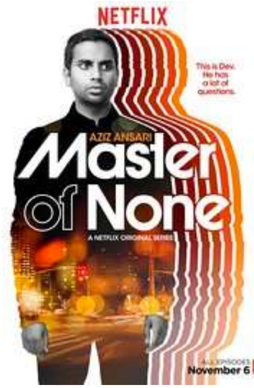


y

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
				$\vdots$
User n	0	1	1	1

Brute Force Bayes m = 3

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

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

$$= \underset{y=\{0,1\}}{\operatorname{argmax}} \; \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

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

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

Brute Force Bayes

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

$X_3 = 0$

$X_2 = 0$

Y \ X_1	0	1
0	θ_0	θ_1
1	θ_2	θ_3

$X_2 = 1$

Y \ X_1	0	1
0	θ_8	θ_9
1	θ_{10}	θ_{11}

$X_3 = 1$

$X_2 = 0$

Y \ X_1	0	1
0	θ_4	θ_5
1	θ_6	θ_7

$X_2 = 1$

Y \ X_1	0	1
0	θ_{12}	θ_{13}
1	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

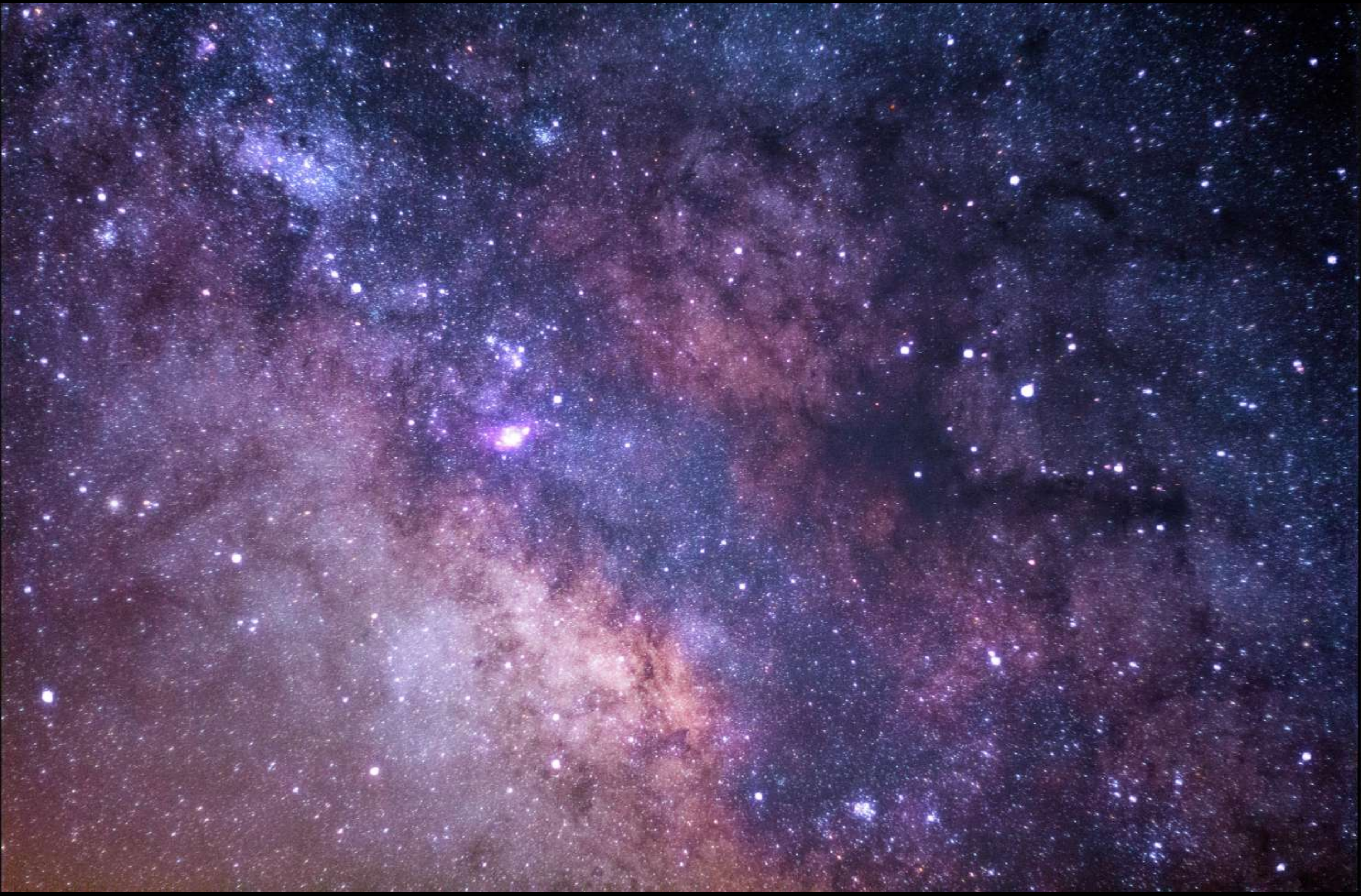
$$\hat{y} = \underset{y=\{0,1\}}{\text{argmax}}\ P(y|\mathbf{x})$$

