6.034 Lecture

11/28

Probabilistic Inference 1

□ intrinsicist - Frequentist - intuitionist

□ joint probability table

↓

□ Belief nets

↓

□ Model Selection + Structure Discovery

* It's all about Models
Representations
Constraint

* Probability is a saftey net

---

probabilistic methods taking over computer theory
last 10 years

New thing outside student center: hack or art

Can look back at past few years to guide our knowledge.

Or guess ~~its~~ that you think reflect admin's attitudes

Produce joint probability table

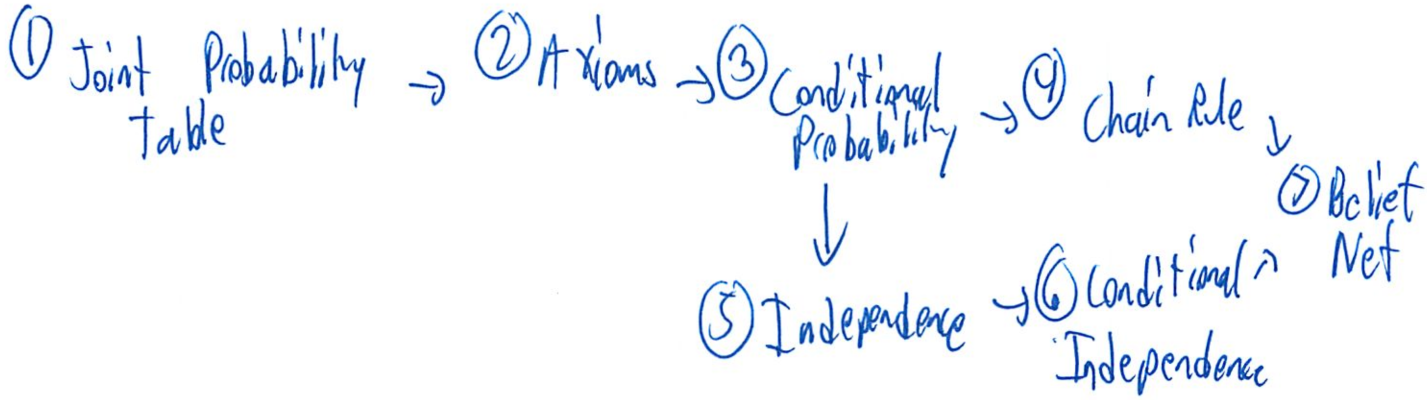
For each possible combos of charastics - tally up event's that fit that set of charaistict

divide tally by total to get probability

But # rows grow exponentially

└ np-problem

Roadmap

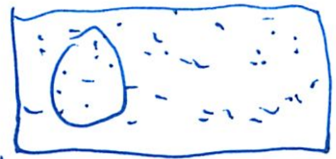


① Joint Probability table → ② Axioms → ③ Conditional Probability → ④ Chain Rule → ⑦ Belief Net

③ Conditional Probability → ⑤ Independence → ⑥ Conditional Independence → ⑦ Belief Net

Axioms

Ⓐ $0 \le P(a) \le 1.0$



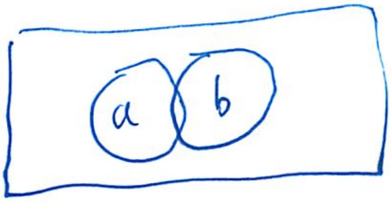
← dots filled in square randomly

P(a) proportional to $\frac{\text{size circle}}{\text{size square}}$

$P(\text{always occurs}) = 1.0$

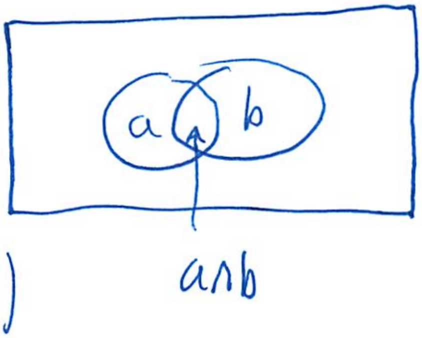
$P(\text{never occurs}) = 0.0$

ⓒ $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$



Conditional Probability

$$P(a \underset{\text{given}}{|} b) = \frac{P(a \wedge b)}{P(b)}$$



$$P(a|b)\,P(b) = P(a \wedge b)$$
$$= P(b|a)\,P(a)$$

Chain Rule

$$P(a \cap \underbrace{b \cap c}_{\text{treat as it if were 1 thing}}) = P(a \mid b \cap c)\, P(b \cap c)$$

$$= P(a \mid b \cap c)\, P(b \mid c)\, P(c)$$

So general form

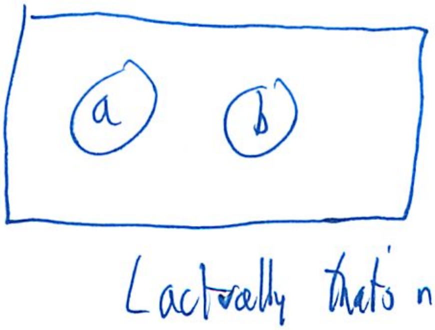
$$P(x_1, \ldots\ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i+1}, \ldots\ldots, x_n)$$

↑ each multiplier has fewer + fewer things its conditional on

---

Independence

$$P(a \mid b) = P(a) \quad \text{when independent}$$



↳ actually that's not strictly independent

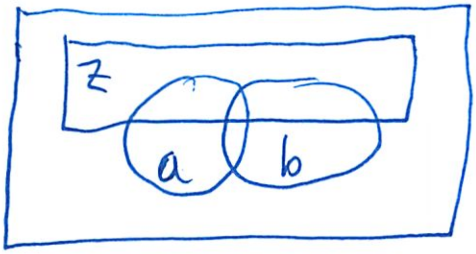
$P(a \cap b) = P(a \mid b)\, P(b)$; when not independent

$P(a \cap b) = P(a)\, P(b)$; when independent

↳ $P(b \mid a) = P(b)$

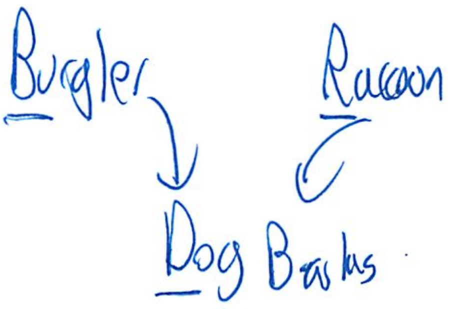
Conditional Independence

$$P(a \mid b \cap z) = P(a \mid z)$$



Can do dog bark tree

- only look at where dog barked
- Then add up remaining rows where burgler is to see prob that it is burgle
- if you had more details ie racoon not present it would help make table more exact



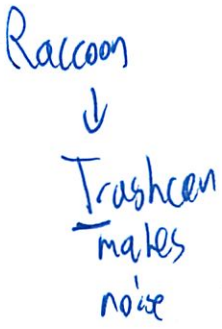
4 possibilities:

B	R	P(D)
T	T	0.75
T	F	0.5
F	T	0.1
F	F	0.01 ← dog barks just because

P(B) = prob that burgler is around tonight = 0.1
↳ a proiri

P(R) = 0.5

Can add to this



R	P(T)
F	0.1 ← makes noise on own
T	0.9

We might call the <u>Police</u> if the dog barks

<u>Dog</u> Barks
↳ <u>Call</u> Police

D	P(C)	
F	.01	← call the police anyway
T	.2	

<u>Better, more complicated way to think abat table</u>

Prob is conditionally independent of all non decendents given parents

ie Dog Barks is conditionally ind of Trash
given Burgler + Raccoon (parents)
So Trash does not matter for dog barks

What is Prob of everything together

$P(C \cap D \cap T \cap B \cap R) =$

↓

$$\downarrow = P(C|D\cap T\cap B\cap R)\,P(D|T\cap B\cap R)\,P(T|B\cap R)\,P(B|R)\,P(R)$$

but some variables conditionally ind. of each other

$$= P(C|D\cap \cancel{T}\cap \cancel{B}\cap \cancel{R})\,P(D|\cancel{T}\cap B\cap R)\,P(T|\cancel{B}\cap R)\,P(B|\cancel{R})\,P(R)$$

$$= P(C|D)\,P(D|\underbrace{B\cap R}_{\text{has 2 columns}})\,P(T|R)\,\underbrace{P(B)\,P(R)}_{\text{a proiri probabilites}}$$

only has this colum

= take probs from our belief map we drew

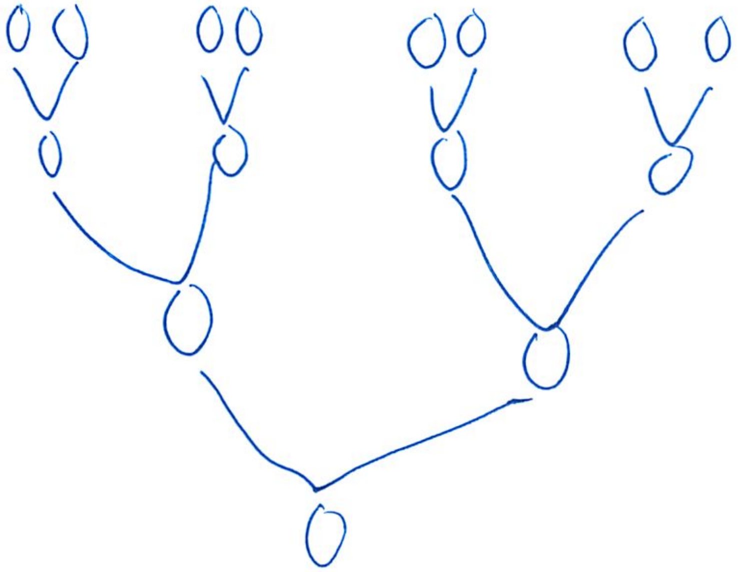
if put it all in a joint prob table we would use a lot more #s

10 #s vs $2^N = 2^5 = 32$ #s

So if P = max parents

$N \cdot 2^P$ is upper bound

When does this matter?



8 + 16 + 8 + 4 vs 2^{15}

36 vs 32,000

↑much eaiser to deal with

Q * Since we had our constraint about which variables inflence one another

---

The more examples you look at the closer you are to actual

(10)

Gold Star Ideas

(see 1st pg)

Probabilistic model does not have constraints

Saffey net if <u>don't</u> have other models

But it's more of a ~~sad~~ seductive saffey net

Floating

What floats?

Plastic - some things float - some ~~that~~ sinks

But its really just density

Prob is ~~the~~ saffey net if we <u>don't know</u> density

But ~~[illegible]~~ figuring out we care about density
would be better

6.031 Tutorial

4/28

(Skipped due to MITCET meeting)

6.034 Lecture

11/30

Probabilistic Influence 2

□ Calculating/Reconstructing JPT
  Simulating / Acquiring

□ De Obfuscation

□ The Reverend Bayes + Native Bayes

□, Model, Selection + Discovery

Program that discovers revenge
  ham → harm

Lots of qu last lecture

  JPT - table ~~a~~ of all possible values
    Can calc prob of each row
    Then add rows to get prob of some event
    Can restrict universe (using conditional prob)

    Once we have JPT can do anyting
    But might get too big

②

Then we get that table/map thing

Not as flexible as JPT since made assumptions

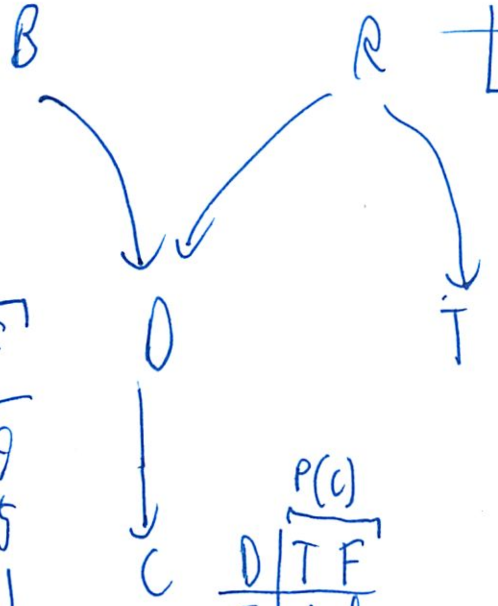
-such as what depends on what

L constraint/model

But dramatically ↓ # of #s you need

T	F
.1	.9

T	F
.4	.6



B	R	P(D) T	F
F	F	.1	.9
F	T	.5	.5
T	F	.9	.1
T	T	1	0

D	P(C) T	F
F	.1	.9
T	.6	.4

R	P(trash) T	F
F	.1	.9
T	.6	.4

$$P(\underset{\overset{\uparrow}{F}}{C},\underset{\overset{\uparrow}{T}}{D},\underset{\overset{\uparrow}{F}}{T},\underset{\overset{\uparrow}{T}}{B},\underset{\overset{\uparrow}{F}}{R})$$

what we mean

$$= P(\underset{\overset{\uparrow}{F}}{C}\mid\underset{\overset{\uparrow}{T}}{D})\,P(\underset{\overset{\uparrow}{T}}{D}\mid\underset{\overset{\uparrow}{T}}{B},\underset{\overset{\uparrow}{F}}{R})\,P(\underset{\overset{\uparrow}{F}}{T}\mid\underset{\overset{\uparrow}{F}}{R})\,P(\underset{\overset{\uparrow}{T}}{B})\,P(\underset{\overset{\uparrow}{F}}{R})$$

3

(table must be loop-free, must be some parent indep. Variables)

Now fill in values

So for $P(D \mid B,R)$ look at the table which has values for B,R

$= .4 \cdot .9 \cdot .4 \cdot .1 \cdot .6$

Now how to reconstruct table?

Since we have each combination

└ simulate every possible combination

Its essentially a bias coin flipping for B

with

$P(B=T) = .1$

$P(B=F) = .9$

Same for R

So say we got

$B=F$

$R=T$

Now look at that row in colmn for D

		T	F
F	T	.5	.5

← now flip w/ prob $P(D=T) = .5$

$P(D=F) = .5$

etc for rest of table

Then do this lots of times

As do it more your prob. get close to tables' value

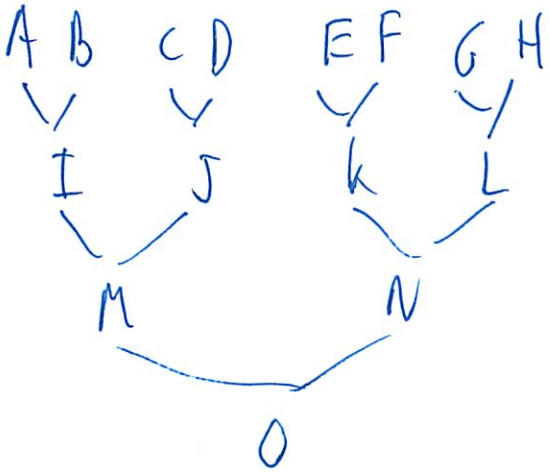
But in nature don't have table - instead observe real life

Simulating off table is kinda pointless

Will be $2^5 = 32$ rows on the table

Any computer can do

Can be more complex - could be a family tree - do you get a disease?



Infrence table

P_{max} = max # of parents for each

M = max # of entries in table $\le$ ~~[illegible]~~ $2^{P_{max}}$ upper bound ~~# of rows~~

~~#~~ $4 \le 2^2$

$n =$ # of variables

$n \cdot m = n 2^{P_{max}}$

↑ # of rows

$15 \cdot 2^2 = 60$

So much better to have an inference table

← Plus not all have parents actually 16

JPT $2^{15} = 3200$

So if doing genetic counsling

- 5 variables
  - lab tests
  - geno/pheno type
  - etc

Talk to your relatives

- max 40

So $40 \times 5 = 200$ variables

JPT would be 2^{200}

way too hard!

Inference net $200 \cdot 2^2 = 800$

much better!

If can't do JPT - can use statistical sampling methods

$$P(a|b) = \frac{P(a,b)}{P(b)} \Rightarrow P(a|b)\,P(b) = P(a,b) = P(b|a)\,P(a)$$

$$P(b|a) = \frac{P(a,b)\;P(b)}{P(a)}$$

not sure if right

What ~~ya~~ can you use this for?

