Bayes

Machine Learning and Deep Learning
Lesson #4

AImage Lab
University of Modena and Reggio Emilia

Probability

- The world is a very uncertain place
- 30 years of Artificial Intelligence and Database research danced around this fact
- And then a few AI researchers decided to use some ideas from the eighteenth century

Discrete Random Variables

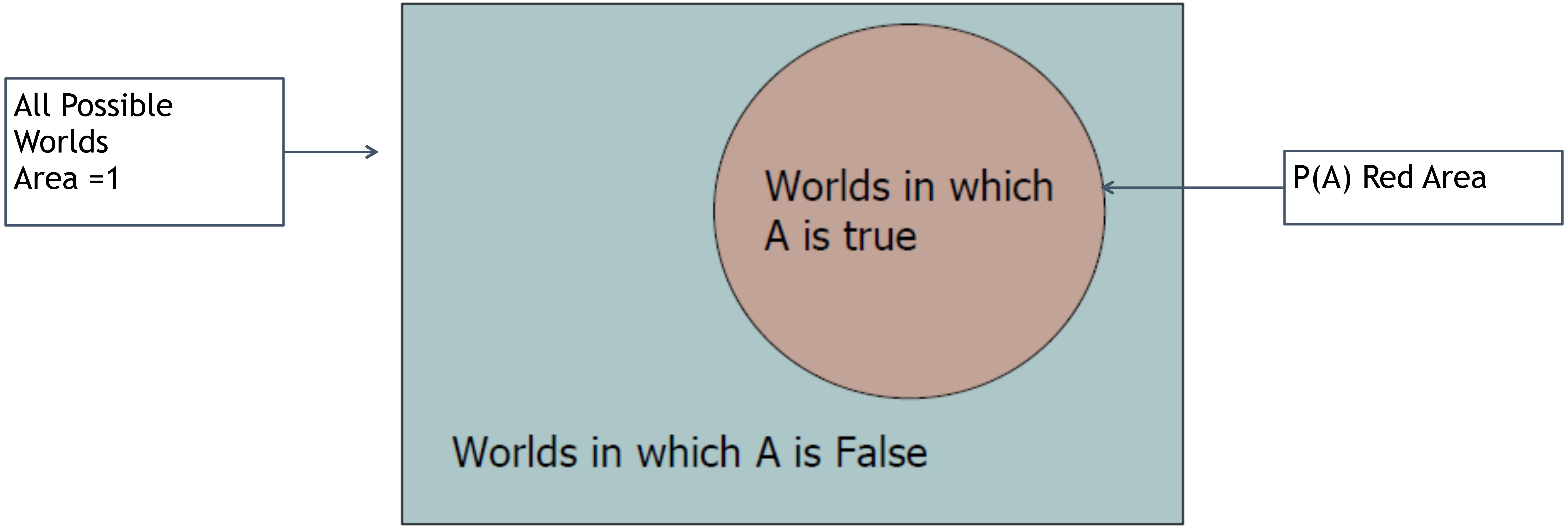
- A is a **Boolean-valued random variable** if A denotes an event, and there is some degree of uncertainty as to whether A occurs.

- Examples:
  - A = The US president in 2023 will be male
  - A = You wake up tomorrow with a headache
  - A = You have Ebola

Probabilities

- We write P(A) as “the fraction of possible worlds in which A is true”

- We could at this point spend 2 hours on the philosophy of this. But we won’t.



Multivalued Random Variables

- Suppose A can take on more than 2 values

- A is a random variable with *arity* k if it can

take on exactly one value out of $\{v_1, v_2, .. v_k\}$

Thus:

$$P(A = v_i \wedge A = v_j) = 0 \text{ if } i \neq j$$
$$P(A = v_1 \vee A = v_2 \vee A = v_k) = 1$$

Properties:

- *From the **Axioms of Probability** we can derive:*
- Sum Rule: $P(A = v_1 \vee A = v_2 \vee A = v_i) = \sum_{j=1}^{i} P(A = v_j)$
- Total Probability Rule: $\sum_{j=1}^{k} P(A = v_j) = 1$
- Thus: $P(B \wedge [A = v_1 \vee A = v_2 \vee A = v_i]) = \sum_{j=1}^{i} P(B \wedge A = v_j)$
- Discrete Marginalization over A: $P(B) = \sum_{j=1}^{k} P(B \wedge A = v_j)$

Conditional Probability

- Definition Conditional Probability:

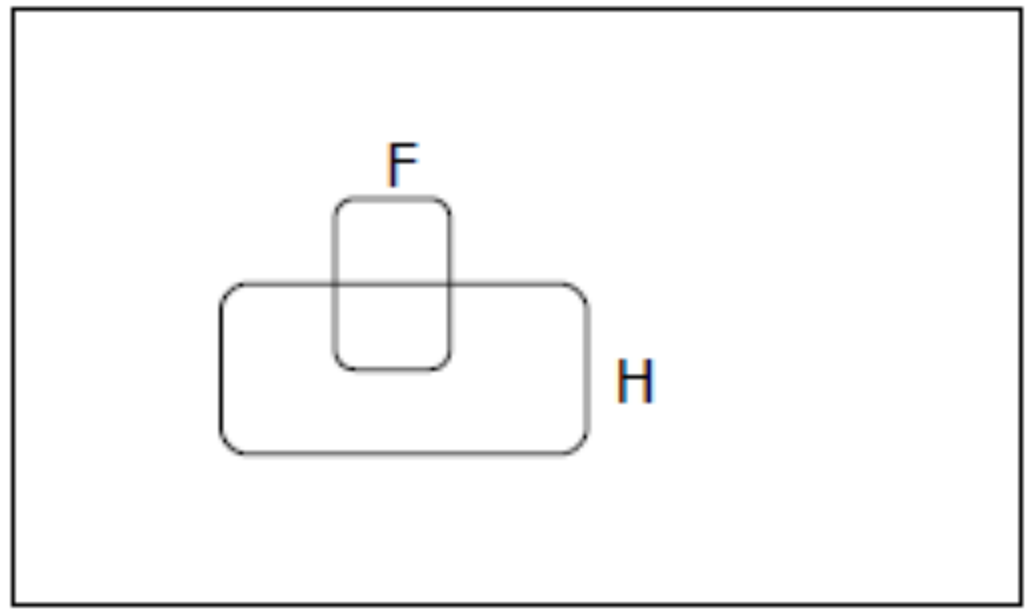
$$P(A \mid B) = \frac{P(A, B)}{P(B)}$$

- Corollary Chain Rule:

$$P(A, B) = P(A \mid B)\, P(B)$$

Probabilistic Inference Problem

One day you wake up with a headache. You think: “Drat! 50% of flus are associated with headaches so I must have a 50-50 chance of coming down with flu”