$$= \underset{y=\{0,1\}}{\text{argmax}}\ \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

$$= \underset{y=\{0,1\}}{\text{argmax}}\ P(\mathbf{x}|y)P(y)$$

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

Oops… Number of atoms in the univserse



What is the big O for # parameters?
m = # features.

Big O of Brute Force Joint

What is the big O for # parameters?
m = # features.

$$O(2^m)$$

Assuming each feature
is binary...

Not going to cut it!

What is the problem here?

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

$$= \underset{y=\{0,1\}}{\operatorname{argmax}} \; \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

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

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

Naïve Bayes Assumption

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

$$= \underset{y=\{0,1\}}{\operatorname{argmax}} \; \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

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

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

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

The Naïve Bayes assumption

Naïve Bayes Assumption:

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

Naïve Bayes Classifier

Naïve Bayes Classifier

- Say, we have m input values $\mathbf{X} = \langle X_1, X_2, \ldots, X_m \rangle$
  - Assume variables $X_1, X_2, \ldots, X_m$ are **conditionally independent** given Y
    - Really don't believe $X_1, X_2, \ldots, X_m$ are conditionally independent
    - Just an approximation we make to be able to make predictions
    - This is called the "Naive Bayes" assumption, hence the name
  - Predict Y using $\hat{Y} = \arg\max_y P(\mathbf{X}, Y) = \arg\max_y P(\mathbf{X} \mid Y) P(Y)$
    - But, we now have:

$$P(\mathbf{X} \mid Y) = P(X_1, X_2, \ldots X_m \mid Y) = \prod_{i=1}^{m} P(X_i \mid Y)$$ by conditional independence

  - Note: computation of PMF table is linear in m : O(m)
    - Don't need much data to get good probability estimates

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 PMFs: $\hat{p}_{X_i,Y}(x_i, y), \ \hat{p}_Y(y)$

Y \ X_1	0	1	MLE estimates	
0	3	10	0.10	0.33
1	4	13	0.13	0.43

Y \ X_2	0	1	MLE estimates	
0	5	8	0.17	0.27
1	7	10	0.23	0.33

Y	#	MLE est.
0	13	0.43
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:

$$\hat{Y} = \arg\max_y \hat{P}(\mathbf{X} \mid Y)\hat{P}(Y) = \arg\max_y \hat{P}(X_1 \mid Y)\hat{P}(X_2 \mid Y)\hat{P}(Y)$$

One SciFi/Fantasy to Rule them All

Y \ X_1	0	1	MLE estimates	
0	3	10	0.10	0.33
1	4	13	0.13	0.43

Y \ X_2	0	1	MLE estimates	
0	5	8	0.17	0.27
1	7	10	0.23	0.33

Y	#	MLE est.
0	13	0.43
1	17	0.57

- Prediction for Y is value of Y maximizing P(**X**, Y):

$$\hat{Y} = \arg\max_y \hat{P}(\mathbf{X} \mid Y)\hat{P}(Y) = \arg\max_y \hat{P}(X_1 \mid Y)\hat{P}(X_2 \mid Y)\hat{P}(Y)$$

  - Compute P(**X**, Y=0): $\hat{P}(X_1 = 1 \mid Y = 0)\hat{P}(X_2 = 0 \mid Y = 0)\hat{P}(Y = 0)$

$$= \frac{\hat{P}(X_1 = 1, Y = 0)}{\hat{P}(Y = 0)} \frac{\hat{P}(X_2 = 0, Y = 0)}{\hat{P}(Y = 0)} \hat{P}(Y = 0) \approx \frac{0.33}{0.43}\frac{0.17}{0.43} 0.43 \approx 0.13$$