If you have a fake + real coin
biases fair

Want to say which one

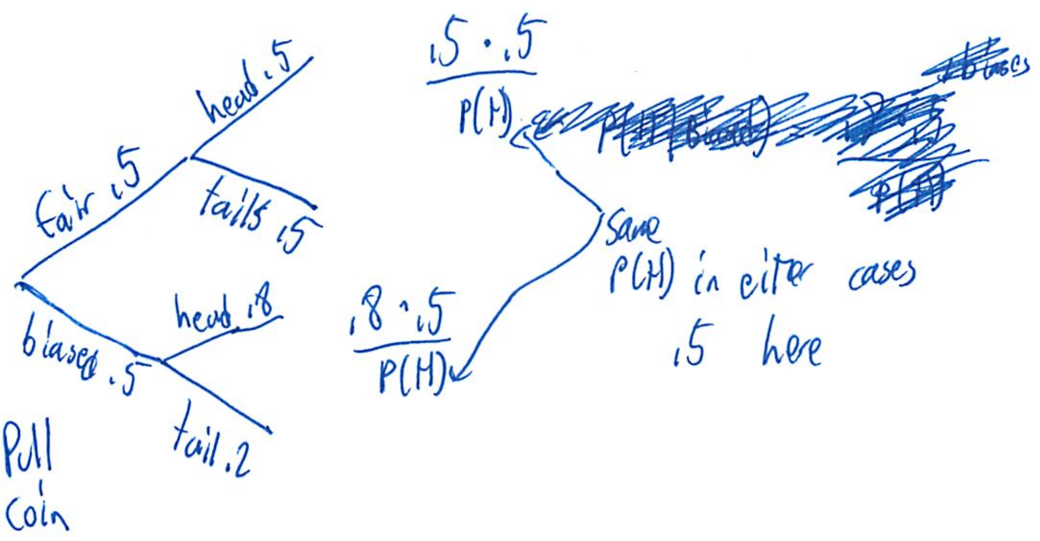
↓ a priori will get biased coin

$$P(\text{class}\,|\,\text{evidence}) = \frac{P(\text{evidence}\,|\,\text{class})\;P(\text{class})}{P(\text{evidence})}$$

7

Could write as inferce map

Or write like this



So

.5

.8

$$P(c \mid e_1, \ldots\ldots, e_n) = \frac{P(e_1, \ldots e_n \mid c)\, P(c)}{P(e_1, \ldots\ldots, e_n)}$$

a mess

but sometimes able to make an assumption

└ That each flip is ind of each other

Probs conditionally ind. given the class

$$= \frac{P(e_1 \mid c)\, P(e_2 \mid c) \ldots\ldots P(e_n \mid c)}{P(E)}$$

T

.5

=.25

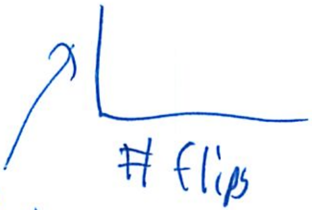
← called nieve baise when we assume ind.

.2

=.16

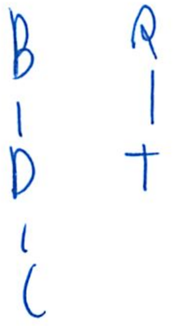
Example demonstration on compute

each line is a particular coin



likelihood line is correct

But what if friend suggests other model



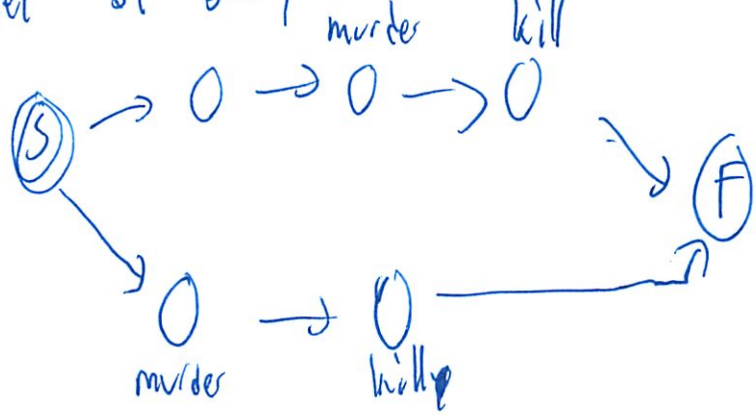
Can calc the prob of what is observed

Can do same as w/ coin $<$ $P(E|\text{model }1)$ / $P(E|\text{model }2)$

Use evidence to determine which is the right model
└ What gives higher probability of evidence

Stories

Model of story



could generate a new model
- perturb

is it more probable than original model ??

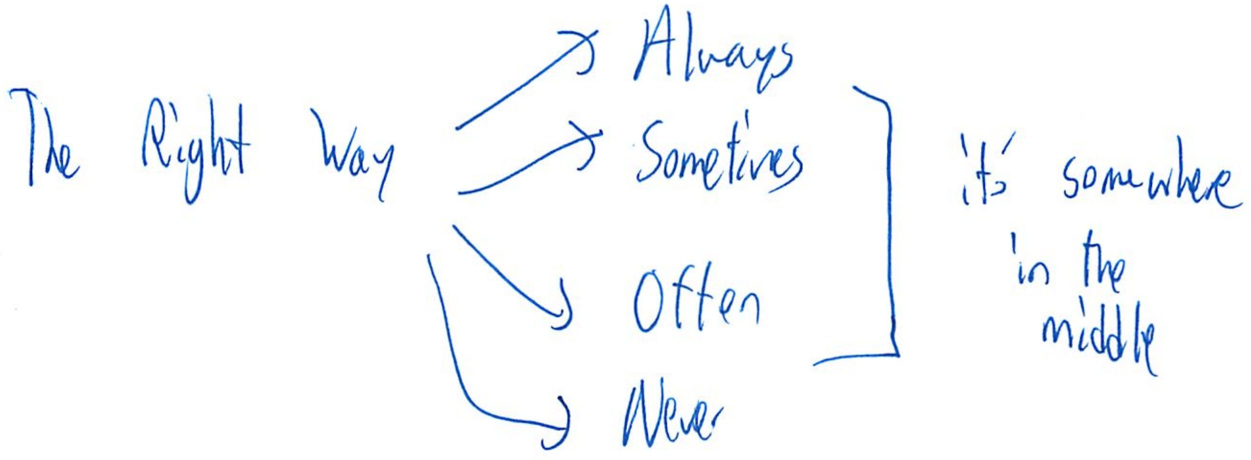
do plain hillclimbing search for models

Can do w/ stories

Find consistencies of human condition

10

AI people fight about this



The Right Way

Always

Sometimes

Often

Never

it's somewhere in the middle

6.034 Recitation

12/1

2 handouts

Quiz: Dec 7 ; SVM, Boosting, Representation
everything up to Thanksgiving is fair game

No mega recitation this Friday

This is 2nd last recitation

End of Boosting Problem

did most of problem last time

page 3 3rd column - classifiers a, b

Pick lowest error rate one

Give classifier weight based on info content

$$\alpha = \frac{1}{2} \ln \frac{1-E}{E}$$

So end up w/ big classifier

$$H(x) = \frac{1}{2} \ln 4 \, F + \frac{1}{2} \ln 3 \, B + \overbrace{\qquad\qquad}^{\text{last classifier}}$$

Rand 3;

Gets it wrong

So weight ↑

Reweighting

$$w' = \frac{1}{2} \frac{1}{E} w$$

$E = 4/16$

$w = \frac{1}{16}$

$$w' = \frac{1}{2} \cancel{\frac{16}{4}} \cancel{\frac{1}{16}} = \frac{1}{8} = \frac{3}{24}$$

other ones $\frac{\ }{24}$

So makes arithmatic eaiser

etc

I is best clasifier on next rand

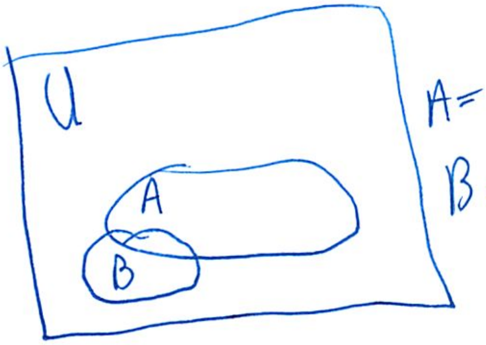
---

might over fit a bit

③

Probability

One of best methods in last 20 years dealing w/ uncertanity



A= Paul Revere wins

B = Weather ~~[illegible]~~ clear

$$0 \le P(\underset{\substack{\uparrow \\ \text{any area}}}{A}) \le 1$$

notation

$$\boxed{P(A \wedge B) = P(A,B)}$$

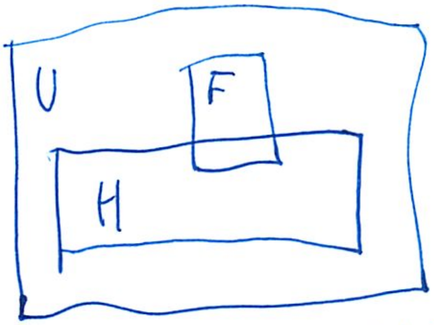
~~P(A~~

~~P(A)+2P(B)~~

$$P(A \vee B) = P(A) + P(B) - \underset{\text{double cant}}{P(A \wedge B)}$$

4

Conditional Probability



F = Have flue
H = Have Headache

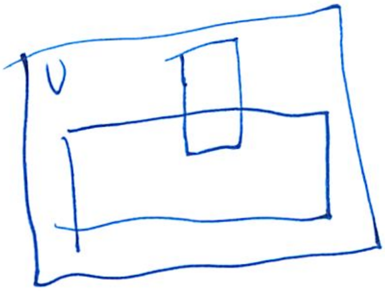
$$\underset{\substack{\uparrow \\ \text{conditional} \\ \text{prob}}}{P(H|F)} = \overset{\downarrow \text{def.}}{\frac{P(H \wedge F)}{P(F)}}$$

↓ (chain rule

$$P(H,F) = P(H|F) \cdot P(F)$$

Use in Nieve Bayes

Bayes Formula



$P(H) = \frac{1}{10}$ ← prior probability

$P(F) = 1/40$ ←

$P(H|F) =$

$P(F|H) =$ $\neq$

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

$$P(F|H) = \frac{P(H \wedge F)}{P(H \wedge F) + P(H)} \rightarrow \frac{P(H \wedge F)}{P(F) + P(H \wedge F)} \cdot \frac{P(H \wedge F) + P(F)}{P(H \wedge F) + P(H)}$$

↑
posterier probability

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

Had some prior estimate → P(F)

Then get info about H

Now have more info → P(F|H)

Really is learning

One of the most used learning algorthms

P(Revere | $\overset{\text{weather}}{\text{W.}}$ clear]

P(Revere | W clear, jockey is friend, worked 5-9)

? What happens to prob?

Bias Shrinks - since more conditioning factors

Varience ↑ But hard to estimate w/ so many times

~~P(Revere wins | V loses~~ , |

P(R wins, V loses, E loses | cler)

need rule to move items to other side

= P(R wins | V loses, $\overset{\text{interconnected}}{\text{E}}$ loses, w clear) *

P(V loses | E loses, w clear) *

P (E loses | w clear

~~(seperate races)~~

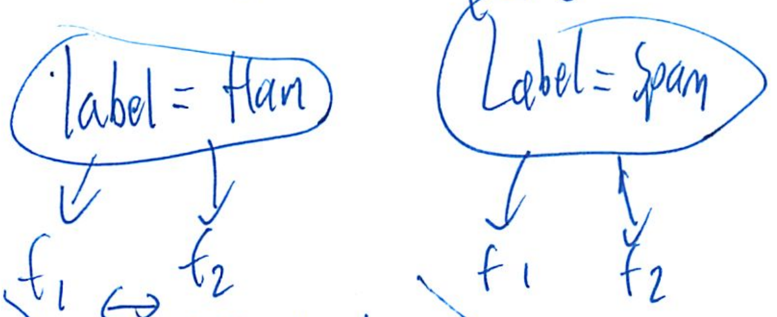
if seperate races - ie others don't matter - can drop

P(R | w clear)   "conditionally independent"

Naive Bayes

L = lable = {H, S}

feat 1: mentions $
Feat 2: contains "buy"



conditional relationship
↓
knowing if Ham lets you know more about what f1, f2 is

$f_1 \leftrightarrow$ Conditionally ind of each other (why its naive Bayes)

Are not really ind in real life assuming none here

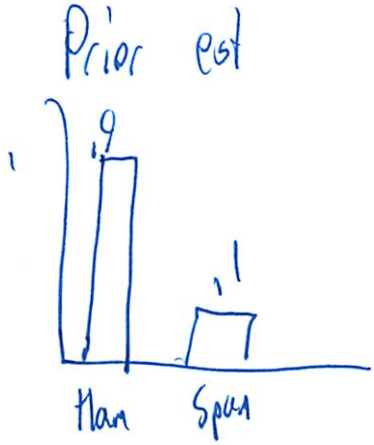
eavation simpler then

$$P(f_1, \ldots f_n \mid \text{lable} = \text{ham})$$

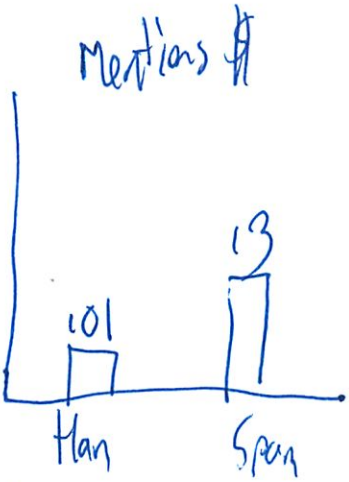
$$P(f_1 \mid \text{lable} = \text{ham}) * P(f_2 \mid \text{lable} = \text{ham})$$

since conditional ind.

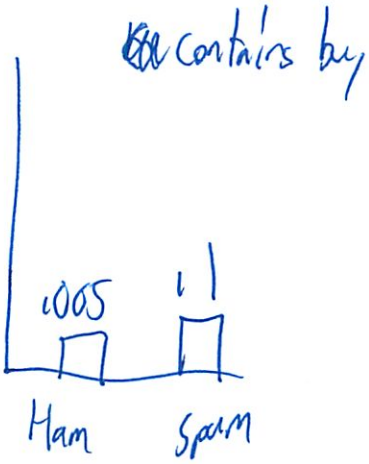
8

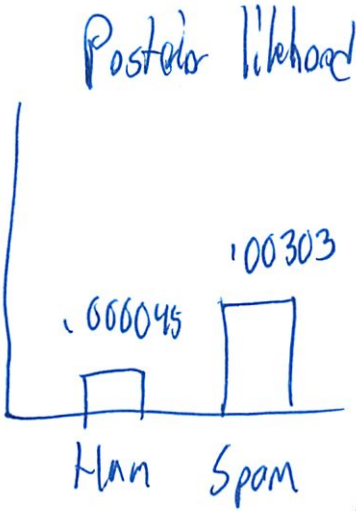


w/o even looking at it



how much spame email mentions $ (from training data)





this email contains both

lable we are looking at (row)

	Prior	·	P(feat 1($) | ℓ)	×	P(feat 2(buy) | ℓ)	= lable likelihood
Ham	.9	x	.01	x	.005	= .000045
spam	.1	x	.30	x	.10	= .00303

Then pick max L email is spam

Massachvsetts Institvte of Technology
Department of Electrical Engineering and Computer Science
6.034 Recitation 11, Thursday, December 1

Probability & Naïve Bayes — Prof. Bob Berwick, 32D-728

Agenda:
1. Finish Boosting
2. Probability: Axioms, Conditional Probability, Chain rule, Conditional independence, Bayes' Theorem
3. Naïve Bayes: another classifier (used for, e.g., Spam Asssasin)
4. Beyond naïve Bayes: the maximum entropy stewpot

1. Boosting and the Adaboost algorithm

The idea behind **boosting** is to find a weighted combination of s "weak" classifiers (classifiers that underfit the data and still make mistakes, though as we will see they make mistakes on less than ½ the data), h_1, $h_2 \ldots, h_s$, into a **single strong** classifer, $H(x)$. This will be in the form:

$$H(\vec{x}) = sign(\alpha_1 h_1(\vec{x}) + \alpha_2 h_2(\vec{x}) + \cdots + \alpha_s h_s(\vec{x})$$

$$H(\vec{x}) = sign\left(\sum_{i=1}^{s} a_i h_i(\vec{x})\right)$$

$$\text{where: } H(\vec{x}) \in \{-1,+1\}, h_i(\vec{x}) \in \{-1,+1\}$$

Recall that the *sign* function simply returns +1 if weighted sum is positive, and –1 if the weighted sum is negative (i.e., it classifies the data point as + or –).
Each training data point is **weighted.** These weights are denoted w_i for i=1, ..., n. **Weights** are *like* probabilities, from the interval (0, 1], with their **sum equal to 1. BUT** weights are **never 0**. This implies that **all data points have some vote** on what the classification shuld be, at all times. (You might contrast that with SVMs.)

The general idea will be to pick a single 'best' classifier h (one that has the lowest error rate when acting all alone), as an initial 'stump' to use. Then, we will **boost** the weights of the data points that this classifier **mis-classifies (makes mistakes on),** so as to focus on the next classifier h that does best on the re-weighted data points. This will have the effect of trying to fix up the errors that the first classifier made. Then, using this next classifier, we repeat to see if we can now do better than in the first round, and so on. In computational practice, we use the same sort of entropy-lowering function we used with ID/classifier trees: the one to pick is the one that lowers entropy the most. But usually we will give you a set of classifiers that is easier to 'see', or will specify the order.