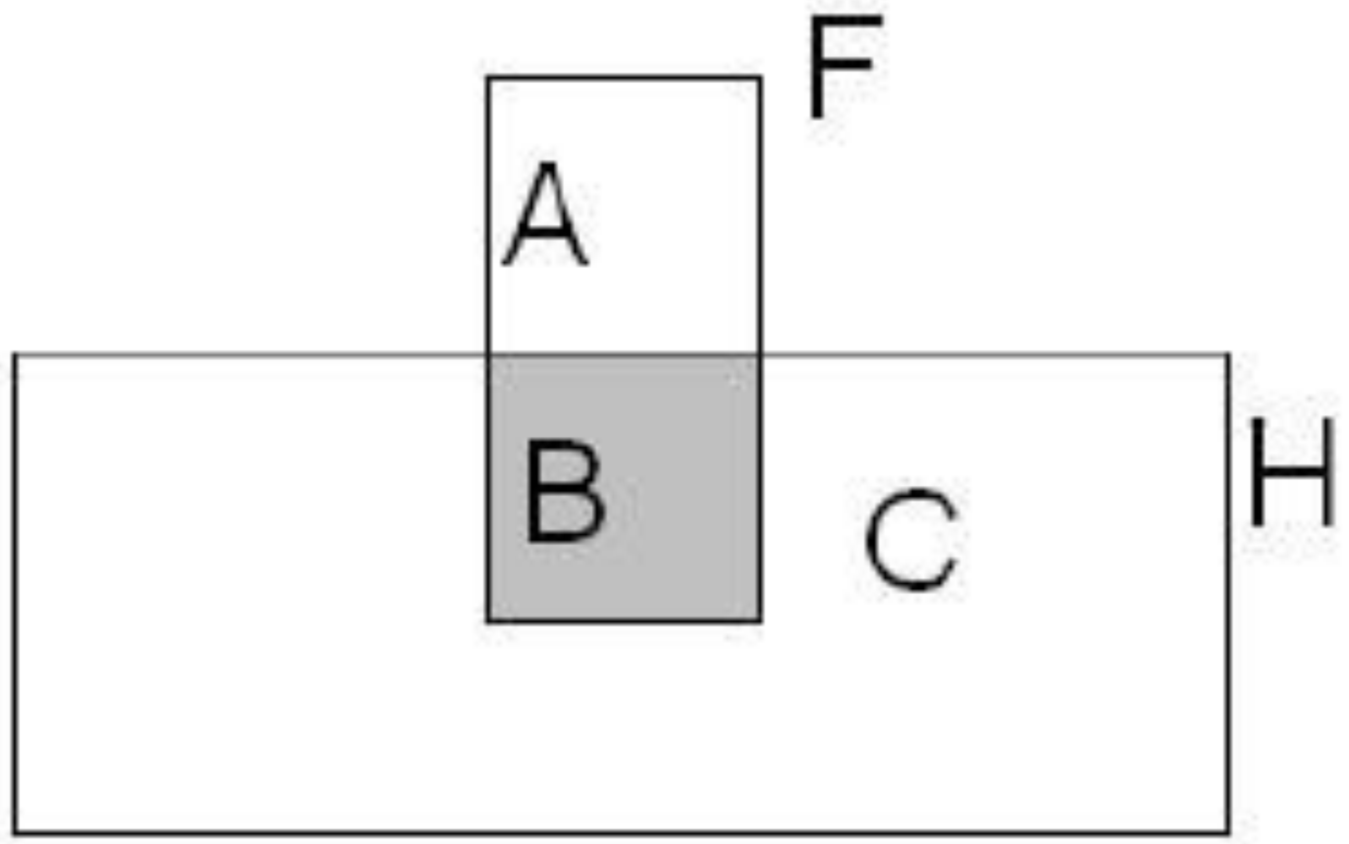


$P(H) = 1/10$
$P(F) = 1/40$
$P(H|F) = ½$

Is this reasoning good?

Geometric Interpretation

Thanks to Jahanzeb Sherwani for contributing this explanation:



Let's say we have P(F), P(H), and P(H|F), like in the example in class.

Areawise, $P(F) = A + B$, $P(H) = B + C$,

Also, $P(H|F) = \frac{B}{A + B}$

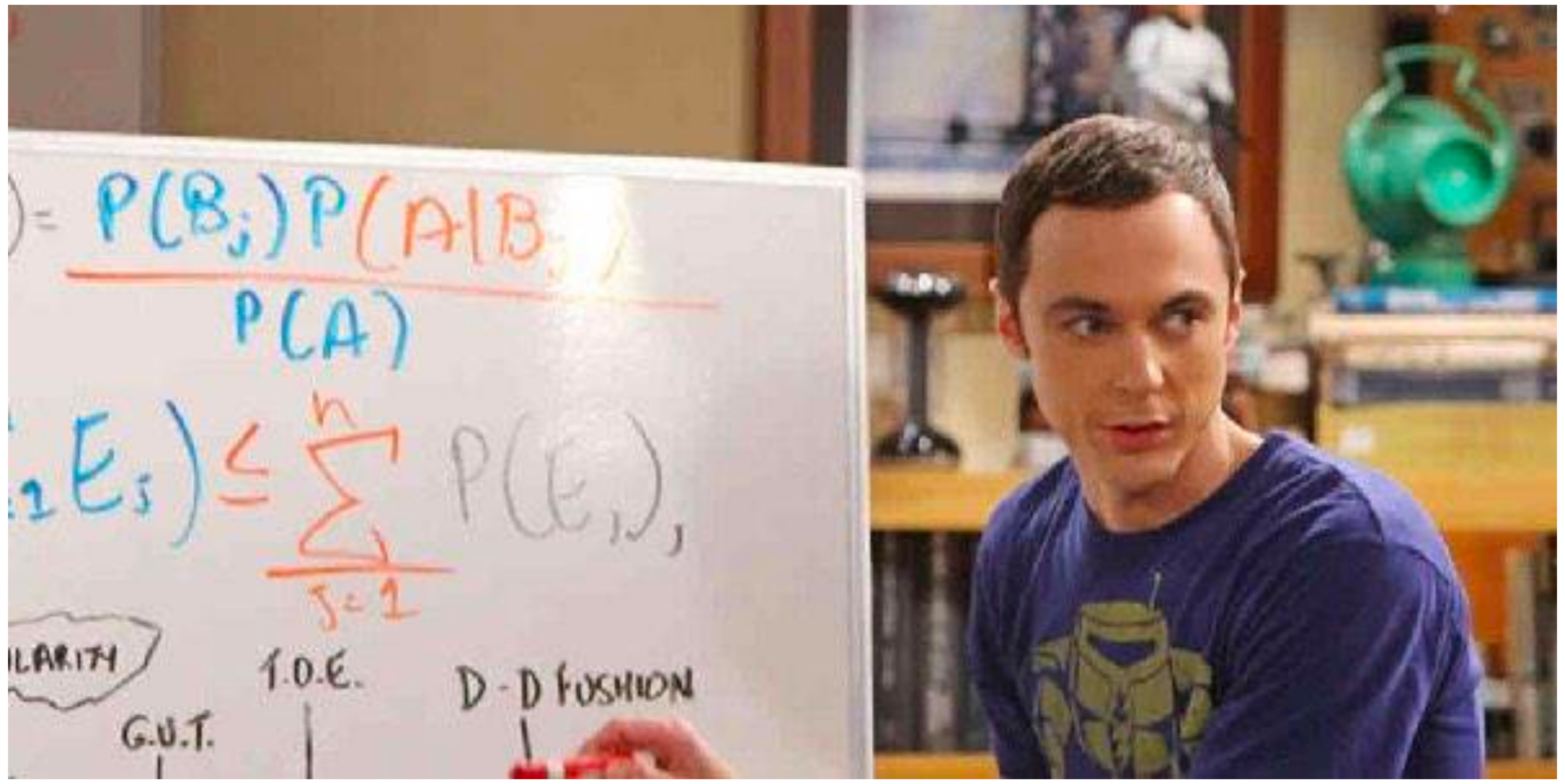
Thus, to get the opposite conditional probability, ie, P(F|H), we need to figure out $\frac{B}{B + C}$