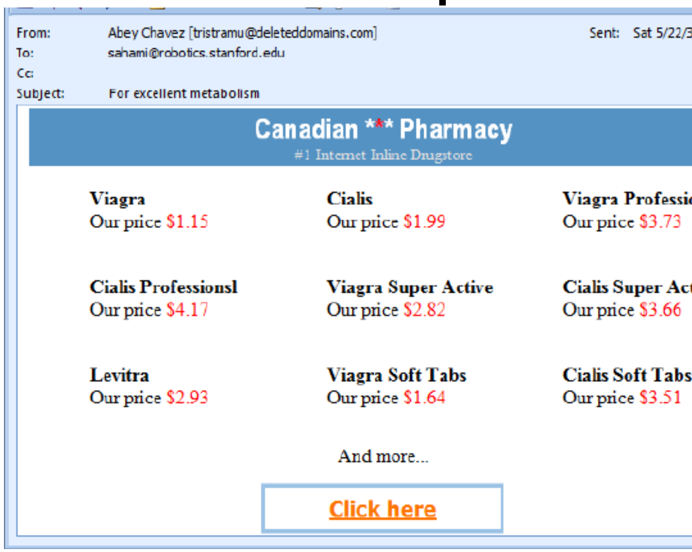
  - Compute P(**X**, Y=1): $\hat{P}(X_1 = 1 \mid Y = 1)\hat{P}(X_2 = 0 \mid Y = 1)\hat{P}(Y = 1)$

$$= \frac{\hat{P}(X_1 = 1, Y = 1)}{\hat{P}(Y = 1)} \frac{\hat{P}(X_2 = 0, Y = 1)}{\hat{P}(Y = 1)} \hat{P}(Y = 1) \approx \frac{0.43}{0.57}\frac{0.23}{0.57} 0.57 \approx 0.17$$

  - Since P(**X**, Y=1) > P(**X**, Y=0), we predict Ŷ = 1

What is Bayes Doing in my Mail Server

- This is spam:



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]
```

Who was crazy enough to think of that?

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

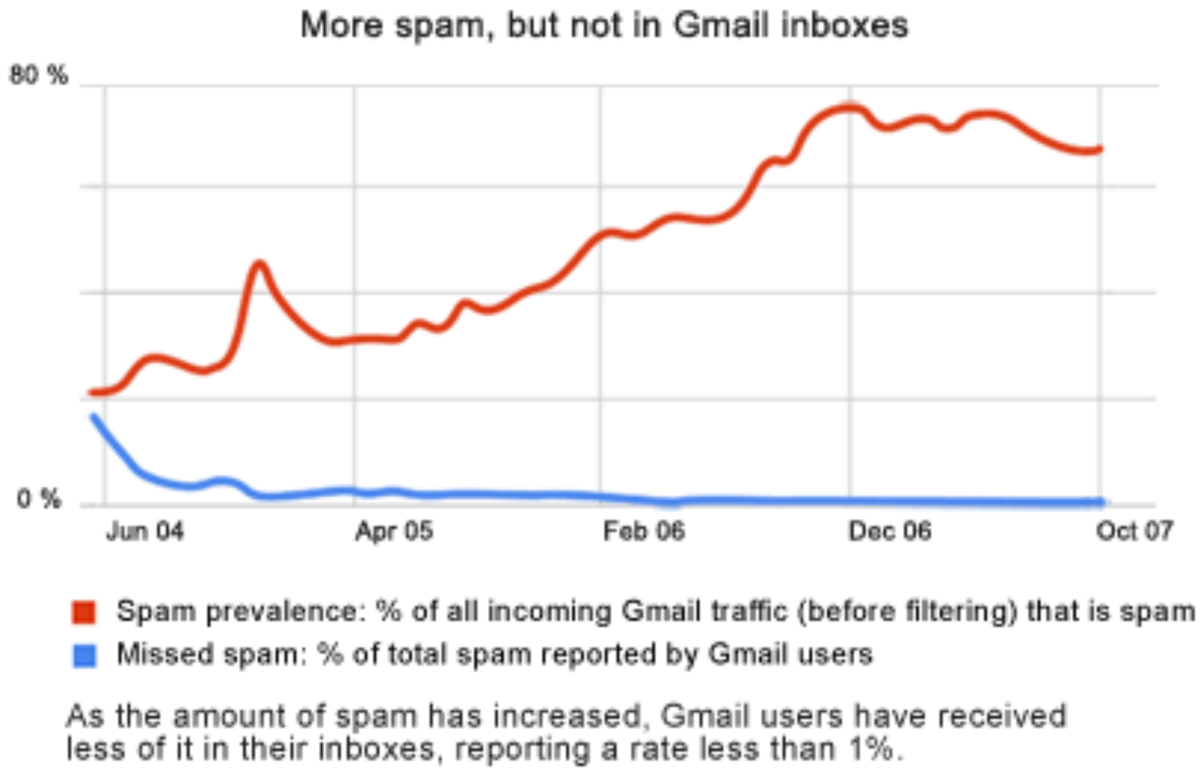
Abstract

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

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."

Source: **`http://www.google.com/mail/help/fightspam/spamexplained.html`**

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 $\mathbf{X} = \langle X_1, X_2, \ldots, X_m \rangle$
    - 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:
$\mathbf{X} = \langle X_1, X_2, \ldots, X_m \rangle$ noting for each word, if it appeared in email
    - Each email message is also marked as spam or not (value of Y)

Training the Classifier

- Given N training pairs:

  $(\mathbf{x}^{(1)},y^{(1)}), (\mathbf{x}^{(2)},y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$

- Learning
  - Estimate probabilities P(Y) and each $P(X_i \mid Y)$ for all i
    - Many words are likely to not appear at all in given set of email
  - Laplace estimate: $\hat{p}(X_i = 1 \mid Y = spam)_{Laplace} = \dfrac{(\#\,\text{spam emails with word } i) + 1}{\text{total } \#\,\text{spam emails} + 2}$
- Classification
  - For a new email, generate $\mathbf{X} = \langle X_1, X_2, \ldots, X_m \rangle$
  - Classify as spam or not using: $\hat{Y} = \arg\max_y \hat{P}(\boldsymbol{X} \mid Y)\hat{P}(Y)$
  - Employ Naive Bayes assumption: $\hat{P}(\boldsymbol{X} \mid Y) = \prod_{i=1}^{m} \hat{P}(X_i \mid Y)$

Training Naïve Bayes, is estimating parameters for a multinomial.

Thus it is just counting.

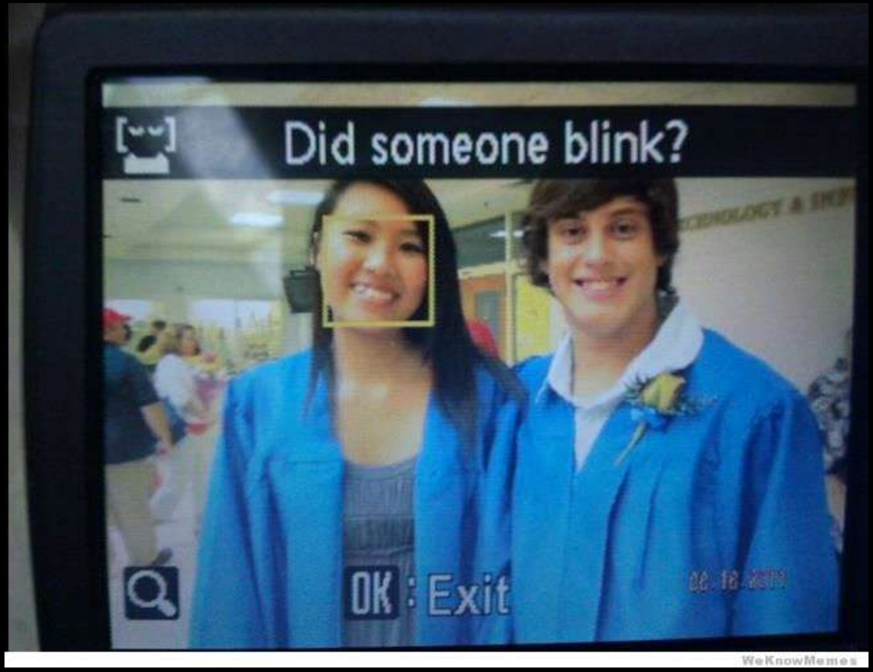
How Does This Do?

- 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%**

On biased datasets

Ethics and Datasets?



Sometimes machine learning feels universally unbiased.

We can even prove our estimators are "unbiased" ☺

Google/Nikon/HP had biased datasets

Ancestry dataset prediction

East Asian
or
Ad Mixed American (Native, European and African Americans)

Is the ancestry dataset biased?

Yes!

It is much easier
to write a binary classifier
when learning ML
for the first time

Learn Three Things From This

1. What classification with DNA Single Nucleotide Polymorphisms looks like.

2. That genetic ancestry paints a more realistic picture of how we are mixed in many nuanced ways.

3. The importance of choosing the right data to learn from. Your results will be as biased as your dataset.

Know it so you can beat it!

Ethics in Machine Learning
is a whole new field