In Boosting we always pick these initial 'stump' classifiers so that the error rate is strictly < ½. Note that if a stump gives an error rate greater than ½, this can always be 'flipped' by reversing the + and – classification outputs. (If the stump said –, we make it +, and vice-versa.) Classifiers with error exactly equal to ½ are useless because they are no better than flipping a fair coin.

1. Here are the definitions we will use.
Errors:
The error rate of a classifier s, E^s, is simply the sum of all the *weights* of the training points classifier h_s gets **wrong**.
$(1–E^s)$ is 1 minus this sum, the sum of all the *weights* of the training points classifier h_s gets **correct.**
By assumption, we have that:
$E^s < ½$ and $(1– E^s) > ½$, so $E^s < (1– E^s)$, which implies that $(1– E^s)/E^s > 1$

Weights:

α_s is **defined** to be ½ ln[(1– E^s)/ E^s)], so from the definition of weights, the quantity inside the ln term is > 1, so all alphas must be positive numbers.

Let's write out the Adaboost algorithm and then run through a few iterations of an example problem.

Adaboost algorithm

Input: training data, $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

1. **Initialize data point weights.**

   Set $w_i^1 = \dfrac{1}{n} \quad \forall i \in (1,\ldots,n)$

2. **Iterate over all 'stumps':** for s=1, ..., T

   a. **Train base** learner using distribution w^s on training data.
      Get a base (stump) classifier $h_s(x)$ that achieves the lowest error rate E^s .
      (In examples, these are picked from pre-defined stumps.)

   b. **Compute the stump weight:** $\alpha_s = \dfrac{1}{2}\ln\dfrac{(1-E^s)}{E^s}$

   c. **Update weights** (3 ways to do this; we pick Winston's method)

      For points that the classifier **gets correct,** $w_i^{s+1} = \left[\dfrac{1}{2}\cdot\dfrac{1}{1-E^s}\right]\cdot w_i^s$

      (Note from above that 1– E^s> ½, so the fraction 1/(1– E^s) must be < 2, so the total factor scaling the old weight must be < 1, i.e., the **weight of correctly classified points must go DOWN in the next round)**

      For points that the classifier **gets incorrect,** $w_i^{s+1} = \left[\dfrac{1}{2}\cdot\dfrac{1}{E^s}\right]\cdot w_i^s$

      (Note from above that E^s < ½, so the fraction 1/E^s) must be > 2, so the total factor scaling the old weight must be > 1, i.e., the **weight of incorrectly classified points must go UP in the next round)**

3. **Termination condition:**

   If $s > T$ or if $H(x)$ has error 0 on training data or < some error threshold, exit;
   If there are no more stumps h where the weighted error is < ½, exit (i.e., all stumps now have error exactly equal to ½)

4. **Output final classifier:**

   $H(\vec{x}) = sign\left(\sum_{i=1}^{s} a_i h_i(\vec{x})\right)$ **[this is just the weighted sum of the original stump classifiers]**

Note that test stump classifiers that are **never** used are ones that make more errors than some pre-existing test stump. In other words, if the set of mistakes stump X makes is a **superset** of errors stump Y makes, then Error(X) > Error(Y) is **always** true, no matter weight distributions we use. Therefore, we will **always** pick Y over X because it makes fewer errors. So X will **never** be used!

Let's try a boosting problem from an exam (on the other handout).

Food for thought questions.

1. How does the weight α^s given to classifier h_s relate to the performance of h_s as a function of the error E^s?
2. How does the error of the classifier E^s affect the new weights on the samples? (How does it raise or lower them?)
3. How does AdaBoost end up treating outliers?
4. Why is not the case that new classifiers "clash" with the old classifiers on the training data?

5. Draw a picture of the training error, theoretical bound on the true error, and the typical test error curve.
6. Do we expect the error of new weak classifiers to increase or decrease with the number of rounds of estimation and re-weighting? Why or why not?

Answers to these questions:

1. How does the weight α^s given to classifier h_s relate to the performance of h_s as a function of the error E^s?

Answer: The lower the error the better the classifier h is on the (weighted) training data, and the larger the weight α^t we give to the classifier output when classifying new examples.

2. How does the error of the classifier E^s affect the new weights on the samples? (How does it raise or lower them?)

Answer: The lower the error, the better the classifier h classifies the (weighted) training examples, hence the larger the increase on the weight of the samples that it classifies incorrectly and similarly the larger the decrease on those that it classifies correctly. More generally, the smaller the error, the more significant the change in the weights on the samples.
Note that this dependence can be seen indirectly in the AdaBoost algorithm from the weight of the corresponding classifier α_t· The lower the error E^t, the larger α_t, the better h_t is on the (weighted) training data.

3. How does AdaBoost end up treating outliers?

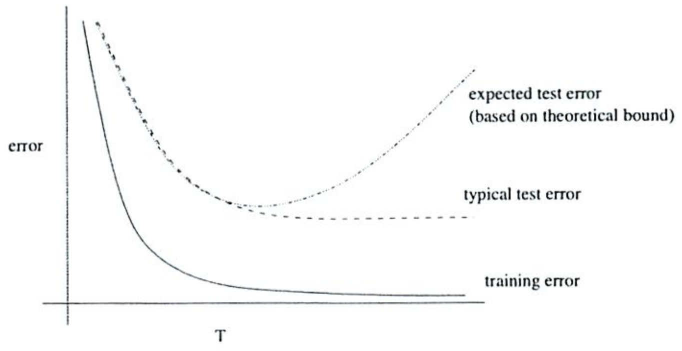
Answer: AdaBoost can help us identify outliers since those examples are the hardest to classify and therefore their weight is likely to keep increasing as we add more weak classifiers. At the same time, the theoretical bound on the training error implies that as we increase the number of base/weak classifiers, the final classifier produced by AdaBoost will classify all the training examples. This means that the outliers will eventually be "correctly" classified from the standpoint of the training data. Yet, as expected, this might lead to overfitting.

4. Why is not the case that new classifiers "clash" with the old classifiers on the training data?

Answer: The intuition is that, by varying the weight on the examples, the new weak classifiers are trained to perform well on different sets of examples than those for which the older weak classifiers were trained on. A similar intuition is that at the time of classifying new examples, those classifiers that are not trained to perform well in such examples will cancel each other out and only those that are well trained for such examples will prevail, so to speak, thus leading to a weighted majority for the correct label.

5. Draw a picture of the training error, theoretical bound on the true error, and the typical test error curve.

Answer:



6. Do we expect the error of new weak classifiers to increase or decrease with the number of rounds of estimation and re-weighting? Why?

Answer: We expect the error of the weak classifiers to increase in general since they have to perform well in those examples for which the weak classifiers found earlier did not perform well. In general, those examples will have a lot of weight yet they will also be the hardest to classify correctly.

2. Basics of probability (review & pictures)

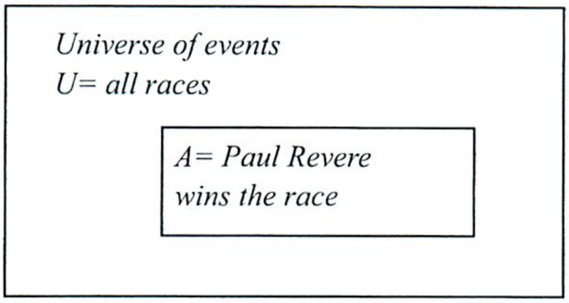
The fundamentals of probability theory: the axioms of probability. Why are these important? The power of the purse: Because while there are *other* attempts to handle the notion of 'uncertainty', e.g., 'fuzzy logic', '3-valued logic', etc., these axioms are the **only** system with the property that **if** you **gamble** with them, you **cannot** be unfairly exploited by an opponent who uses some other system (Di Finetti, 1932 theorem).

So, some first concepts.

We say that A **is a random variable** if A denotes an event and there is some uncertainty if A is true.

Typically, we let U denote the **universe** of all possible events (= all "possible worlds"). Then a subset of U, call it A, corresponds to the set of events in which A is true.

Example. Let the universe U be the set of all horse races. Let *Paul Revere* (abbreviation: P-R) be a horse. Then we can let A denote the set of racing events in which Paul Revere wins. We can draw this as a picture, where *races* labels the outer square, the universe, and the circle inside is the set of all events where Paul Revere wins the race:



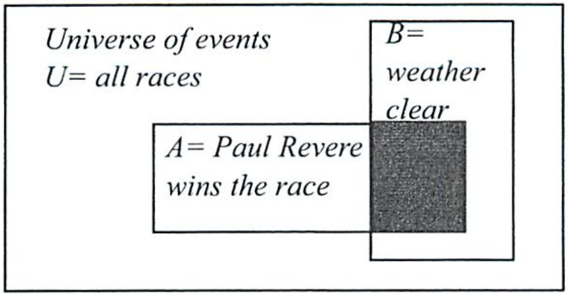
Let us denote by ***P*(*A*)** the fraction of events (possible worlds in the universe of events) in which A turns out true. We could spend the next 2 hours on the philosophy of possible worlds and this business. But we won't.

We will compute probabilities using an informal notion of *areas* (formally, we'd use measure theory).

The Universe of all events has total area 1, $P(U)=1$, because it denotes all the events that are true. $P(A)$ then is the area of the smaller rectangle with respect to U (= the fraction of the total universe in which Paul Revere wins). $P(\neg A)$= the races in which Paul Revere does **not** win = the set difference between U and A. From this we will posit 3 axioms regarding $P(A)$:

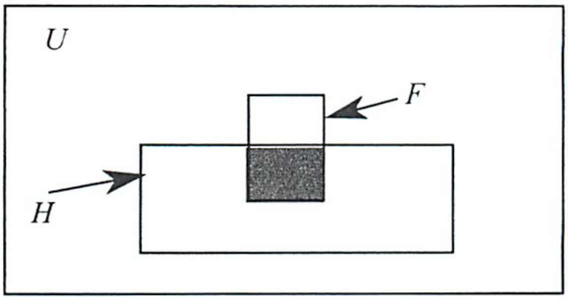
(1) $0 \leq P(A) \leq 1$ [because: the area of A cannot be < 0 or > 1]
(2) $P(true)=1$
(3) $P(false)=0$
(4) $P(A \vee B) = P(A) + P(B) - P(A,B)$ [where $\vee$ means "or", i.e., **either** A or B must be true; + means "add together", and the comma in A, B means "and", i.e., both A **and** B must be true]

To see how this last axiom works, let's look at the racing universe with event A= Paul Revere wins and a second event, B= the weather is clear. The **shaded area** represents the fraction of events when **both** A and B are true, i.e., $P(A,B)$= true:



It should be apparent that in order to figure out the probability of *A* **or** *B*, we need to add up the areas corresponding to *A* and to *B*, but then subtract out the shaded area so that it is not counted twice. In this way, we arrive at the formula for the probability of *A* **or** *B*.

We next turn to the notion of **conditional probability.**
We let *P*(*A*|*B*) denote the fraction of events/possible worlds in which *B* is true, and then *also* have *A* true. That is, we 'shrink' the universe from *U* down to *B*, focusing in on a subset possibly more relevant to our situation, and use *that* as our basis to calculate probabilities.
Example. In the figure below, we illustrate the following situation. Let *H*= probability that "I have a headache"; *F*= probability that "I am getting the flu". These are denoted by the rectangles *H* and *F* in the figure below. Let us assume:
P(*H*) = 1/10; *P*(*F*)=1/40. Now let's compute the conditional probability *P*(*H*|*F*), i.e., the probability that I have a headache **given** that I have the flu. This is the fraction of flu-events that are also headache events – that is, if we just look at the rectangle *F*, what proportion of *F* overlaps with *H*? (The answer is 1/2). Thus, *P*(*H*|*F*)=1/2.



In other words, to find *P*(*H*|*F*), we compute:
(# worlds in which *H* **and** *F* are true)/(# worlds in which *F* is true) or,
(area *H* **and** *F*)/(area of *F*), or
P(*H*, *F*)/*P*(*F*)

So this is the **formula for conditional probability:**

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

Note how *P*(*B*) is in the denominator here. Multiplying out, we obtain the important formula called the **chain rule** which we will uses in the naïve Bayes classifier:

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

Some other manipulations of conditional probability will be used in what follows. We consider two: (i) simplifications to the *right* of the conditioning bar symbol |; and (ii) simplifications to the *left* of the conditioning bar symbol.
Simplifications to the *right* of the bar:
Suppose we have *lots* of conditions to impose on whether or not Paul Revere wins. For example, this could depend on not only if the weather's clear, but also whether the jockey's brother is a friend of mine, whether Paul Revere won its last race, etc. In other words:

P(Paul Revere wins | weather clear, jockey's brother a friend, P-R won last race)

Note that *adding* terms to the right only makes the conditions more stringent, so that this probability should get lower and lower every time we add a new factor. (Why? Think about intersection.) With more factors then, we have less *bias*, because we are focusing in on our particular situation, but we will have more *variance*, because it will become harder and harder to measure all these terms perfectly. So, sometimes we will want to reduce the number of factors to the right of the conditioning symbol to those we are more confident we can estimate; this is called *back off*. (We will see this in action soon). There is no problem in simply doing this:

P(Paul Revere wins | weather clear, ~~jockey's brother a friend~~, ~~P-R won last race~~)

And then of course just having *P*(Paul Revere wins | weather clear) remaining. But what about if there are more terms to the *left* of the bar, as in this case:

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)

If we just care about Paul Revere, are we allowed to simply strike out the other two horses, this way?

P(Paul Revere wins, ~~Valentine loses~~, ~~Epitaph loses~~ | weather clear)

The answer is: No! We need to carry out a more complex expansion to isolate Paul Revere on the left. To see how, let's abbreviate Paul Revere wins as *R*, Valentine loses as *V*, Epitaph loses as *E*, and the Weather is clear as *W*. Then our conditional probability:

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)

Can be abbreviated as:

$$\frac{P(R,V,E,W)}{P(W)}$$

We can use this formula to derive the **chain rule for conditional probability:**

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)=
P(Paul Revere wins| Valentine loses, Epitaph loses, weather clear) ×
P(Valentine loses | Epitaph loses, weather clear) ×
P(Epitaph loses | weather clear)

Proof. Writing out the 3 terms:

$$\frac{P(R,V,E,W)}{P(W)}=\frac{P(R,V,E,W)}{P(V,E,W)}\times\frac{P(V,E,W)}{P(E,W)}\times\frac{P(E,W)}{P(W)}$$

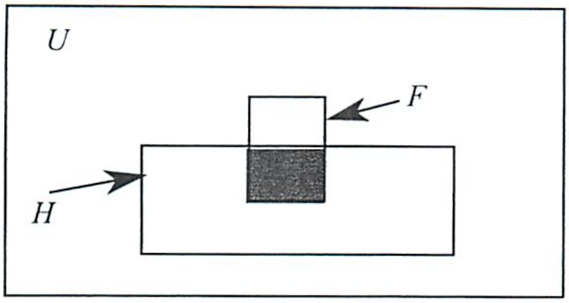
Now, supposed it is the case that the following simpler expansion holds:
P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear) =
P(Paul Revere wins| ~~Valentine loses, Epitaph loses~~, weather clear) ×
P(Valentine loses | ~~Epitaph loses~~, weather clear) ×
P(Epitaph loses | weather clear)

In this case, whether Paul Revere wins or not depends *only* on whether the weather's clear…and not on what the other two horses do. They are irrelevant factors, so we can strike them out. In this case, when the probability is *unchanged* when we drop out conditioning factors, we say that the probability is **conditionally independent** (independent of the other horses, but still conditioned on the weather). More generally, if there are *n* factors *f*, and each factor is independent of the other, but still dependent on a condition *c*, we can write the following, which will be another key ingredient in our naïve Bayes classifier model:

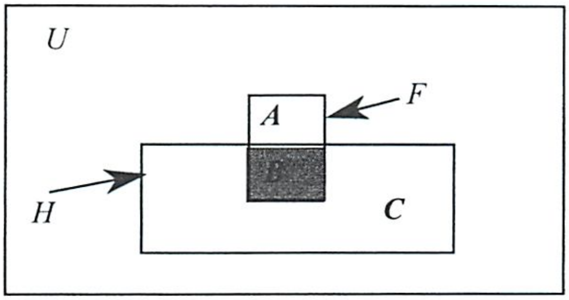
$$P(f_1,\ldots,f_n \mid c)=P(f_1 \mid c)\times\ldots\times P(f_n \mid c)$$

That is, we can just write out the probability as the product of the *n* factors, **assuming they are independent from one another** (the outcomes of these events do not affect the outcomes of one another); note the factors are still dependent on the outcome of event *c*.