Since we know B / (A+B), we can get B / (B+C) by multiplying by (A+B) and dividing by (B+C). But since we already calculated, A+B = P(F), and B+C = P(H), so we are actually multiplying by P(F) and dividing by P(H). Which is Bayes Rule:

$$P(F|H) = P(H|F) * \frac{P(F)}{P(H)}$$

The Bayes Rule

- What we did geometrically?



- The **Bayes Rule**

$$P(B\,|\,A) = \frac{P(A,B)}{P(B)} = \frac{P(A\,|\,B)\,P(B)}{P(A)}$$

Bayes, Thomas (1763) **An essay towards solving a problem in the doctrine of chances.** *Philosophical Transactions of the Royal Society of London, 53:370-418*

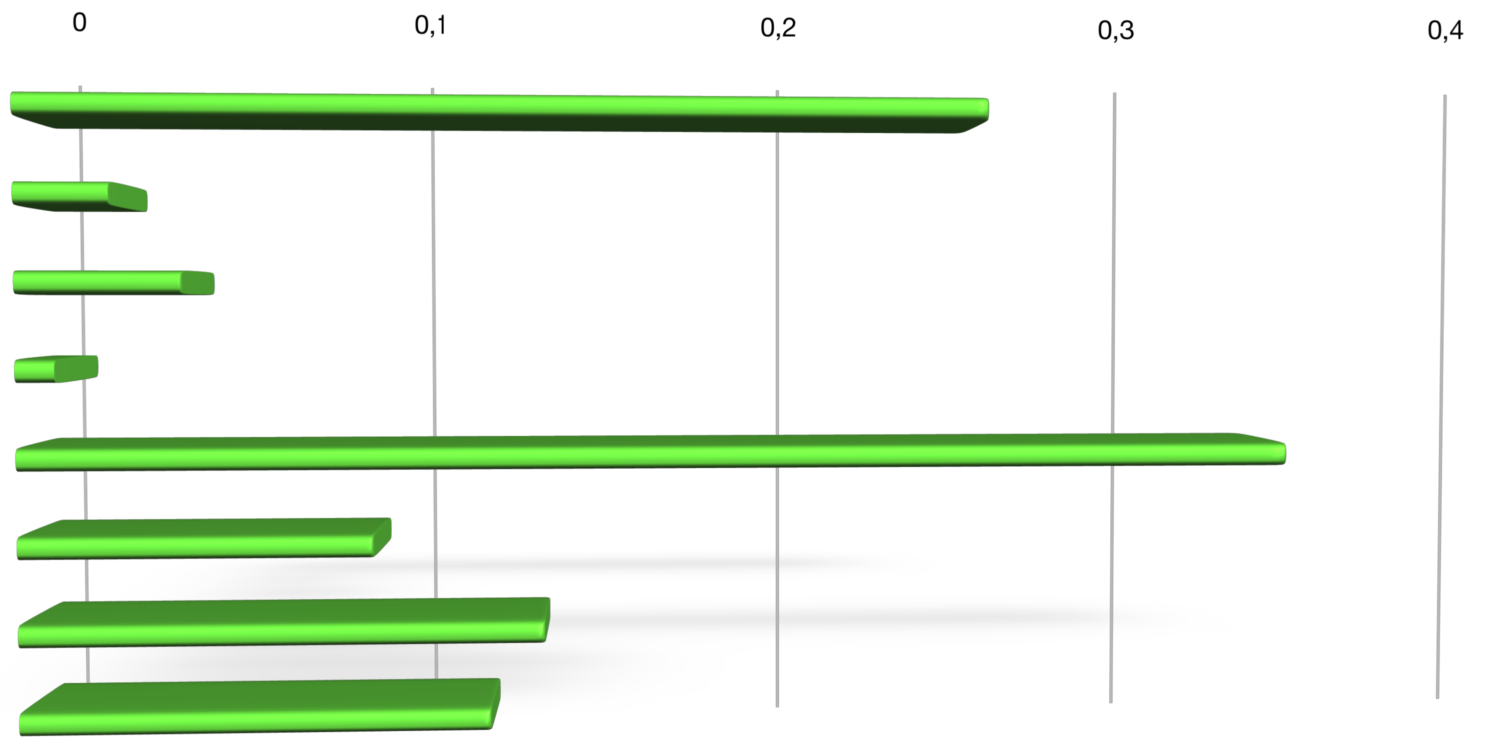
«The intuition of a reverend of XVIII century changed the modern world and yours!!»

Joint Probability

- Two multivalued Random Variables A and B

JD

Gender	Hours worked	Wealth	Prob
Female	< 40	Poor	0,25
		Rich	0,024
	>40	Poor	0,042
		Rich	0,01
Male	< 40	Poor	0,33
		Rich	0,09
	>40	Poor	0,134
		Rich	0,12



- **Inference**
  “get insight about the occurence of an Event from the JOINT”

- E.g. if I work <40 what is the probability I am poor

$$P(poor\,|<40) = \frac{P(poor, <40)}{P(<40)} = \frac{\sum_{Male,Female} P(sex, poor, <40)}{P(<40)} = approx\ 80\,\%$$