OK, we come to the last ingredient we shall need, **Bayes' Law**. Again we can illustrate this with the simple picture of headache and flu as before. Recall *P*(*H*)=1/10; *P*(*F*)=1/40, *P*(*H*|*F*)=1/2.



Now we will **label** each of the distinct regions in this diagram, *A, B,* and *C*, as follows. *A*+*B*=area of *F*; *B*+*C*= area of *H*:



By the definition of conditional probability, *P*(*H*|*F*)= *P*(*H*,*F*)/ *P*(*F*) = *B*/(*A*+*B*).
Now consider this reasoning: one day you wake up with a headache, and you think, OMG, 50% of flus are associated with headaches, so now I have a 50-50 chance of getting the flu." Is this reasoning correct?

What we *want* to compute is: *P*(*F*|*H*). We already know the *other* conditional probability, that of headache given the flu. Further, by the definition of conditional probability, in terms of the regions *A, B,* and *C*, we have that: *P*(*F*|*H*) = *B*/(*B*+*C*). To find this last ratio of regions, we can take the conditional probability *P*(*F*|*H*) = *B*/(*A*+*B*), and multiply it by (*A* +*B*)/(*B* + *C*), as follows:

$$\frac{B}{B+C} = \frac{B}{A+B} \cdot \frac{A+B}{B+C} \text{ i.e.,}$$

$$P(F \mid H) = P(H \mid F) \bullet \frac{P(F)}{P(H)}$$

$$\text{in our example, } \frac{1/2 \times 1/40}{1/10} = \frac{1/80}{1/10} = \frac{1}{8}$$

The term *P*(*F*) is called the **prior probability** (of getting the flu); the term *P*(*H*|*F*) is called the **likelihood;** the term *P*(*H*) is the **evidence** (e.g., that you have a headache); and the term *P*(*F*|*H*) is called the **posterior probability** of getting the flu (given that you have a headache). So this updated probability is a kind of learning: given the fact (data) that you indeed have a headache, how does the probability of getting the flu change? (It increases from 1/40 to 1/8.) Inverting from *P*(*H*|*F*) to *P*(*F*|*H*) is called **Bayes' Law.** It follows from a very simple manipulation of the definition of conditional probability and then application of the chain rule, i.e., that *P*(*A*,*B*)= *P*(*A*|*B*)×*P*(*B*):

$$P(B \mid A) = \frac{P(A,B)}{P(B)} \text{ (by dfn of conditional probability)}$$

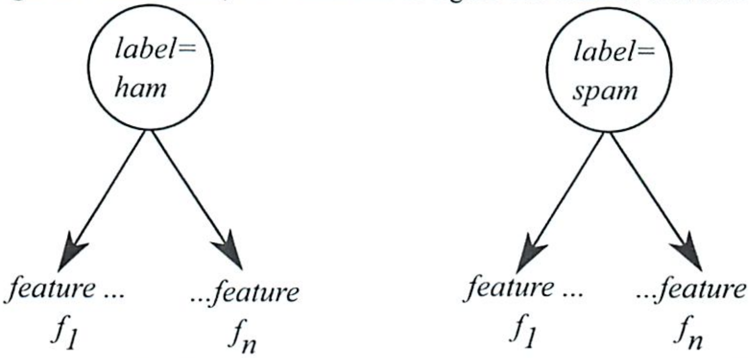
$$= \frac{P(A \mid B) \cdot P(B)}{P(B)} \text{ (by chain rule, replacing } P(A,B))$$

Or in words we can say this:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Now let's put this all to work to build a classifier called **Naïve Bayes**. Like k-means and ID-trees, and Boosting, etc., this will take as input the values of some **features** and then output a classification **label.**

As our example, we will use the common, but valuable task of classifying email into 1 of 2 categories: either good email ("ham") or bad email ("spam"). The underlying probability model follows what is called a **Bayes' net**. We can imagine the following generative process: we pick a label, e.g., "ham", and given this label, email documents of this type will have a certain distribution of feature values $f_1, \ldots, f_n$. If we pick the other label, "spam", we will get another distribution for the feature values (hopefully distinct). So the picture looks like this, and the idea of course is that **given** a **new** email, we would like to figure out whether it is ham or spam:



Crucially, we assume that **the features are independent from one another.** (This is the "naïve" part of Naïve Bayes.) Their values depend on (are conditioned on) **only** the value of the label. That is why we draw the networks as above, with **no links** between the features, only from the label directed down to the features.

Now here's the idea behind the classificiation.. Suppose we have estimated that 90% of our email is "ham" (OK), and that 10% is "spam". This gives us our **prior probability estimates** $P(label{=}ham){=}0.9$ and $P(label{=}spam){=}0.1$. That's what we can say about any new email **without any additional information.** (We'll see below how we get these estimates.)

Now, when we get a new email, we will get the values of its **features** and use these to adjust the prior probabilities, as with our headache example. (In our example, to keep things simple, we will use only two features.)

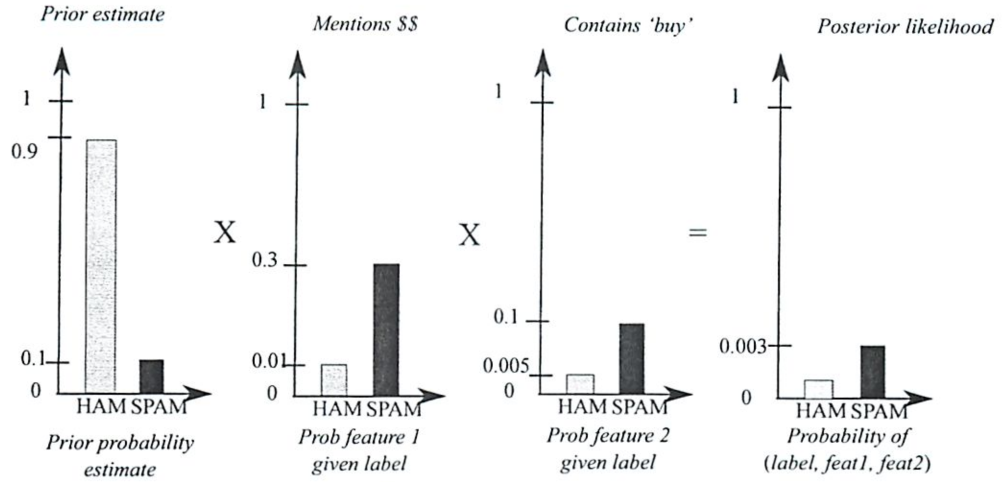
So, this new email comes along: "*Buy this amazing new Ginsu knife for only $39.....*"

Is this ham or spam? We'll assume that we use the following 2 features:

Feature 1: The email mentions money; this occurs in 30% of spam, and in 1% of ham

Feature 2: The email contains the word 'buy'; this occurs in 10% of spam, and in 0.5% of ham

We can picture our calculation as follows: our initial prior probabilities for each category are adjusted by **multiplying** the contribution **each feature** 'votes' (independently) as to how likely each category is. Then we pick the **most likely = biggest probability category** at the end:



So, in this case, our new email is classified as "spam" because this yields the largest posterior likelihood. Note how we got this value. It is simply this:

$P(label) \times P(f_1 \,|\, label) \times P(f_2 \,|\, label) = P(label, f_1, f_2)$ [recall from dfn of conditional prob that:
$\dfrac{P(label, f_1, f_2)}{P(label)} = P(f_1 \,|\, label) \times P(f_2 \,|\, label)$ IF f_1, f_2 are independent of one another]

In other words, we multiple the following out to find the label likelihood, and pick the biggest likelihood:

Prior probability of a label × Probability of feature contributions = Posterior label likelihood

In our case, for the two labels "ham" and "spam":

	Prior ×	Pr(feat1 ($)| l) ×	Pr(feat 2 ('buy')|l)	= Label likelihood	
Ham:	0.9 ×	0.01 ×	0.005	= 0.000045	(log of this likelihood: –4.34)
Spam:	0.1 ×	0.30 ×	0.10	= 0.00303	(log of this likelihood: –2.52)

So, our email is more likely to be spam than ham. In fact, taking the ratios of the log likelihoods, –2.52/–4.32, the email is about 2 orders of magnitude (100x) more likely to be spam than ham. Recall that: (1) the features **must** be independent of one another; (2) we can add other features, of course…this is what a program like Spam Assassin can do, by training; and (3) one can use this method with lots more categories to **classify** documents (see the end of the handout).

Let's turn to justifying this approach probabilistically, as well as how we actually estimate the probability values above, via training, and highlighting some pitfalls.

First, why is this justified? We are computing the **maximum** probability that an input email will have a particular label (category), **given** that it has a particular set of features. We pick the label that maximizes: *P(l=value | observed features)*. Let's follow out this logic. We are maximizing the following quantity over label values:

$$\max P(label \,|\, features) = \max \frac{P(features, label)}{P(features)}$$ [by dfn of conditional probability]

But note that the denominator in the expression above, $P(features) = P(f_1, \ldots, f_n)$ is *constant* no matter what our choice of label value. So, to maximize the above quantity, it suffices to maximize the numerator:

$$\max P(features, label) = P(f_1, \ldots, f_n, label)$$

By the chain rule, this quantity in turn is just:

$$\max P(label) \times P(f_{1,}, \ldots, f_n \,|\, label)$$

But given that the features are all independent of one another, this is the same as (recall our Paul Revere example!):

$$\max P(label) \times P(f_1 \mid label) \times \ldots \times P(f_n \mid label)$$

$$\max prior \quad \times 'vote' f_1 \quad \times \ldots \times 'vote' f_n$$

This is exactly the computation we have carried out. It remains to figure out how we 'train' our classifier – that is, how do we get the various estimates of the probabilities above? The simplest thing is just to estimate them from counts in training text, that is, known examples of ham and spam emails. These are the so-called *maximum likelihood estimates*:

$$P(label = ham) = \frac{count\ (\#\ ham\ emails)}{count(total\ \#\ emails)} \qquad P(label = spam) = \frac{count\ (\#\ spam\ emails)}{count(total\ \#\ emails)}$$

$$P(f_1 \mid label = ham) = \frac{count\ (\#\ ham\ emails\ mention\ \$)}{count(total\ \#\ ham\ emails)}$$

$$P(f_1 \mid label = spam) = \frac{count\ (\#\ spam\ emails\ mention\ \$)}{count(total\ \#\ spam\ emails)}$$

$$P(f_2 \mid label = ham) = \frac{count\ (\#\ ham\ emails\ contain\ \text{'buy'})}{count(total\ \#\ ham\ emails)}$$

$$P(f_2 \mid label = spam) = \frac{count\ (\#\ spam\ emails\ contain\ \text{'buy'})}{count(total\ \#\ spam\ emails)}$$

So this is how we get the estimates. For example, if we have 1000 emails, 900/1000 are ham, and 100/1000 are spam. Of the 100 spam emails, 30/100 mention money, and 1/100 contain 'buy'. For ham emails, 1/100 mention money and 5/1000 contain 'buy'.
Note that as the # of data samples (amount of training data) increases, then our estimates should get better; one of the properties of the maximum likelihood estimates is that they will converge to the 'true' values as the amount of data goes to infinity. (The mean approaches the true average.) But, if the # of training examples is small, our estimate will be very lousy, and have more noise (variance); there are a variety of things we can do to improve this, but that's for a machine learning course.

However, there is one particular case we should note. Suppose a particular count is actually 0 – that is, we *never* observe a particular feature associated with a particular label – this will happen especially if we keep adding more and more features. In this case, note that the entire probability product to find the likelihood will *all* be zero, just because one of the estimates is 0. So this is very bad!

There is a whole cottage industry devoted to fixing this problem, and it is called *smoothing*. It is basically the Robin Hood strategy: we rob probability mass from the rich and give it to the poor. In particular, the *simplest* smoothing strategy, invented by Laplace, is called *add*–1 *smoothing*: if a count is 0, we add 1 to it, so that, e.g., 0/100 goes to 1/100. (We must also *subtract* the appropriate probability mass, i.e., counts, from the *rest* of our estimates, so that the probabilities still add up to 1 in all.)

A second method of smoothing (probability mass redistribution) is due to Alan Turing. He figured this out when he was developing probability formulas for estimating the likelihood of finding German submarines in particular areas of the ocean. What if a submarine had *never* been observed in a particular spot? (Something that's actually quite likely!) Turing reasoned that a fairly good probability estimate of 'things never seen' would be quite close to the estimate of 'things seen *exactly* once'. This method, now called Good-Turing smoothing (only published until decades after WWII), works well but is finicky. There are whole books devoted to this subject, for machine learning and especially in natural language processing, where we quickly get word sequences never seen before.

One more thing. You may note that in our calculation we multiply together a (possibly long) string of probabilities, one for each feature. With a 1000 features, this value will quickly get very

very small. So, the usual method is to operate in log space, where multiplication is just addition, so we can maintain accuracy. (That's why we used log likelihoods above.)

Beyond Naïve Bayes (Optional)
OK, this method is fine so far as it goes, but it can be improved enormously. Here we will just sketch one method, known as **maximum entropy classification** that can gobble down any set of features, even if they are not independent. Yet remarkably, as first shown by Jaynes (1957), it is the most probabilistically sound method of **combining diverse features.** It rationalizes the general notion of just 'scoring' features and adding them up. We won't prove this here, but just indicate the general approach, which is now broadly used in, e.g., figuring out the part of speech labels in text. (For instance, in the sentence, *police police police*, is the first *police* a Noun or a Verb?)

1. To begin, let's assume there are now 10 labels for documents, with categories *A, B, C, D, E, F, G, H, I, J.* (So, e.g., category *A* could be travel; *B* sports; *C* business; etc.) If we know this, and **no other information** then given an email *m*, what is our best guess for category *C* (business) given this email, i.e., $P(C \mid m)$?
The maximum entropy approach would claim it is 1/10: that is, we maximize the quantity in each of the 10 bins, uniformly, by spreading out the total probability mass of 1 among 10 bins.

2. Now suppose I tell you that 55% of all emails are in category *A*, travel? Now what is the quantity $P(C| m)$? I think it should not be too hard to see that *A* gobbles up 0.55 of the probability mass, leaving 0.45 to be distributed evenly over the remaining 9 categories, or 0.05 for each of the remaining categories, including category *C*, business. So the maximum entropy estimate for $P(C| m)$ is 0.05.

3. Now suppose I add *another* constraint: that *in addition* to the fact in (2), we know that 10% of all emails contain the word 'buy'. What is $P(C| m)$ now? This gets harder to visualize, so we'll write it out as a table, where the first row is the probability of containing 'buy' (which thus must add up to 0.1 of all emails), and second row is the probability of not containing 'buy', which we have labeled *other* (which thus must add up to 0.9). Once again following the maximum entropy idea, since we don't know anything else about the 'contains buy' row, we should distribute its 0.1 total *evenly* among the 10 bins, thus giving 0.01 to each. Next, since *all* of category *A* must add up to 0.55, and since the 'contains buy' cell holds 0.01, it must be that the cell in the row labeled *other* and in column *A* must have the value 0.54 (so that the column total is 0.55). That leaves 0.9–0.54 = 0.36 for the rest of the 9 bins in the *other* row. Once again, spreading this evenly, we get 0.36/9 = 0.04 for each of these bins (so that each column here adds to 05). Thus we have the following table:

	1	2	3	4	5	6	7	8	9	10
	A	B	C	D	E	F	G	H	I	J
buy	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
other	0.54	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

So, *why* is this called *maximum entropy*? You should realize that by spreading out the values evenly, we are *maximizing the entropy of the cell values*: $-p \log p$ summed over all entries is at a maximum. (Below we indicate *why* this is a good thing to do.) In any case, we are maximizing the entropy *subject to the constraints specified.* (We have two so far.)

4. So let's add one more constraint. Suppose that in addition, 80% of the 'buy' emails are in *either* category *A* or category *C*. Now we want to figure out $P(C| m)$. Gulp! This one is much harder to figure out – in fact, in general to do this, it is like spreadsheets, but we can indicate what has to be true in our table now: the probability of the *buy* row, column *A*, plus the probability in the *buy* row, column *C*, must add up to 0.08 (80% of the 10%). That turns out to be the values 0.051 and 0.029. Since that leaves 0.020 for the rest of the bins in the *buy* row, these must be 0.020/8=0.0025. Since column *A* must still add up to 0.55, then that leaves 0.499 for row *other*, column *A*. Since

the *other* row must still sum to 0.9, we have 0.9–0.499= 0.401 to distribute evenly over the rest of the *other* bins, so this is 0.401/9 = 0.0446. If we impose these constraints, you'll see that this is the answer (we don't say how we figured it out!)