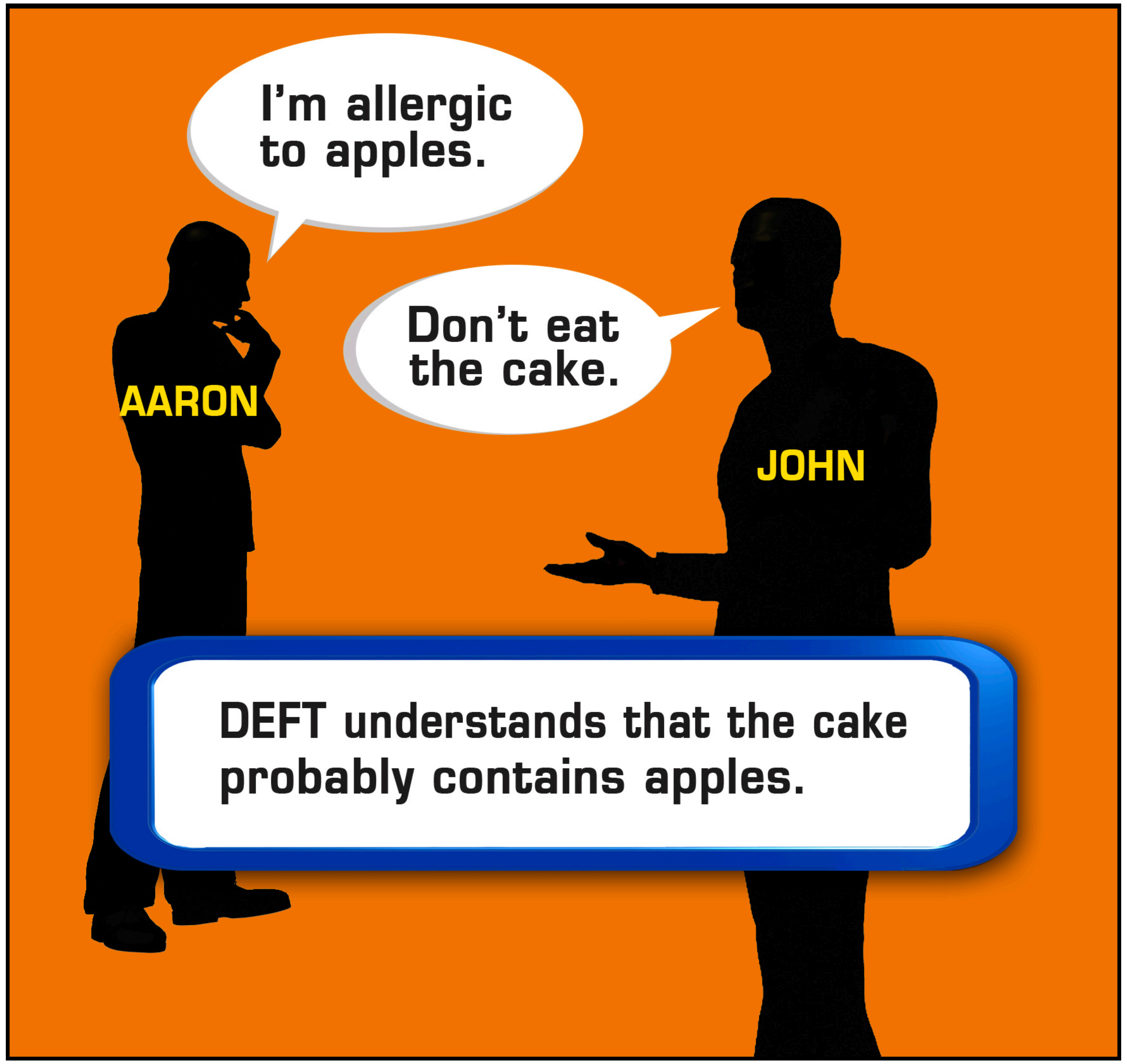
Inference is a big deal

- I’ve got this evidence.
  “What’s the chance that this conclusion is true?”

- I’ve got a sore neck: how likely am I to have meningitis?

- There’s a thriving set of industries growing based around Bayesian Inference.

  Highlights are:
  Medicine, Pharma, Help Desk Support, Engine Fault Diagnosis



How to compute Joint Probability

- **Idea One**: Expert Humans

- **Idea Two**: Simpler probabilistic facts and some algebra

Example: Suppose you knew

$P(A) = 0.7$	$P(C \mid A \wedge B) = 0.1$
	$P(C \mid A \wedge \sim B) = 0.8$
$P(B \mid A) = 0.2$	$P(C \mid \sim A \wedge B) = 0.3$
$P(B \mid \sim A) = 0.1$	$P(C \mid \sim A \wedge \sim B) = 0.1$

Then you can automatically compute the JD using the chain rule

$$P(A{=}x \wedge B{=}y \wedge C{=}z) = P(C{=}z \mid A{=}x \wedge B{=}y)\, P(B{=}y \mid A{=}x)\, P(A{=}x)$$

In another lecture: Bayes Nets, a systematic way to do this.

- **Idea Three**: Learn from Data

How to compute Joint Probability

- **Idea Two**: Simpler probabilistic facts and some algebra

Example: Suppose you know

P(A) = 0.7	P(C|A^B) = 0.1
	P(C|A^~B) = 0.8
P(B|A) = 0.2	P(C|~A^B) = 0.3
P(B|~A) = 0.1	P(C|~A^~B) = 0.1

Then you can automatically compute the JD using the chain rule

P(A=x ^ B=y ^ C=z) =
P(C=z|A=x^ B=y) P(B=y|A=x) P(A=x)

In another lecture: Bayes Nets, a systematic way to do this.

- **Idea Three**: Learn from Data

How to compute Joint Probability

- **Idea Three**: Learn from Data

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

How to compute Joint Probability

- **Idea Three**: Learn from Data

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

How to compute Joint Probability

- **Idea Three**: Learn from Data

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

How to compute Joint Probability

- **Idea Three**: Learn from Data

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

How to compute Joint Probability

- **Idea Three**: Learn from Data

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

How to compute Joint Probability

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

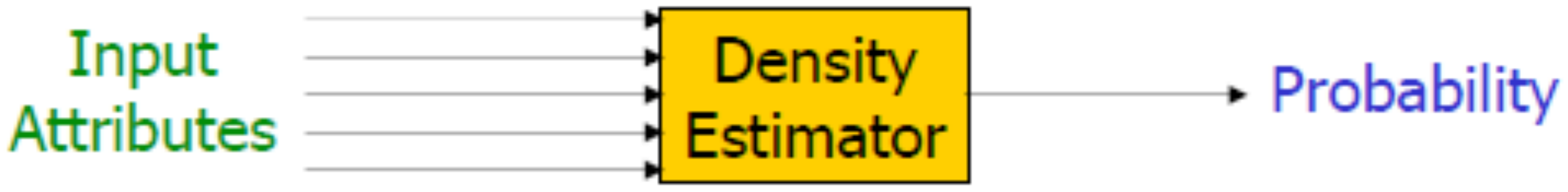
A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

Density Estimation

- Our Joint Distribution learner is our first example of something called Density Estimation

- A Density Estimator learns a mapping from a set of attributes to a Probability



- Density estimation can be:
    - Observing variables values: Discrete/Continuous
    - Observing probability equation: Parametric/Non Parametric

Density Estimation Evaluation

- Given a record x, a density estimator M can tell you how likely the record is

$$\hat{P}(\mathbf{x}|M)$$

- Given a dataset with R records the DE can tell you how likely the dataset is
    - (assuming data independently generated from DE JD)

$$\hat{P}(\text{dataset}|M) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

- Since probabilities of datasets get so small we usually use **log** probabilities

$$\log \hat{P}(\text{dataset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

Density Estimators Pros

- We have a way to learn a Density Estimator from data.