	1	2	3	4	5	6	7	8	9	10
	A	B	C	D	E	F	G	H	I	J
buy	0.051	.0025	0.029	.0025	.0025	.0025	.0025	.0025	.0025	.0025
other	0.499	.0446	.0446	.0446	.0446	.0446	.0446	.0446	.0446	.0446

Now we know that *P(buy,C)*= 0.029; *P(C| buy)*= 0.29 (= 0.029/0.1); *P(A | buy)*= 0.51.
This is our classifier, a *maximum entropy* classifier.

The punchline. While there are many possible distributions that could yield the three observed constraints, that 55% of the emails are in category *A*, that 10% of the emails contain *buy*, and that of these 10%, 80% are in category *A* or *C*, the **one distribution** that we picked, where we have **maximized** the entropy of the probability mass subject to these constraints, turns out to be the **only one** having the following two properties, the second one quite remarkable:

1. This distribution follows the form: $P(email,label) = \frac{1}{Z(\lambda)} \exp \sum_i \lambda_i f_i(email,label)$ where the lambdas are the weights associated with each feature f_i; the function f_i returns 1 if the feature is in the email, and 0 otherwise; and Z is a normalizing constant to make sure the probabilities all add up to 1.
2. This distribution **maximizes the probability of the training data,** $\prod_j P(email_j, label_j)$

This is what justifies the method!

Problem 2: Boosting (50 points)

After wearing Sauron's ring for several months, Frodo is rapidly losing his sanity. He fears that the ring will interfere with his better judgement and betray him to an enemy. To ensure that he doesn't put his trust into enemy hands, he flees Middle Earth in search of a way to classify his enemies from his friends. In his travels he had heard rumors of the magic of Artificial Intelligence and has decided to hire you to build him a classifier, which will correctly differentiate between his friends and his enemies. Below is all of the information Frodo remembers about the people back in Middle Earth.

ID	Name	Friend	Species	Has Magic	Part of the Fellowship	Has/Had a ring of power	Length of hair (feet)
1	Gandalf	**Yes**	Wizard	Yes	Yes	No	2
2	Sarumon	**No**	Wizard	Yes	No	No	2.5
3	Sauron	**No**	Wizard	Yes	No	Yes	0
4	Legolas	**Yes**	Elf	Yes	Yes	No	2
5	Tree-Beard	**Yes**	Ent	No	No	No	0
6	Sam	**Yes**	Hobbit	No	Yes	No	0.25
7	Elrond	**Yes**	Elf	Yes	No	Yes	2
8	Gollum	**No**	Hobbit	No	No	Yes	1
9	Aragorn	**Yes**	Man	No	Yes	No	0.75
10	Witch-king of Angmar	**No**	Man	Yes	No	Yes	2.5

Part A: Picking Classifiers (10 points)

A1 (6 points)

The data has a high dimensionality and so rather than trying to learn an SVM in a high dimension space you think it would be a smart approach to come up with a series of 1 dimensional stubs that can be used to construct a boosting classifier. Fill in the classifier table below. Each of the different classifiers are given a unique ID and a test returns +1 (friend) if true and -1 (enemy) if false.

Classifier	Test	Misclassified
A	Species is a Wizard	2, 3, 4, 5, 6, 7, 9
B	Species is an Elf	1, 5, 6, 9
C	Species is **not** a Man	2, 3, 8, 9
D	Does **not** have magic	1, 4, 7, 8
E	Is **not** part of the Fellowship	1, 2, 3, 4, 6, 8, 9, 10
F	Has **never** owned a ring of power	2, 7
G	Hair <= 1ft	1, 3, 4, 7, 8
H	Hair <= 2 ft	3, 8
I	Friend	2, 3, 8, 10
J	Enemy	1, 4, 5, 6, 7, 9

A2 (4 points)

Looking at the results of your current classifiers, you quickly see two more good weak classifiers (make fewer than 4 errors). What are they?

Classifier	Test	Misclassified
K		1, 8, 10
L		

Part B: Build a Strong Classifier (30 points)

B1 (25 points)

You realize that many of your tests are redundant and decide to move forward using only these four classifiers:{**B, D, F, I**}. Run the Boosting algorithm on the dataset with these four classifiers. Fill in the weights, classifiers, errors and alphas for three rounds of boosting. In case of ties, favor classifiers that come first alphabetically. Note: initial weights are set to be EQUAL and so 1/10 (they must add up to 1)

	Round 1		**Round 2**		**Round 3**	
w1	1/10	h_1 =F (why?)	F correct: 1/16	h_2 = B		h_3 = I
w2	1/10	Err =2/10	F incorrect: 4/16	Err = 4/16		Err =
w3	1/10	α =	1/16	α =		α =
w4	1/10		1/16			
w5	1/10		1/16			
w6	1/10		1/16			
w7	1/10		4/16			
w8	1/10		1/16			
w9	1/10		1/16			
w10	1/10		1/16			
Err(B)	/10		4/16			
Err(D)	/10		7/16			
Err(F)	2/10 WHY?		8/16			
Err(I)	/10		7/16			

So we pick F as our first 'stump' - why?

B2 (5 points)

What is the resulting classifier that you obtain after three rounds of Boosting?

H(x) = Sign[(1/2 ln) * F(x) + (1/2ln) * + (1/2ln)*

Part C: Boost by Inspection (10 points)

As you become frustrated that you must have picked the wrong subset of classifiers to work with, one of the 6.034 TA's, Martin, happens to walk by and sees your answer to part A1. He reminds you why the boosting algorithm works and then tells you that there is no reason to actually run boosting on this dataset. A boosted classifier of the form:

$$H(x) = \text{Sign}[h_1(x) + h_2(x) + h_3(x)]$$

can be found which solves the problem. What three classifiers $\{h_1, h_2, h_3\}$ is Martin referring to, and why is the resulting H(x) guaranteed to classify all of the points correctly?

Michael E Plasmeier

From: Patrick Henry Winston <phw@MIT.EDU>
Sent: Sunday, December 04, 2011 4:53 PM
To: fa12-6.034@mit.edu
Subject: Important note on 6.034 end game

Friends,

Tomorrow's lecture, given by Professor Nancy Kanwisher, from the Department of Brain and Cognitive Science, will address where in your brain you think various sorts of thoughts.

A substantial part of Quiz 5 on the final will come from material presented in the remaining lectures, especially this one. If you show up, and pay attention, you will do well, but because the material is not yet available in textbook or note form, you will likely find parts of Quiz 5 mysterious. References will be supplied, insofar as practicable, but the coverage will be neither complete nor efficiently connected to the lectures.

Regards,
Patrick

Quiz 4 Wed
Quiz 5 only on final

--
Professor Patrick H. Winston
Massachusetts Institute of Technology
Room 251 | 32 Vassar Street | Cambridge, MA 02139
Email: phw@mit.edu | URL: http://people.csail.mit.edu/phw/ | Voice: 617.253.6754

6.034 Lecture 12/5

Guest lecturer: Nancy Kanwisher

Functional Specificity in the Human Brain w/ a fMRI

* will be part 3 on the exam

MIT intellegence initative

- coooperation

Franz Joseph Gall

brain seat of mind

disinct regions faculties

Franology - feel bums on brain

Flourens - attacked Gall

- no specific regions

Broca - saw damaged brain - able to associate

Today: basic agreement on simple regions

②

Are higher level processes specific?

Why do we care?

- one of the most fundamental qu
- makes possible a divide + conquer research strategy
- lets us ~~see~~ better understand computatrtation
    - can closer copy humans

Various ways to investigate

- brain imaging
    - L need to send oxygen when brain is processing
    - blood flow to that region ↑
    - FMRI looks at changes in blood flow
- put face upside down
    - much harder than words upside down

<u>FMRI</u>

- ~~a~~ looking at dot is basically off
- then show faces or objects
  do any parts of the brain diff between the 2?
- l + r swapped

- Don't belive blobs
  - L lots of things that prodces blobs

Could it be other things?

- feels / emotions to person
- is it seeing or recognizing?
  - very hard qu to answer
- lots of alt. hypothesis

Need to test the other hypothesis
- none of them work

Same image upside down → very different!

Does not respond to people's bodies or hands
L w/o heads

Get an intermediate response for ☺

Also found PPA
- responds to <u>places</u> - spacial layouts you can be in
- empty room = strong response

Also EBA – responds to bodies + body parts

– but not faces

– including stick figures

└ if just same lines randomly arranged – doesn't exist

Found in virtually all normal brains same place

Raises Questions

– Specificity – Are they engaged in a specific mental process?

– Origins – How do they get wired up/placed in development?

– Generality – How much of the brain is ~~general~~ specific?

Specificity

Face area also reporting somewhat on object

fMRI ~~does~~ can only see millions of neurons

Also seen in monkies

└ can stick a electron in to measure specific neuron

Data from monkies; appears more responsive

Do we need it for object recognition?

└ can't tell causal relationship

5

Can study from patients w/ brain damage
- L person able to recognize objects but completly unable to recognize faces

Face Recognition - same or different?
- L is a famas face Familiar?
  - L but <u>not</u> asking their name

Can temp turn brain area off
- L <u>Transcranial Magnetic Simulation</u>

- but face area too far from skull
- can reach a second smaller area
- performance on face ~~[illegible]~~ unchanged
- but ~~[illegible]~~ change on body perception for EBA
- face change on FBA

- TMS is crude - amazing it works at all

When doing brain surgery - when simulate face area
patient said - just for a minute you looked differently

kids - same at age 5
- very good at face recognition

Face recognition genetic?
- different in identical twins and fraternal twins
- or are you ~~recognizing~~ social so you learn faces

Can test 1-3 day old interns
- see how long they look
- if less time → then its the same
- 1-3 day olds are good at face recognition
  - even different angles
  - but not upside down

Also w/ monkies who never saw faces
- same as regular adult monkies

All this shows very strong gene role

Bt some things w/ experience

- reading is fairly recent in humans
- natural selection has not ~~sha~~ let reading area grow

Areas stronger when people read different languages

Many open qu

- Why do some things get own areas
- Can regions "move over"
  (after injury
- How do areas work together for real ~~d~~ world

(could not copy rest)

6.034 Recitation

12/8

1. handout

Quiz back

	Thorough	Adequate
P1	≥39	≥34
P2	≥41	≥37
P3	≥8	≥6
	≥88	≥77

1. Naive Bayes Example
2. Google translate + Bayes
3. Be smooth

---

1. Bayes $$P(B|A) = \frac{P(B) \times P(A|B)}{P(A)}$$

2. Chain Rule $$\frac{P(A \cap B)}{P(B)} = P(A|B)\,P(B)$$

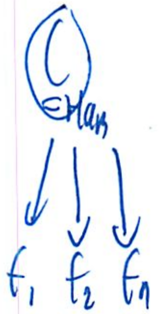
3. Conditional independence $$P(f_1, \ldots, f_n | c) = P(f_1|c) \cdot P(f_2|c) \cdots P(f_n|c)$$

<u>if</u> f_i are ind from each other

②

↙ features conditionally ind from each other

Naive Bayes - find max prob of label (cat) given features $f_1, \dots, f_n$ assumed ind



$$P(c \mid f_1, \dots, f_n) = \text{argmax} \frac{\overset{\downarrow \text{prior}}{P(c)} \cdot P(f_1, \dots f_n \mid c)}{P(f_1, \dots, f_n)}$$ by Bayes

arg max c

~~argmax~~

iterate of all cs. See which finds maximum

can rewrite w/ Bayes (see above)

$$= \overset{\text{argmax over } c}{\wedge} P(c) \cdot P(f_1, \dots f_n \mid c)$$

$$= \underset{c}{\text{argmax}}\ P(c) \left[P(f_1 \mid c)\, P(f_2 \mid c) \dots P(f_n \mid c) \right]$$

↖ since [illegible] conditionally ind

③

Example =30 student

Dorm = {East, West, FSILG}

↑C

3 qus

- Py/o →T →F
- Foreign Lang →T →F
- Good Shape →T →F

	Py/o	FL	GS	#s
East	8/10	1/10	3/10	10
West	3/10	6/10	3/10	10
FSILG	1/10	3/10	8/10	10

← does not add to 10

↑adds to 10

$P(C) = 1/3$ prior prob - knowing nothing

$\frac{10}{30}$

(4)

Suppose new student
$$\begin{aligned} Pyro. &= \text{true} \\ FL &= F \\ GS &= F \end{aligned}$$

which dorm ~~she~~ should they be?

(1) $= P(C=\text{East}) \cdot \left[P(\text{Pyro} \mid EC) \cdot P(\neg FL \mid EC) \cdot P(\neg GS \mid EC)\right]$

$$= \frac{1}{3} \cdot \left[\frac{8}{10} \cdot \left(1-\frac{1}{10}\right) \overset{?\text{not}}{\cdot} \left(1-\frac{3}{10}\right)\right]$$

= prob that you are in EC Given your responses to survey

(2) $= P(C=\text{West})$

$$= \frac{1}{3} \cdot \left[\frac{3}{10} \cdot \left(1-\frac{6}{10}\right) \cdot \left(1-\frac{3}{10}\right)\right]$$

(3) $= P(C=\text{FSILG})$

$$= \frac{1}{3} \cdot \left[\frac{1}{10} \cdot \left(1-\frac{3}{10}\right) \cdot \left(1-\frac{8}{10}\right)\right]$$

Now multiply each through

Find largest value → EC.

So student most likely from EC

If ans all true

∟ <u>shortcut</u> can just look at table

See WC has largest

But how to actually get these values?

Google can detect which lang you are typing. Uses nieve Bayes.

It has a lot of examples of text in various language.

Google has a lot of text to do this right

So how does Google translate?

English ——→ French

French ——→ English

Sentance: George Bush is not an idiot

Result: G.B. n'est pas un idiot"

But if

Sentance = G.B n'est pas un idiot

Result: GB is an idiot

Also

Sentance: Le pomme mange le garcon
(Apple eats the boy)

Result: Boy eats the apple

Using Bayes rules w/ lots + lots of examples
↳ no real understanding of language

$$P(\underset{\uparrow \text{ English sentence}}{E} \mid \underset{\uparrow \text{ french input}}{F^*}) = \text{argmax } E \; P(E) \cdot \underbrace{P(F^* \mid E)}_{\text{lang model}}$$

So why does the wrong example come out
Since George Bush <u>is</u> an idiot appears <u>much</u> more frequently!

$$\left.\begin{array}{l} P(E') = \text{Is idiot} \\ P(E'') = \text{Is NOT idiot} \end{array}\right\} 10^6 \text{ more frequently likely to appear}$$

They try to patch these.
But sheer size eludes them

8

n-grams $P(E)$

$\overset{w_1}{\text{The}}$ $\overset{w_2}{\text{sky}}$ $\overset{w_3}{\text{is}}$ $\overset{w_4}{\text{blue}}\overset{w_5}{.}$

Ask $P(\text{sky} \mid \text{Previous word} = \text{The})$

Calc $$\frac{P(\text{sky} \mid \text{P.W.} = \text{The})}{P(\text{sky})}$$

Larger than $P(\text{The} \mid \text{sky})$ ← bigram
2-gram

$$P(\text{blue} \mid \underbrace{\text{the sky is}}_{\text{4 gram}})$$

Google has 5-gram and 6-gram for many pairs of languages

F	le	cie	est	~~blue~~ bleu
E	the	sky	is	blue

V = # of distinct vocab / word types

36k – 40k for English speakers

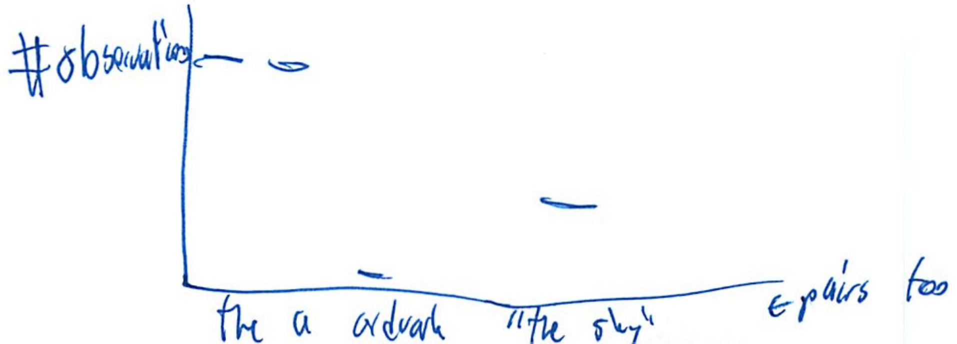
kids 5–6 learn 10–12 words every day

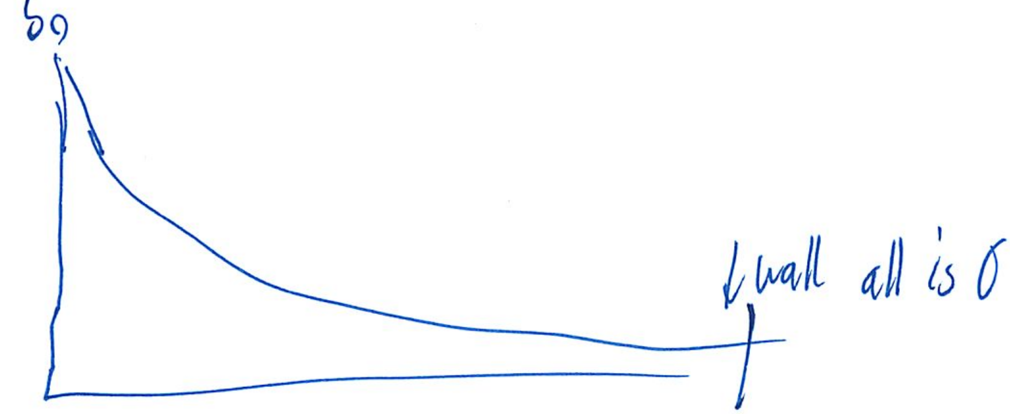
So 5-gram = $V \cdot V \cdot V \cdot V \cdot V = V^5$

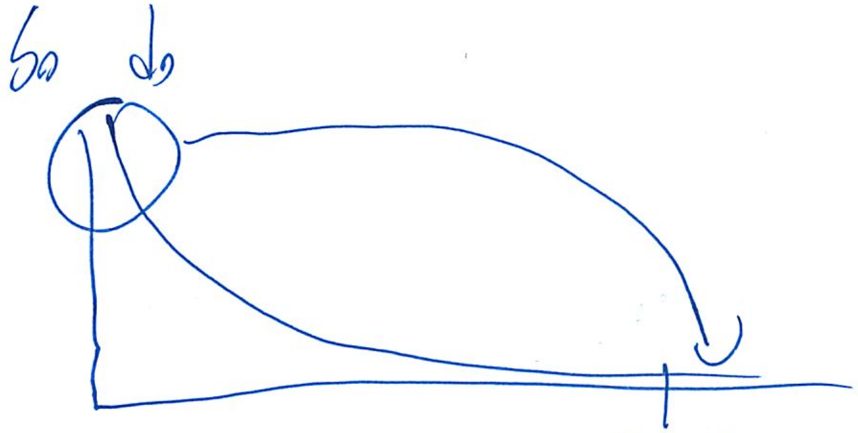
But if one is 0 – then whole multiple is 0

↳ Need a fix → Smoothing

Sometimes called Robin Hood Solution







Rob from righ to give to poor

Called add-1 smoothing

~~[illegible]~~ count(w_i) + 1

? for all counts (even seen)

Re normalize

$$\frac{\text{Count}(w_i)+1}{N+V}$$

Next big leap - by Turing during WW2

- P(U boat in location where never observed)
- figured out a clever ~~[illegible]~~ method

11

≈ # of times something appears uniquly
↑only once

before that it never appeared

then it appeared once

then not again

---

related to [illegible] binomial theorm

Called good - Turing fix

He teaches 6.863 J natural lang

6.049 / 7.33 Evolutionary biology

Final

There will be a pourperi - Question 3

Massachvsetts Institvte of Technology
Department of Electrical Engineering and Computer Science
6.034 Recitation 12, Thursday, December 8

Naïve Bayes & the Holy Grail — Prof. Bob Berwick, 32D-728

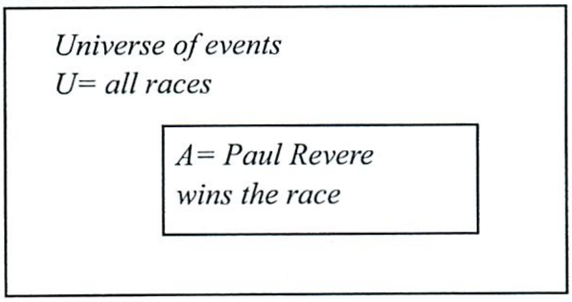
Agenda:
0. Probability revie
1. Naïve Bayes: another classifier (used for, e.g., Spam Asssasin)
2. How Google does translation
3. Beyond naïve Bayes: the maximum entropy stewpot

0. Basics of probability (review & pictures)

The fundamentals of probability theory: the axioms of probability. Why are these important? The power of the purse: Because while there are *other* attempts to handle the notion of 'uncertainty', e.g., 'fuzzy logic', '3-valued logic', etc., these axioms are the **only** system with the property that **if** you **gamble** with them, you **cannot** be unfairly exploited by an opponent who uses some other system (Di Finetti, 1932 theorem).

So, some first concepts.
We say that ***A* is a random variable** if *A* denotes an event and there is some uncertainty if *A* is true.
Typically, we let *U* denote the **universe** of all possible events (= all "possible worlds"). Then a subset of *U*, call it *A*, corresponds to the set of events in which *A* is true.

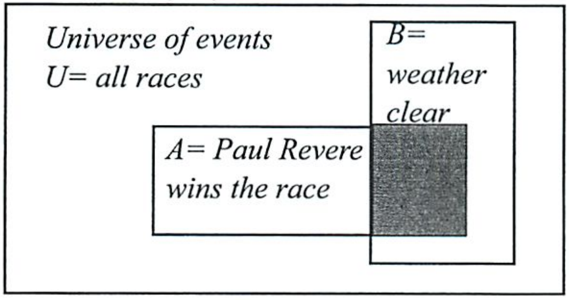
Example. Let the universe *U* be the set of all horse races. Let *Paul Revere* (abbreviation: P-R) be a horse. Then we can let *A* denote the set of racing events in which Paul Revere wins. We can draw this as a picture, where *races* labels the outer square, the universe, and the circle inside is the set of all events where Paul Revere wins the race:



Let us denote by ***P(A)*** the fraction of events (possible worlds in the universe of events) in which *A* turns out true. We could spend the next 2 hours on the philosophy of possible worlds and this business. But we won't.
We will compute probabilities using an informal notion of *areas* (formally, we'd use measure theory).
The Universe of all events has total area 1, $P(U)=1$, because it denotes all the events that are true. $P(A)$ then is the area of the smaller rectangle with respect to *U* (= the fraction of the total universe in which Paul Revere wins). $P(\neg A)$= the races in which Paul Revere does **not** win = the set difference between *U* and *A*. From this we will posit 3 axioms regarding $P(A)$:

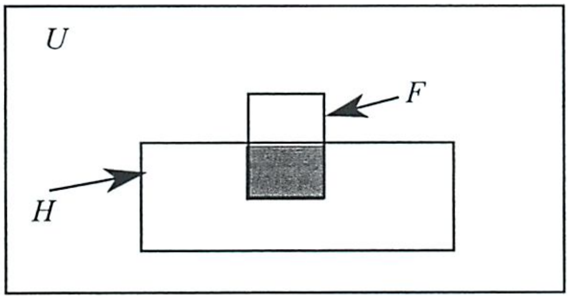
(1) $0 \le P(A) \le 1$ [because: the area of *A* cannot be < 0 or > 1]
(2) $P(true)=1$
(3) $P(false)=0$
(4) $P(A \vee B) = P(A) + P(B) - P(A,B)$ [where ∨ means "or", i.e., **either** *A* or *B* must be true; + means "add together", and the comma in *A, B* means "and", i.e., both *A* **and** *B* must be true]

To see how this last axiom works, let's look at the racing universe with event *A*= Paul Revere wins and a second event, *B*= the weather is clear. The **shaded area** represents the fraction of events when **both** *A* and *B* are true, i.e., *P(A,B)*= true:



It should be apparent that in order to figure out the probability of *A* **or** *B*, we need to add up the areas corresponding to *A* and to *B*, but then subtract out the shaded area so that it is not counted twice. In this way, we arrive at the formula for the probability of *A* **or** *B*.

We next turn to the notion of **conditional probability.**
We let *P(A|B)* denote the fraction of events/possible worlds in which *B* is true, and then *also* have *A* true. That is, we 'shrink' the universe from *U* down to *B*, focusing in on a subset possibly more relevant to our situation, and use *that* as our basis to calculate probabilities.
Example. In the figure below, we illustrate the following situation. Let *H*= probability that "I have a headache"; *F*= probability that "I am getting the flu". These are denoted by the rectangles *H* and *F* in the figure below. Let us assume:
P(H) = 1/10; *P(F)*=1/40. Now let's compute the conditional probability *P(H|F)*, i.e., the probability that I have a headache **given** that I have the flu. This is the fraction of flu-events that are also headache events – that is, if we just look at the rectangle *F*, what proportion of *F* overlaps with *H*? (The answer is 1/2). Thus, *P(H|F)*=1/2.



In other words, to find *P(H|F)*, we compute:
(# worlds in which *H* **and** *F* are true)/(# worlds in which *F* is true) or,
(area *H* **and** *F*)/(area of *F*), or
P(H, F)/*P(F)*

So this is the **formula for conditional probability:**

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

Note how *P(B)* is in the denominator here. Multiplying out, we obtain the important formula called the **chain rule** which we will uses in the naïve Bayes classifier:

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

Some other manipulations of conditional probability will be used in what follows. We consider two: (i) simplifications to the *right* of the conditioning bar symbol |; and (ii) simplifications to the *left* of the conditioning bar symbol.
Simplifications to the *right* of the bar:
Suppose we have *lots* of conditions to impose on whether or not Paul Revere wins. For example, this could depend on not only if the weather's clear, but also whether the jockey's brother is a friend of mine, whether Paul Revere won its last race, etc. In other words:

P(Paul Revere wins | weather clear, jockey's brother a friend, P-R won last race)

With more factors then, we have less *bias*, because we are focusing in on our particular situation, but we will have more *variance*, because it will become harder and harder to measure all these terms perfectly. So, sometimes we will want to reduce the number of factors to the right of the conditioning symbol to those we are more confident we can estimate; this is called *back off.* (We will see this in action soon). There is no problem in simply doing this:

P(Paul Revere wins | weather clear, ~~jockey's brother a friend~~, ~~P-R won last race~~)

And then of course just having *P*(Paul Revere wins | weather clear) remaining. But what about if there are more terms to the *left* of the bar, as in this case:

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)

Note that if we *add* terms to the left the probability should get lower and lower every time we add a new factor. (Why? Think about intersection.) If we just care about Paul Revere, are we then allowed to simply strike out the other two horses, this way?

P(Paul Revere wins, ~~Valentine loses~~, ~~Epitaph loses~~ | weather clear)

The answer is: No! We need to carry out a more complex expansion to isolate Paul Revere on the left. To see how, let's abbreviate Paul Revere wins as *R*, Valentine loses as *V*, Epitaph loses as *E*, and the Weather is clear as *W*. Then our conditional probability:

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)

Can be abbreviated as:

$$\frac{P(R,V,E,W)}{P(W)}$$

We can use this formula to derive the **chain rule for conditional probability:**

P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear)=
P(Paul Revere wins| Valentine loses, Epitaph loses, weather clear) ×
P(Valentine loses | Epitaph loses, weather clear) ×
P(Epitaph loses | weather clear)

Proof. Writing out the 3 terms:

$$\frac{P(R,V,E,W)}{P(W)}=\frac{P(R,V,E,W)}{P(V,E,W)}\times\frac{P(V,E,W)}{P(E,W)}\times\frac{P(E,W)}{P(W)}$$

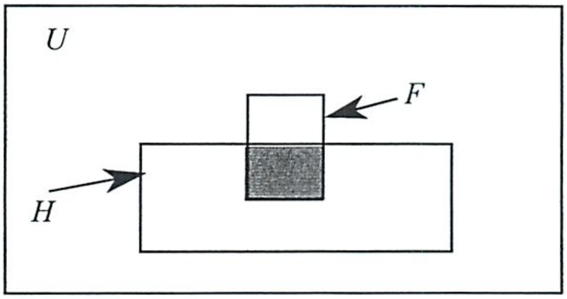
Now, supposed it is the case that the following simpler expansion holds:
P(Paul Revere wins, Valentine loses, Epitaph loses | weather clear) =
P(Paul Revere wins| ~~Valentine loses, Epitaph loses~~, weather clear) ×
P(Valentine loses | ~~Epitaph loses~~, weather clear) ×
P(Epitaph loses | weather clear)

In this case, whether Paul Revere wins or not depends *only* on whether the weather's clear...and not on what the other two horses do. They are irrelevant factors, so we can strike them out. In this case, when the probability is *unchanged* when we drop out conditioning factors, we say that the probability is **conditionally independent** (independent of the other horses, but still conditioned on the weather). More generally, if there are *n* factors *f*, and each factor is independent of the other, but still dependent on a condition *c*, we can write the following, which will be another key ingredient in our naïve Bayes classifier model:

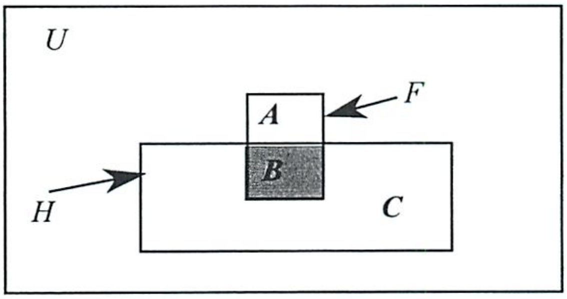
$$P(f_1,\ldots,f_n \mid c) = P(f_1 \mid c) \times \ldots \times P(f_n \mid c)$$

That is, we can just write out the probability as the product of the *n* factors, **assuming they are independent from one another** (the outcomes of these events do not affect the outcomes of one another); note the factors are still dependent on the outcome of event *c*.

OK, we come to the last ingredient we shall need, **Bayes' Law**. Again we can illustrate this with the simple picture of headache and flu as before. Recall *P(H)*=1/10; *P(F)*=1/40, *P(H|F)*=1/2.



Now we will **label** each of the distinct regions in this diagram, *A, B,* and *C*, as follows. *A*+*B*=area of *F*; *B*+*C*= area of *H*:



By the definition of conditional probability, *P(H|F)*= *P(H,F)*/ *P(F)* = *B*/(*A*+*B*).
Now consider this reasoning: one day you wake up with a headache, and you think, OMG, 50% of flus are associated with headaches, so now I have a 50-50 chance of getting the flu." Is this reasoning correct?

What we *want* to compute is: *P(F|H)*. We already know the *other* conditional probability, that of headache given the flu. Further, by the definition of conditional probability, in terms of the regions *A, B,* and *C*, we have that: *P(F|H)* = *B*/(*B*+*C*). To find this last ratio of regions, we can take the conditional probability *P(F|H)* = *B*/(*A*+*B*), and multiply it by (*A* +*B*)/(*B* + *C*), as follows:

$$\frac{B}{B+C} = \frac{B}{A+B} \cdot \frac{A+B}{B+C} \text{ i.e.,}$$

$$P(F \mid H) = P(H \mid F) \bullet \frac{P(F)}{P(H)}$$

$$\text{in our example, } \frac{1/2 \times 1/40}{1/10} = \frac{1/80}{1/10} = \frac{1}{8}$$

The term *P(F)* is called the **prior probability** (of getting the flu); the term *P(H|F)* is called the **likelihood**; the term *P(H)* is the **evidence** (e.g., that you have a headache); and the term *P(F|H)* is called the **posterior probability** of getting the flu (given that you have a headache). So this updated probability is a kind of learning: given the fact (data) that you indeed have a headache, how does the probability of getting the flu change? (It increases from 1/40 to 1/8.) Inverting from *P(H|F)* to *P(F|H)* is called **Bayes' Law.** It follows from a very simple manipulation of the definition of conditional probability and then application of the chain rule, i.e., that *P(A,B)*= *P(A|B)*×*P(B)*:

$$P(B \mid A) = \frac{P(A,B)}{P(B)} \text{ (by dfn of conditional probability)}$$

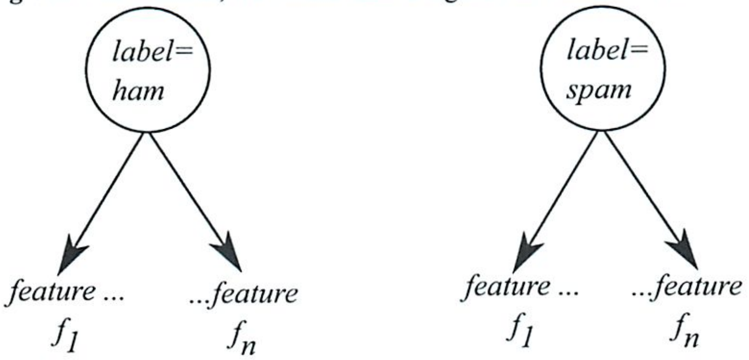
$$= \frac{P(A \mid B) \cdot P(B)}{P(B)} \text{ (by chain rule, replacing } P(A,B))$$

Or in words we can say this:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Now let's put this all to work to build a classifier called **Naïve Bayes**. Like k-means and ID-trees, and Boosting, etc., this will take as input the values of some **features** and then output a classification **label.**

As our example, we will use the common, but valuable task of classifying email into 1 of 2 categories: either good email ("ham") or bad email ("spam"). The underlying probability model follows what is called a **Bayes' net**. We can imagine the following generative process: we pick a label, e.g., "ham", and given this label, email documents of this type will have a certain distribution of feature values $f_1, \ldots, f_n$. If we pick the other label, "spam", we will get another distribution for the feature values (hopefully distinct). So the picture looks like this, and the idea of course is that **given** a **new** email, we would like to figure out whether it is ham or spam:



Crucially, we assume that **the features are independent from one another.** (This is the "naïve" part of Naïve Bayes.) Their values depend on (are conditioned on) **only** the value of the label. That is why we draw the networks as above, with **no links** between the features, only from the label directed down to the features.