- Density estimators can do many good Things:

  - Can sort the records by probability, and thus spot weird records (anomaly detection)

  - Can do inference: P(E1|E2) (Automatic Doctor / Help Desk etc)

Density Estimators Pros

- We have a way to learn a Density Estimator from data.

- Density estimators can do many good Things:

  - Can sort the records by probability, and thus spot weird records (anomaly detection)

  - Can do inference: P(E1|E2) (Automatic Doctor / Help Desk etc)

BUT

Density estimation by directly learning the joint is trivial, mindless and dangerous

Overfitting

If this ever happens, it means there are certain combinations that we learn are impossible

mpg	modelyear	maker	
bad	70to74	america	0.27551
		asia	0.0255102
		europe	0.0153061
	75to77	america	0.153061
		asia	0.0255102
		europe	0.0357143
	78to83	america	0.0561224
		asia	Never
		europe	Never
good	70to74	america	0.0102041

$$\log \hat{P}(\text{testset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

$$= -\infty \text{ if for any } k \; \hat{P}(\mathbf{x}_k|M) = 0$$

Overfitting

If this ever happens, it means there are certain combinations that we learn are impossible

mpg	modelyear	maker	
bad	70to74	america	0.27551
		asia	0.0255102
		europe	0.0153061
	75to77	america	0.153061
		asia	0.0255102
		europe	0.0357143
	78to83	america	0.0561224
		asia	Never
		europe	Never
good	70to74	america	0.0102041

$$\log \hat{P}(\text{testset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

$$= -\infty \text{ if for any } k \ \hat{P}(\mathbf{x}_k|M) = 0$$

We need Density Estimators that are less prone to overfitting

Overfitting

Naive Density Estimator

- The problem with the Joint Estimator is that it just mirrors the training data.

- We need something which **generalizes** more usefully.

The **naïve model** generalizes strongly:

"Assume that each attribute is distributed independently of any of the other attributes."

IID Independently Distributed Data

- Let x[i] denote the i-th field of record x.

- The independently distributed assumption says that:

  for any i, *v*, u1, u2… ui-1, ui+1… Um

- x[i] is independent of {x[1],x[2],..x[i-1], x[i+1],…x[M]}

$$x[i] \perp \{x[1], x[2], \ldots x[i-1], x[i+1], \ldots x[M]\}$$

Independence Theorems

- Given A and B random variables
- A is independent of B «if and only if» $P(A|B)=P(A)$

Consequences:

- $P(A,B)=P(A)P(B)$
- $P(B|A)=P(B)$
- $P(\sim A|B)=P(\sim A)$
- $P(A|\sim B)=P(A)$

Naive DE General Case

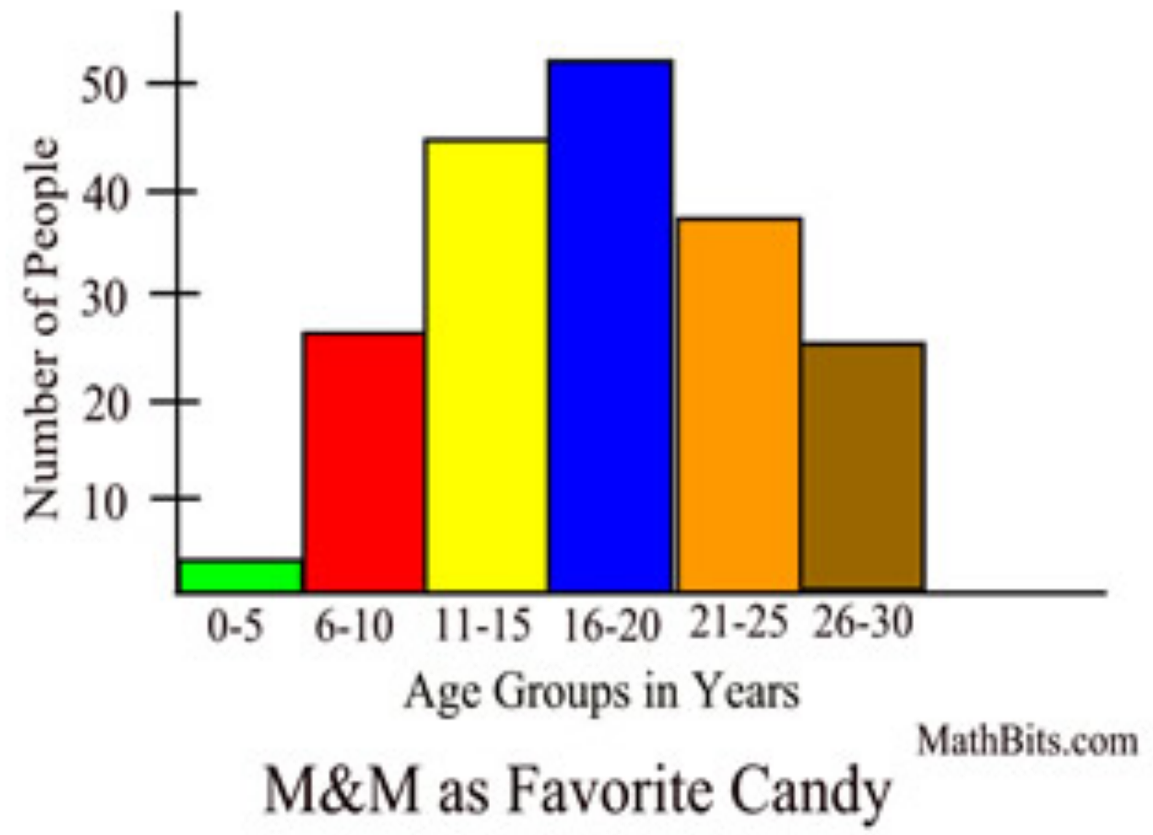
- Suppose x[1], x[2], … x[M] are independently distributed.