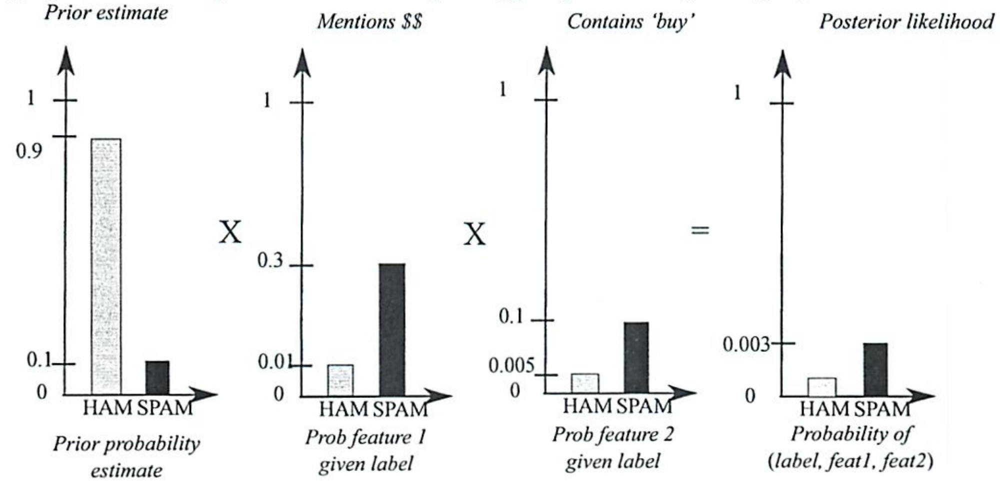
Now here's the idea behind the classification.. Suppose we have estimated that 90% of our email is "ham" (OK), and that 10% is "spam". This gives us our **prior probability estimates** *P(label=ham)*=0.9 and *P(label=spam)*=0.1. That's what we can say about any new email **without any additional information.** (We'll see below how we get these estimates.)

Now, when we get a new email, we will get the values of its **features** and use these to adjust the prior probabilities, as with our headache example. (In our example, to keep things simple, we will use only two features.)

So, this new email comes along: "*Buy this amazing new Ginsu knife for only $39.....*"
Is this ham or spam? We'll assume that we use the following 2 features:

Feature 1: The email mentions money; this occurs in 30% of spam, and in 1% of ham
Feature 2: The email contains the word 'buy'; this occurs in 10% of spam, and in 0.5% of ham

We can picture our calculation as follows: our initial prior probabilities for each category are adjusted by **multiplying** the contribution **each feature** 'votes' (independently) as to how likely each category is. Then we pick the **most likely = biggest probability category** at the end:



So, in this case, our new email is classified as "spam" because this yields the largest posterior likelihood. Note how we got this value. It is simply this:

$P(label)\times P(f_1 \mid label)\times P(f_2 \mid label)=P(label,f_1,f_2)$ [recall from dfn of conditional prob that:

$\dfrac{P(label,f_1,f_2)}{P(label)}=P(f_1 \mid label)\times P(f_2 \mid label)$ IF f_1,f_2 are independent of one another]

In other words, we multiple the following out to find the label likelihood, and pick the biggest likelihood:

Prior probability of a label × Probability of feature contributions = Posterior label likelihood

In our case, for the two labels "ham" and "spam":

	Prior ×	Pr(feat1 ($)| l) ×	Pr(feat 2 ('buy')|l)	= Label likelihood
Ham:	0.9 ×	0.01 ×	0.005	= 0.000045 (log of this likelihood: –4.34)
Spam:	0.1 ×	0.30 ×	0.10	= 0.00303 (log of this likelihood: –2.52)

So, our email is more likely to be spam than ham. In fact, taking the ratios of the log likelihoods, –2.52/–4.32, the email is about 2 orders of magnitude (100x) more likely to be spam than ham. Recall that: (1) the features **must** be independent of one another; (2) we can add other features, of course…this is what a program like Spam Assassin can do, by training; and (3) one can use this method with lots more categories to **classify** documents (see the end of the handout).

Let's turn to justifying this approach probabilistically, as well as how we actually estimate the probability values above, via training, and highlighting some pitfalls.

First, why is this justified? We are computing the **maximum** probability that an input email will have a particular label (category), **given** that it has a particular set of features. We pick the label that maximizes: *P(l=value | observed features)*. Let's follow out this logic. We are maximizing the following quantity over label values:

$$\max P(label \mid features)=\max\frac{P(features,label)}{P(features)}\quad \text{[by dfn of conditional probability]}$$

But note that the denominator in the expression above, *P(features)* = $P(f_1, \ldots, f_n)$ is *constant* no matter what our choice of label value. So, to maximize the above quantity, it suffices to maximize the numerator:

$$\max P(features,label)=P(f_1,\ldots,f_n,label)$$

By the chain rule, this quantity in turn is just:

$$\max P(label)\times P(f_{1,},\ldots,f_n \mid label)$$

But given that the features are all independent of one another, this is the same as (recall our Paul Revere example!):

$$\max P(label) \times P(f_1 \mid label) \times \ldots \times P(f_n \mid label)$$
$$\max prior \quad \times 'vote' f_1 \quad \times \ldots \times 'vote' f_n$$

Putting this down as a formula, we have:

$$\arg\max_C P(C|f_1,\ldots,f_n) = \arg\max_C \frac{P(C)\prod_{i=1}^n P(f_i|C)}{P(f_1,\ldots,f_n)} = \arg\max_C P(C)\prod_{i=1}^n P(f_i|C)$$

This is exactly the computation we have carried out. It remains to figure out how we 'train' our classifier – that is, how do we get the various estimates of the probabilities above? The simplest thing is just to estimate them from counts in training text, that is, known examples of ham and spam emails. These are the so-called *maximum likelihood estimates*:

$$P(label = ham) = \frac{count\ (\#\ ham\ emails)}{count(total\ \#\ emails)} \qquad P(label = spam) = \frac{count\ (\#\ spam\ emails)}{count(total\ \#\ emails)}$$

$$P(f_1 \mid label = ham) = \frac{count\ (\#\ ham\ emails\ mention\ \$)}{count(total\ \#\ ham\ emails)}$$

$$P(f_1 \mid label = spam) = \frac{count\ (\#\ spam\ emails\ mention\ \$)}{count(total\ \#\ spam\ emails)}$$

$$P(f_2 \mid label = ham) = \frac{count\ (\#\ ham\ emails\ contain\ \text{'buy'})}{count(total\ \#\ ham\ emails)}$$

$$P(f_2 \mid label = spam) = \frac{count\ (\#\ spam\ emails\ contain\ \text{'buy'})}{count(total\ \#\ spam\ emails)}$$

So this is how we get the estimates. For example, if we have 1000 emails, 900/1000 are ham, and 100/1000 are spam. Of the 100 spam emails, 30/100 mention money, and 1/100 contain 'buy'. For ham emails, 1/100 mention money and 5/1000 contain 'buy'.

Note that as the # of data samples (amount of training data) increases, then our estimates should get better; one of the properties of the maximum likelihood estimates is that they will converge to the 'true' values as the amount of data goes to infinity. (The mean approaches the true average.) But, if the # of training examples is small, our estimate will be very lousy, and have more noise (variance); there are a variety of things we can do to improve this, but that's for a machine learning course.

A second worked example:

MIT decides to use surveys to determine how to sort students into into dorms. They decide to use Naive Bayes and survey data from current residents to classify where to put future students.

To collect this "training data:, they surveyed 30 random students.
Each surveyed student is asked to fill out a simple questionaire with 3 true/false questions.

0. Which dorm do you live in: {East Campus, West Campus, or FSILG}
1. Are a Pyro – i.e. do you enjoy performing feats with fire (or inadvertently trigger fire alarms)?
2. Are you a foreign student or do you like studying foreign languages?
3. Are you in Good shape?

Here are the results. It turns out that our random survey gave us exactly 10 students from each dorm group.

	Pyro	ForeignLang	GoodShape	# surveyed
East Campus	8/10	1/10	3/10	10 10/30
West Campus	3/10	6/10	3/10	10 10/30
FSILG	1/10	3/10	8/10	10 10/30

What can you do this data? We can use these counts to make estimates of the following probabilities:

$P(C)$ (*the prior probability of being in any dorm*)
$P(f_i \mid C)$ (*the likelihood of having one of the 3 features* ***given*** *being in a particular dorm*)

E.g. *P*(Pyro=*True* | *C*=East Campus) = 8 /10 *P*(Language = *True*|*C*= FSILG) = 3/10

Now we can use these probability estimates to classify new students by applying Bayes rule, i.e., our formula:

$$\arg\max_{C} P(C|f_1,\ldots,f_n) = \arg\max_{C} P(C)\prod_{i=1}^{n} P(f_i|C)$$

Question 1: where would a new student who loves foreign languages most likely be classified if they filled in their incoming survey as follows:

Pyro = *True*
ForeignLang = *False*
GoodShape = *False*

To do this, we compute *P*(*C*i| **Pyro**=*True*, **ForeignLang**=*False*, **Goodshape**=*False*) for all three possible campuses, and find the largest one! (That is what the "arg max" part means.)

For *C*= East campus:
argmax *P*(*C*=East | P=*T*, F=*F*, G=*F*)
= argmax *P*(*C*=East) * [*P*(P=*T* | *C*=East) *P*(L=*F*|*C*=East) *P*(G=*F*|*C*=East)]
= (10/30) * [(8/10) (1–1/10)(1–3/10)] = 1/3*[(8*9*3)/1000] = 1/3 * [216/1000]
= 0.072000

For *C*= West campus:
argmax *P*(*C*=West | P=*T*, F=*F*,G=*F*)
= argmax *P*(*C*=West) * [*P*(P=*T* | *C*=West) *P*(L=*F*|*C*=West) *P*(G=*F*|*C*=West)]
= (10/30) * [(3/10) (1–6/10)(1–3/10)]
= 1/3 * [3*4*7/1000] = 1/3 * [84/1000]
= 0.028000

For *C*= FSILG:
argmax *P*(*C*=FSILG | P=*T*,F=*F*,G=*F*)
= *P*(*C*=FSILG)*[*P*(P=*T*|*C*=FSILG) *P*(L=*FC*=FSILG) *P*(G=*F*|*C*=FSILG)]
= (10/30) * [(1/10) (1–3/10)(1–8/10)]
= 1/3*[(1*7*2*)/1000 = 1/3 * [14/1000] = 1/3[14/1000]
= 0.004667

The largest value for such a student (Pyros *true*, all other attributes, *false*) is **East Campus.**

Question 2. What about an all-round student who checks all the boxes in the incoming survey?
$P(C=? \mid \text{Pyro} = \mathit{True}, \text{ForeignLang} = \mathit{True}, \text{GoodShape} = \mathit{True})$

$$\begin{aligned}
&P(C=\text{East} \mid \text{P}=T, \text{F}=T, \text{G}=T)\\
&\quad = P(C=\text{East}) * [P(\text{P}=T|C=\text{East})P(\text{L}=T|C=\text{East})P(\text{G}=T|C=\text{East})\ ]\\
&\quad = 10/30* [(8/10)\ (1/10)(3/10)]\\
&\quad = 1/3 * [8*1*3/1000] = 1/3 * [24/1000]\\
&\quad = 0.008000
\end{aligned}$$

$$\begin{aligned}
&P(C=\text{West Campus}|\ \text{P}=T, \text{F}=T, \text{G}=T)\\
&\quad = \text{P}(C=\text{West}) * [P(\text{P}=F|C=\text{West})\ P(\text{L}=T|C=\text{West})\ P(\text{G}=T|C=\text{West})]\\
&\quad = 10/30 * [(3/10)\ (6/10)(3/10)]\\
&\quad = 1/3 * [(3*6 *3/1000) = 1/3 *[54/1000]\\
&\quad =
\end{aligned}$$

$$\begin{aligned}
&P(C=\text{FSILG}|\ \text{P}=T, \text{F}=T,\text{G}=T)\\
&\quad = \text{P}(C=\text{FSILG}) * [\ P(\text{P}=T\ |C=\text{FSILG})P(\text{L}=T|C=\text{FSILG})P(\text{G}=T|C=\text{FSILG})]\\
&\quad = (1/3) * [\ (1/10)(3/10)(8/10)]\\
&\quad = 1/3 * \quad [1*3*8/1000] = 1/3 \quad * [24/1000]\\
&\quad = 0.008000
\end{aligned}$$

The maximum C is **West Campus**.

In Naive Bayes, the $P(C=$ some value) is also known as the "prior". Knowledge about the prior probabilities can help us distinguish what proportion to assign to each class. In our case we got lucky and it just happened that each campus got 10 students, so the prior in this case is *Uniform*.

Estimation & its discontents

There is at least one particular case about estimating the probabilities from data counts that we should note. Suppose a particular count is actually 0 – that is, we *never* observe a particular feature associated with a particular label – this will happen especially if we keep adding more and more features. In this case, note that the entire probability product to find the likelihood will *all* be zero, just because one of the estimates is 0. So this is very bad!

There is a whole cottage industry devoted to fixing this problem, and it is called *smoothing*. It is basically the Robin Hood strategy: we rob probability mass from the rich and give it to the poor. In particular, the *simplest* smoothing strategy, invented by Laplace, is called *add–1 smoothing*: if a count is 0, we add 1 to it, so that, e.g., 0/100 goes to 1/100. (We must also *subtract* the appropriate probability mass, i.e., counts, from the *rest* of our estimates, so that the probabilities still add up to 1 in all.)

A second method of smoothing (probability mass redistribution) is due to Alan Turing. He figured this out when he was developing probability formulas for estimating the likelihood of finding German submarines in particular areas of the ocean. What if a submarine had *never* been observed in a particular spot? (Something that's actually quite likely!) Turing reasoned that a fairly good probability estimate of 'things never seen' would be quite close to the estimate of 'things seen *exactly* once'. This method, now called Good-Turing smoothing (only published until decades after WWII), works well but is finicky. There are whole books devoted to this subject, for machine learning and especially in natural language processing, where we quickly get word sequences never seen before.

One more thing. You may note that in our calculation we multiply together a (possibly long) string of probabilities, one for each feature. With a 1000 features, this value will quickly get very,

very small. So, the usual method is to operate in log space, where multiplication is just addition, so we can maintain accuracy. (That's why we used log likelihoods above.)

Beyond Naïve Bayes (Optional)

OK, this method is fine so far as it goes, but it can be improved enormously. Here we will just sketch one method, known as **maximum entropy classification** that can gobble down any set of features, even if they are not independent. Yet remarkably, as first shown by Jaynes (1957), it is the most probabilistically sound method of **combining diverse features.** It rationalizes the general notion of just 'scoring' features and adding them up. We won't prove this here, but just indicate the general approach, which is now broadly used in, e.g., figuring out the part of speech labels in text. (For instance, in the sentence, *police police police*, is the first *police* a Noun or a Verb?)

1. To begin, let's assume there are now 10 labels for documents, with categories *A, B, C, D, E, F, G, H, I, J.* (So, e.g., category *A* could be travel; *B* sports; *C* business; etc.) If we know this, and **no other information** then given an email *m*, what is our best guess for category *C* (business) given this email, i.e., $P(C \mid m)$?
The maximum entropy approach would claim it is 1/10: that is, we maximize the quantity in each of the 10 bins, uniformly, by spreading out the total probability mass of 1 among 10 bins.

2. Now suppose I tell you that 55% of all emails are in category *A*, travel? Now what is the quantity $P(C| m)$? I think it should not be too hard to see that *A* gobbles up 0.55 of the probability mass, leaving 0.45 to be distributed evenly over the remaining 9 categories, or 0.05 for each of the remaining categories, including category *C*, business. So the maximum entropy estimate for $P(C| m)$ is 0.05.