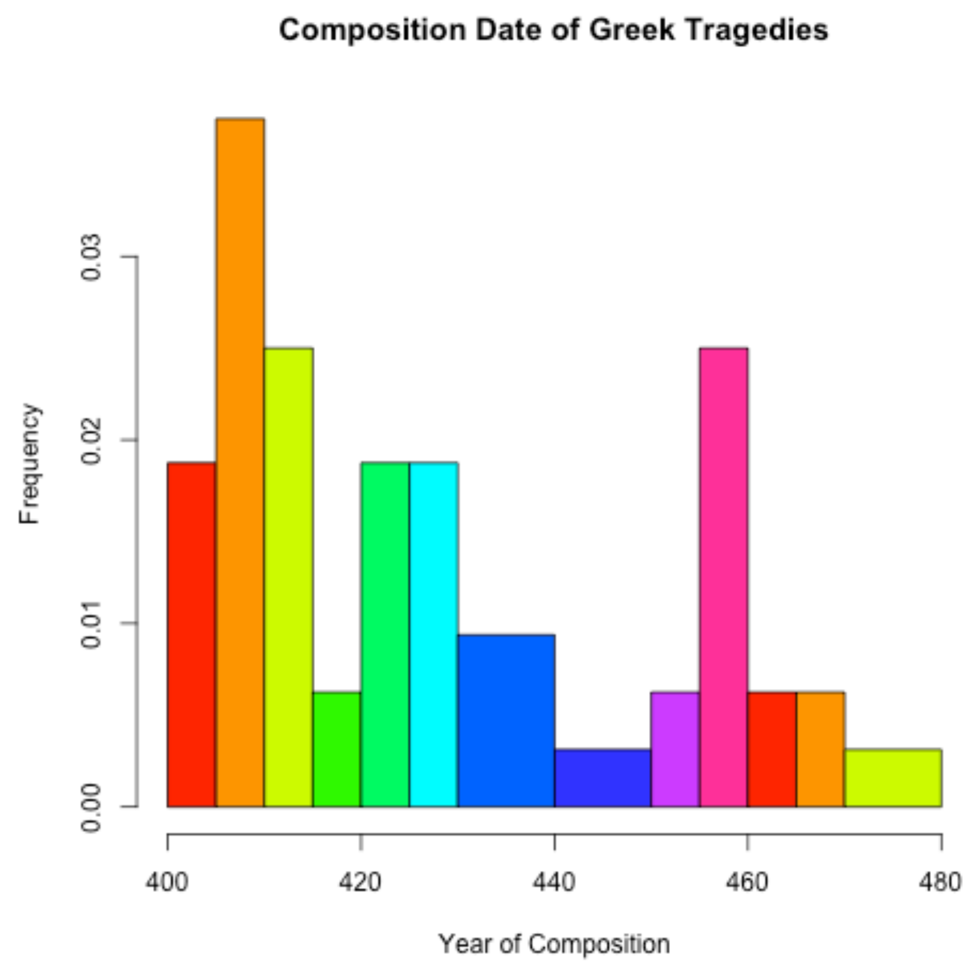
$$P(x[1]=u_1, x[2]=u_2, \ldots x[M]=u_M) = \prod_{k=1}^{M} P(x[k]=u_k)$$

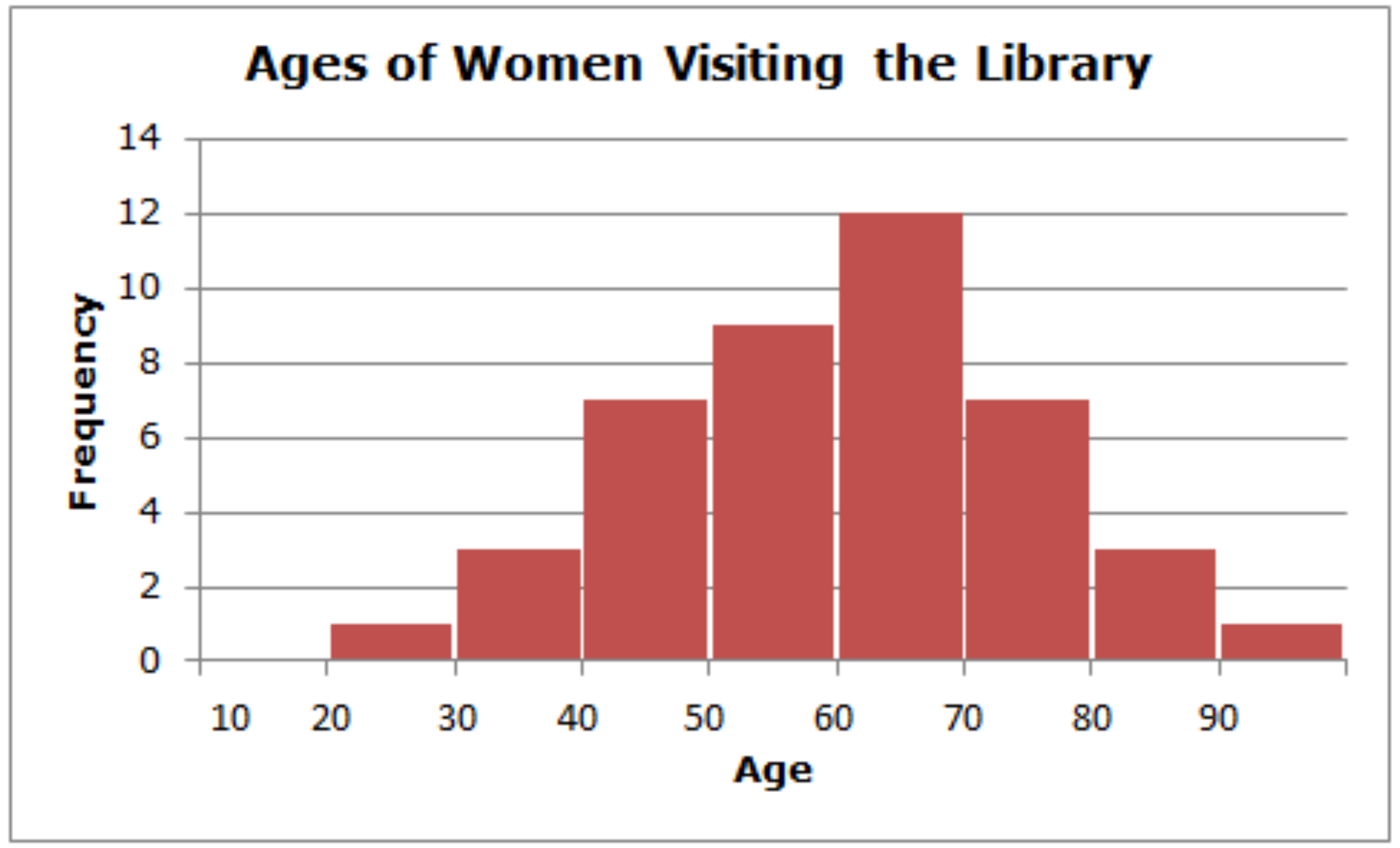
But How do we learn a naïve density estimator:

$$\hat{P}(x[i]=u) = \frac{\#\text{records in which } x[i]=u}{\text{total number of records}}$$

- Normalized Histogram is a discrete Non Parametric DE







Bayes Classifier

Build a Bayes Classifier (Preliminary Step)

1. Assume you want to predict output Y which has arity nY and values v_1, v_2, … v_{ny}

3. Assume there are m input attributes called X_1, X_2, … X_m

5. Break dataset into nY smaller datasets called DS_1, DS_2, … Ds_{ny}

7. Define DS_i = Records in which $Y=v_i$

9. For each DS_i learn Density Estimator M_i to model the input distribution among the $Y=v_i$ records.

Build a Bayes Classifier (Preliminary Step)

1. Assume you want to predict output Y which has arity nY and values $v_1, v_2, \dots v_{ny}$

3. Assume there are m input attributes called $X_1, X_2, \dots X_m$

5. Break dataset into nY smaller datasets called $DS_1, DS_2, \dots Ds_{ny}$

7. Define DS_i = Records in which $Y=v_i$

9. For each DS_i learn Density Estimator M_i to model the input distribution among the $Y=v_i$ records.

M_i estimates

$$P(X_1, X_2, \dots X_m \mid Y=v_i)$$

ML Classifier

- Idea: When a new set of input values (X1= u1, X2= u2, …. Xm= um) come along to be evaluated predict the value of Y that makes P(X1, X2, …Xm | Y=vi) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

ML Classifier

- Idea: When a new set of input values (X1= u1, X2= u2, …. Xm= um) come along to be evaluated predict the value of Y that makes P(X1, X2, …Xm | Y=vi) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

Is this a good idea?

ML Classifier

- Idea: When a new set of input values (X1= u1, X2= u2, …. Xm= um) come along to be evaluated predict the value of Y that makes P(X1, X2, …Xm | Y=vi) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

Is this a good idea?

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

This is a **Maximum Likelyhood Classifier**

Cons:

- Not Bayesian
- Silly if some Y_i are unlikely

Build a Bayes Classifier

- Much Better Idea!!!:

- When a new set of input values ($X_1= u_1$, $X_2= u_2$, …. $X_m= u_m$) come along to be evaluated predict the value of Y that makes most likely

$P(Y=v_i|X_1, X_2, \ldots X_m)$

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} P(Y=v \mid X_1 = u_1 \cdots X_m = u_m)$$

We can get the posterior using Bayes Rule

$$P(Y=v \mid X_1 = u_1 \cdots X_m = u_m)$$

$$= \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y=v) P(Y=v)}{P(X_1 = u_1 \cdots X_m = u_m)}$$

$$= \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y=v) P(Y=v)}{\sum_{j=1}^{n_Y} P(X_1 = u_1 \cdots X_m = u_m \mid Y=v_j) P(Y=v_j)}$$

Naive Version Bayes Classifiers

- Hypothize X are **independent** and use product rule to build the joint DE

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v)\prod_{j=1}^{n_Y} P(X_j = u_j \mid Y = v)$$

- Technical Hint:If you have 10,000 input attributes that product will underflow in floating point math. You should use **logs**.

Naive Version Bayes Classifiers

- Hypothize X are **independent** and use product rule to build the joint DE

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} P(Y = v) \prod_{j=1}^{n_Y} P(X_j = u_j \mid Y = v)$$

- Technical Hint:If you have 10,000 input attributes that product will underflow in floating point math. You should use **logs**.

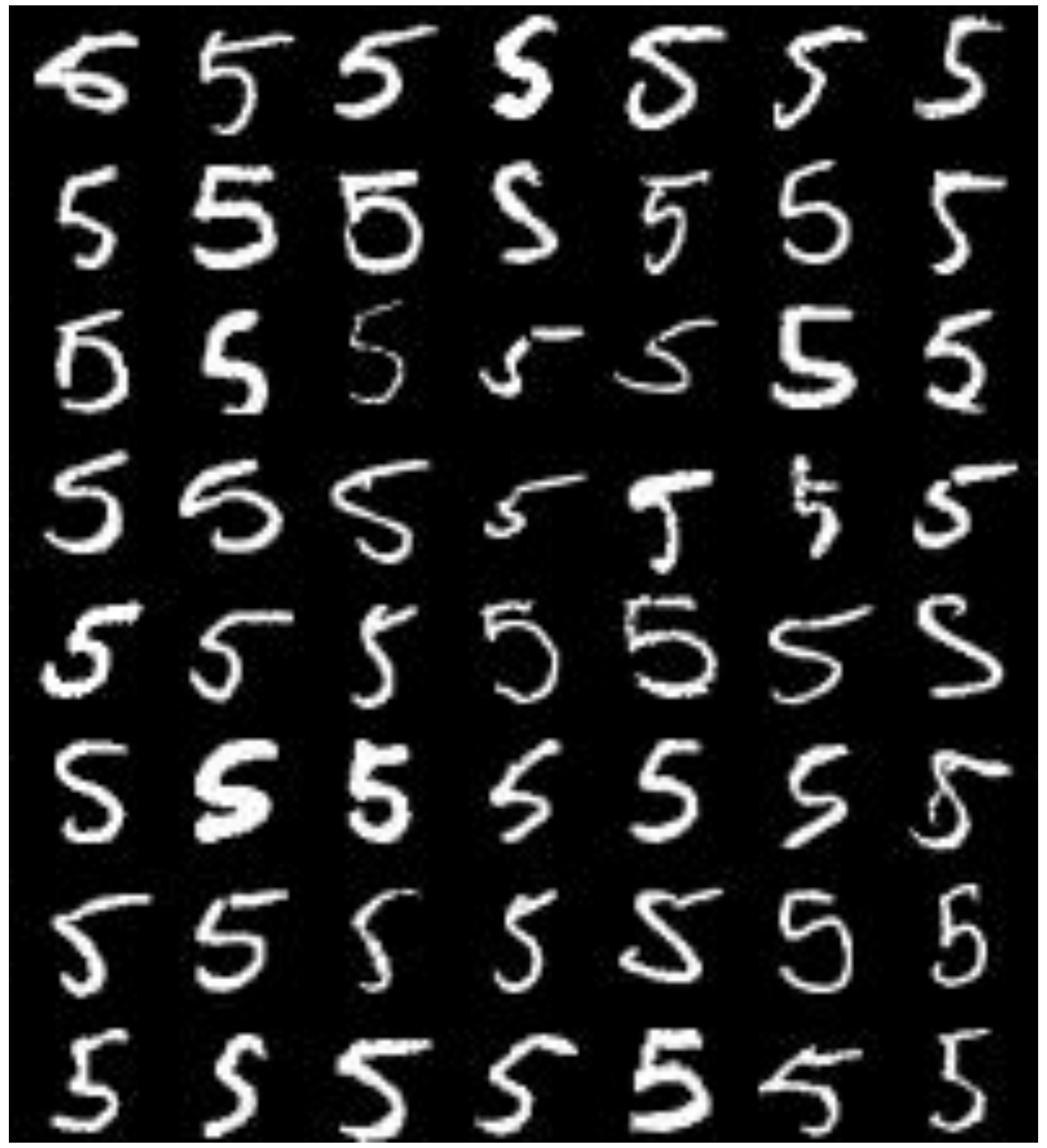
$$\boxed{Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} \left(\log P(Y = v) + \sum_{j=1}^{n_Y} \log P(X_j = u_j \mid Y = v) \right)}$$

Example Digit Recognition

Naïve Bayes Training

- Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data:

MNIST Training Data



Naïve Bayes Training

- Training in Naïve Bayes is easy:
  - Estimate P(Y=v) as the fraction of records with Y=v
  - Estimate P(X_i=u|Y=v) as the fraction of records with Y=v for which X_i=u

$$P(Y = v) = \frac{Count(Y = v)}{\#\ records}$$

- (This corresponds to Maximum Likelihood estimation of model parameters)

$$P(X_i = u | Y = v) = \frac{Count(X_i = u \wedge Y = v)}{Count(Y = v)}$$

Naïve Bayes Training

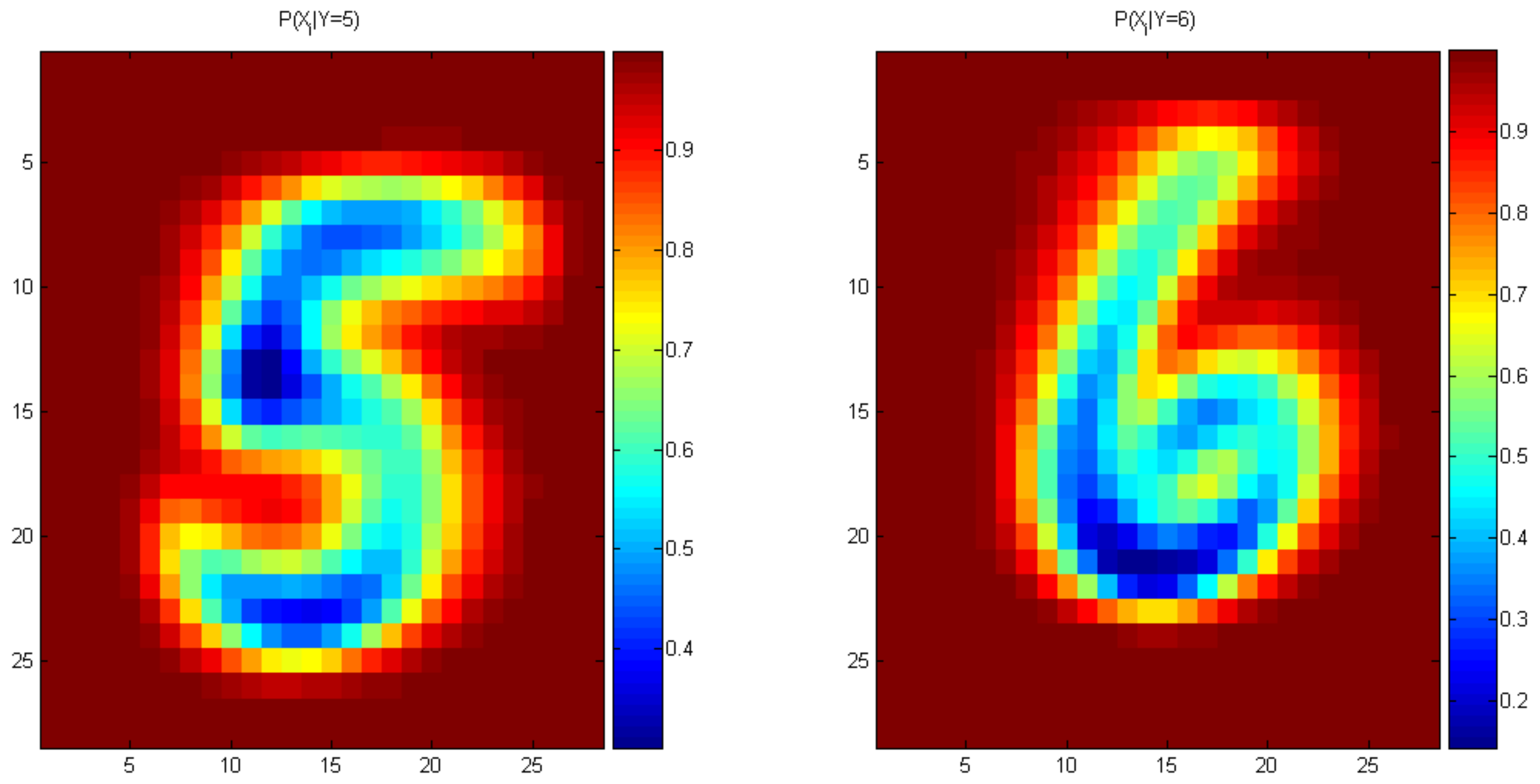
- In practice, some of these counts can be zero
- Fix this by adding "virtual" counts:

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v) + 1}{Count(Y = v) + 2}$$

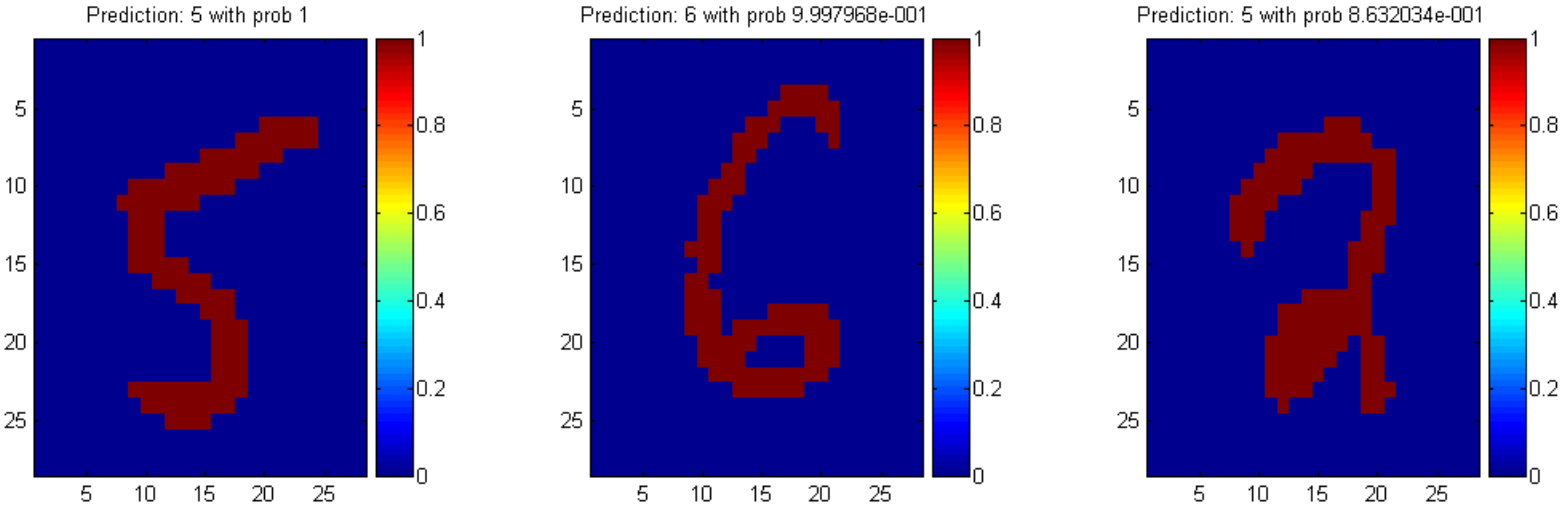
  - (This is like putting a prior on parameters and doing MAP estimation instead of MLE)
  - This is called *Smoothing*

Naïve Bayes Training

- For binary digits, training amounts to averaging all of the training fives together and all of the training sixes together.



Naïve Bayes Classification



Performance on a Test Set

- Naïve Bayes is often a good choice if you don't have much training data!