3. Now suppose I add *another* constraint: that *in addition* to the fact in (2), we know that 10% of all emails contain the word 'buy'. What is $P(C| m)$ now? This gets harder to visualize, so we'll write it out as a table, where the first row is the probability of containing 'buy' (which thus must add up to 0.1 of all emails), and second row is the probability of not containing 'buy', which we have labeled *other* (which thus must add up to 0.9). Once again following the maximum entropy idea, since we don't know anything else about the 'contains buy' row, we should distribute its 0.1 total *evenly* among the 10 bins, thus giving 0.01 to each. Next, since *all* of category *A* must add up to 0.55, and since the 'contains buy' cell holds 0.01, it must be that the cell in the row labeled *other* and in column *A* must have the value 0.54 (so that the column total is 0.55). That leaves 0.9–0.54 = 0.36 for the rest of the 9 bins in the *other* row. Once again, spreading this evenly, we get 0.36/9 = 0.04 for each of these bins (so that each column here adds to 05). Thus we have the following table:

	1	2	3	4	5	6	7	8	9	10
	A	B	C	D	E	F	G	H	I	J
buy	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
other	0.54	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

So, *why* is this called *maximum entropy*? You should realize that by spreading out the values evenly, we are *maximizing the entropy of the cell values*: $-p \log p$ summed over all entries is at a maximum. (Below we indicate *why* this is a good thing to do.) In any case, we are maximizing the entropy *subject to the constraints specified.* (We have two so far.)

4. So let's add one more constraint. Suppose that in addition, 80% of the 'buy' emails are in *either* category *A* or category *C*. Now we want to figure out $P(C| m)$. Gulp! This one is much harder to figure out – in fact, in general to do this, it is like spreadsheets, but we can indicate what has to be true in our table now: the probability of the *buy* row, column *A*, plus the probability in the *buy* row, column *C*, must add up to 0.08 (80% of the 10%). That turns out to be the values 0.051 and 0.029. Since that leaves 0.020 for the rest of the bins in the *buy* row, these must be 0.020/8=0.0025. Since column *A* must still add up to 0.55, then that leaves 0.499 for row *other*, column *A*. Since

the *other* row must still sum to 0.9, we have 0.9–0.499= 0.401 to distribute evenly over the rest of the *other* bins, so this is 0.401/9 = 0.0446. If we impose these constraints, you'll see that this is the answer (we don't say how we figured it out!)

	1	2	3	4	5	6	7	8	9	10
	A	B	C	D	E	F	G	H	I	J
buy	0.051	.0025	0.029	.0025	.0025	.0025	.0025	.0025	.0025	.0025
other	0.499	.0446	.0446	.0446	.0446	.0446	.0446	.0446	.0446	.0446

Now we know that *P(buy,C)*= 0.029; *P(C| buy)*= 0.29 (= 0.029/0.1); *P(A | buy)*= 0.51.
This is our classifier, a *maximum entropy* classifier.

The punchline. While there are many possible distributions that could yield the three observed constraints, that 55% of the emails are in category *A*, that 10% of the emails contain *buy*, and that of these 10%, 80% are in category *A* or *C*, the **one distribution** that we picked, where we have **maximized** the entropy of the probability mass subject to these constraints, turns out to be the **only one** having the following two properties, the second one quite remarkable:

1. This distribution follows the form: $P(email,label)=\frac{1}{Z(\lambda)}\exp\sum_i \lambda_i f_i(email,label)$ where the lambdas are the weights associated with each feature f_i; the function f_i returns 1 if the feature is in the email, and 0 otherwise; and *Z* is a normalizing constant to make sure the probabilities all add up to 1.
2. This distribution **maximizes the probability of the training data,** $\prod_j P(email_j,label_j)$

This is what justifies the method!

6.034 Mega R

12/9

Everyone does part 5

Not really time to do all 5

If near a cutoff, then might upgrade

Bayes

One of the newest topics

He thinks hard

- Probability difficult to understand.

  Humans don't have intuitive sense

Not all students agree

Silver Star

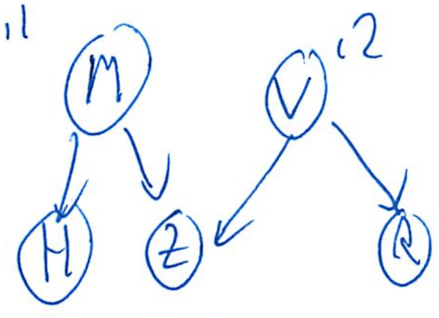
- ✱ Bayes' Rule: $P(A|B)P(B) = P(B|A)P(A)$
- ✱ Independence: If ind., $P(A|B) = P(A)$
- ✱ Expanding: $P(A) = P(A|B)P(B) + P(A|\neg B)P(\neg B)$
- ✱ Explaining away
- ✱ Naïvete

Bayes Things depend / are influenced by

Also $P(A \cap B) = P(A)\,P(B)$
~~Counting~~ if ind.

Expanding Away – lots of things are unrelated

M	H
F	.7
T	.8



V	R
F	.1
T	.3

M	V	Z
F	F	.01
F	T	.4
T	F	.5
T	T	.6

But don't really use #s since can leave stuff in terms of formula

Q) $P(M \cap H \cap Z \cap V \cap R) =$

Can use some of the previous rules to try + figure out
if all ind just multiply together
↳ but they are not!

But actually w/ this chart its easy
└ gives us # we need

H, Z are ind once we know M

$= P(H|M)\,P(M) \cdot P(R|V)\,P(V) \cdot \ldots$

$$\boxed{P(Z \cap \Omega) = P(Z|\Omega)\,P(\Omega)}$$

$$= \underbrace{\left(P(H|M)\,P(M) \cdot P(R|V)\,P(V)\right)}_{\text{already have}} \underset{\uparrow\ \text{don't need } P(M)P(V)}{P(Z|M,V)}$$

b) $P(Z|V) =$

We know $P(Z|V,M)$

Only want $P(Z|V)$

Use expanding

$= P(Z|V,M)\,P(M) + P(Z|V,\neg M)\,P(\neg M)$

④

c) $P(z) =$

expanding again

$= P(z|v)P(v)$

? but can't read that
but answer to previous

since M, V are ind

$$= \left(P(z|v,m)P(m) + P(z|v,\neg m)P(m)\right)P(v)$$

$$+ P(z|m,\neg v)P(m)P(\neg v) + P(z|\neg m,\neg v)P(\neg m)P(\neg v)$$

d) $P(v|z) =$

Use Bayes Rule

$$= \frac{P(z|v)P(v)}{P(z)}$$

e) $\overset{A}{P(v|z)}$ $\overset{B}{P(v|z,\neg m)}$ $\overset{C}{P(v|z,m)}$ Rank least to greatest

trickiest qv

CAB

We want to figue out virus

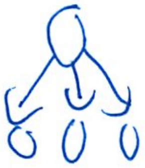
↳ if zombie - pretty likly is a virus

p(virus) goes up after seeing zombie

↳ if one cause is there, less of a chance of the other cause being there (explaining away)

Naieve Bayes

↑ trying to bild model that is obseralbe but make assumption that all dep. on world model



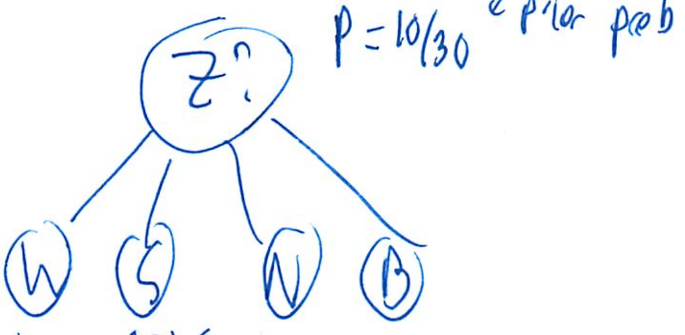
- Sounds restictive, but works supprisingly well

	W	S	N	B	#
Zombie	6	8	9	9	10
Healthy	10	15	12	1	20

↑ characteristics - all ind of each other

⑥

* Hidden world model - observable data



P = 10/30 ← prior prob

Z	W
Z	6/10
H	16/20

Z	S
Z	8/10
H	15/20

Z	N
Z	9/10
H	12/20

Z	B
Z	9/10
H	1/20

New guy Emric. Zombie or not?

we see that

W = T

S = T

N = T

B = F

Do math . . . $P(Z \mid W \cap S \cap N \cap \neg B)$

$P(\neg Z \mid W \cap S \cap N \cap \neg B)$

Not a zombie

$\rightarrow \frac{1}{3} \cdot \frac{9}{10} \cdot \frac{8}{10} \cdot \frac{6}{10} \cdot \frac{1}{10}$

$\rightarrow \frac{2}{3} \cdot \frac{12}{20} \cdot \frac{15}{20} \cdot \frac{10}{20} \cdot \frac{19}{20}$ ← larger

6.004 Quiz 4 Debreif 12/9

Actually thought I did pretty good

finished very early - 20 - 25 min left

Caching was hardest but I think I might of got it

rest I think I got

Prediction 24 - 28

6.034 Lecture

12/12

The Right Way

Five Hypotheses

The Right Way

- □ Inner Language Hypothesis
- □ Strong Story Hypothesis
- □ Directed Perception Hypothesis
- □ Social Animal Hypothesis
- □ Exotic ~~Animal~~ Hypothesis

Engineering

- * Talk
- * Look
- * Draw
- * Collaberate

(Notes very bad today
- tired
- lecture unstructured)

A Guest Lecture: Mim

Well what do you think + why

Applications of AI
- not engineering

Original goal of AI: understand human intellegence

(working on slides, will show afterwards)

What motivates him?

- Member of Navel Sci Board
- Visits orangitangs at a zoo
  - Using a tool

Why are we different?

Paleoanthropoligists

- we made same tools as naderitals for tens of thousands of years

in Southern Africa - we became different

- making jewlery
- " sculptures
- painting caves

Is it since we are simbolic?

Or significantly simbolic

Noam Chomsky

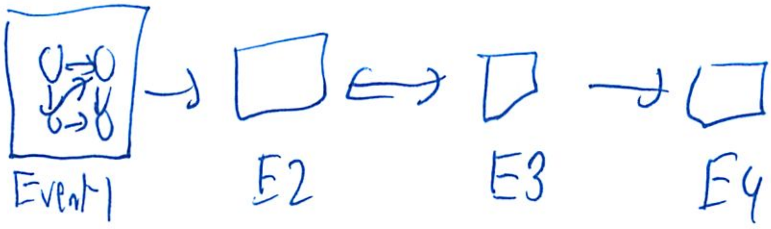
Combine concepts w/o limit

∞ly

~~A~~ (someone) Skokey?

- people + cats spping in a room
- whats different ~~is~~ we combine
  - info from diff pts of brain

Both ~~[illegible]~~ building syntatic nets
and be able to tie togeter
call each string a story



this ability is what makes humans different

Inner Lang Hyp

(see slides)

Strong Story Hypothesis

the mechanisms that enable . . .

~~th~~ (see slides)

Types of stories

- fairy tales
- religion
- law
- business (all case studies)
- Math (follow recipe - special case of story)

Other AI people should think story first

↳ makes the reasoning possible

What to do about it?

1. Characterize behavior
   - not a specific method yet
2. Formulate Computational problems
3. Propose Computational Solutions
   └ difference from a thugs typical psychologist
4. Exploratory Implementation
5. Principles

repeat

Commonsense lang

If someone kills you, then you become dead

Reflective lang

revenge XX's harming YY leads to YY's harming XX

If you ~~d~~ can't build it, you don't understand it
(? or was it other way around)

---

2 cultures thinking about MacBeth story
- 2 sep. personas └ knowledge bases
  rules
  reflective langs

6

Program reading background knowledge
Then reading story

I bias : revenge
~~western~~
Other : senseless violence
Rev.

One : situational Macduff crazy
Other : dispositional Something made Macbeth crazy

If-then rules make a graph
L Move back on graph to see what happens

Estonia

Moved a Russian war monument
So their network was hacked
Can system find out what happened?

One view : revenge
teaching you a lesson

Little thing on side looks at response
└ sees if lesser or more harm
– using political ~~dictionary~~ : Goldstien analysis
science scale

* Story is figured out above what the words say *

"Elaboration" graph
White – written
Grey – elaboration added

Most of story is us hallucinating – filling in the gaps

Minskey 6 levels of thinking

Book: Emotion machines

1. Self concious reflective thinking – what do others think about my revenge policy
2. Self reflective thinking – this will be revenge – I don't do that
3. Reflective thinking – how think about your thinking
4. Deliberative thinking – if I anger you, you'll kill me
5. Leared reflex – if I kill you, your' dead
6. Inate reflex

Genesis

What's New

1 - Story ~~[illegible]~~ alignment + analogy
- knowing someone knows ...

every ~~[illegible]~~ actor is rational
- just get a model of their decion making

alignment comes from bio
- adapted to deal w/ stories

Tet offinsive + egyptians
- both political, not military reason behind

2 Story telling story
- persona retells story to other persona
- knowing how it works
  ^ target propensy
- how much detail to include

It's the spoon feeding level

But How far to generalize

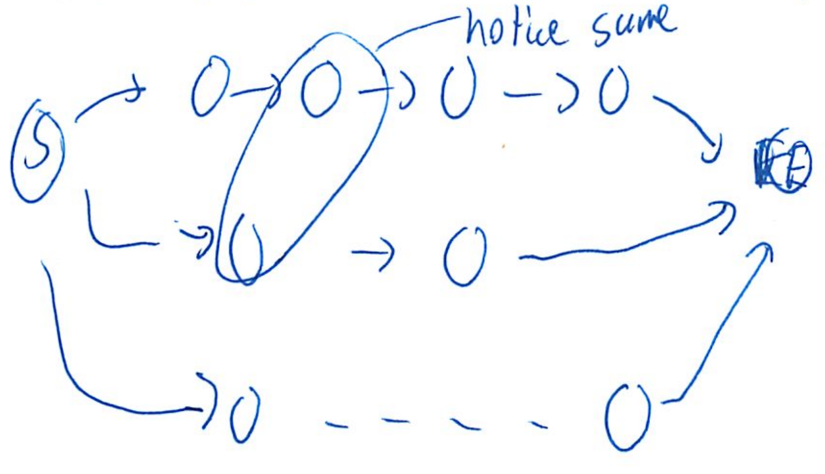
└ can give it the rules

National reconciliation after civil war

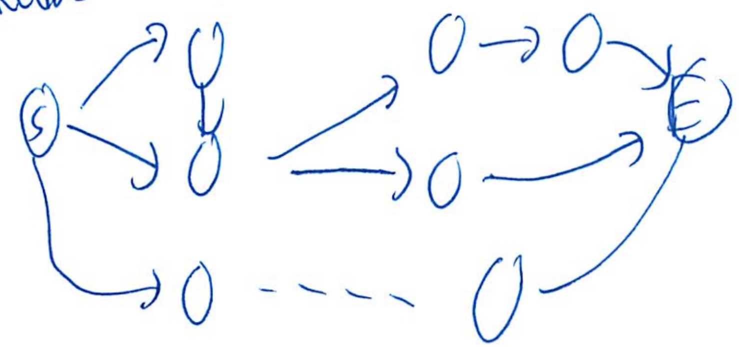
- need to ~~[illegible]~~ believe otherside has legitimate pov

3 Concept ~~[illegible]~~ discovery - we were told xx harms yy
yy harms xx is "revenge" when we were young

How do we ~~[illegible]~~ extract this concept from stories?



Redo ~~[illegible]~~ arrows



Now more paths

Think about the prob of taking each path

L can use Nieve Bayes

a priori based on size

Event sequences PHD thesis

Refine models over + over

<u>Social Animal Hyp</u> - we develop outer lang b/c of

- need someone to say it <u>to</u>
- it amplifies everything else
- we substitute education for genes

<u>Directed Perception Hyp</u> (see slides

Cat drinking

looks different from human drinking

Ask: How many countries in Africa ~~[illegible]~~ cross equator?

5?

need a map

11

Perceptual level
- visual processing
- learning to recognize a jump

But can also produce video to do that!

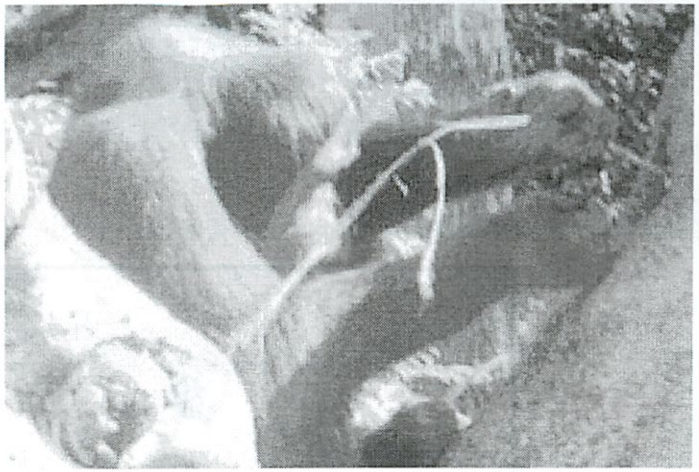
If this is the way we think

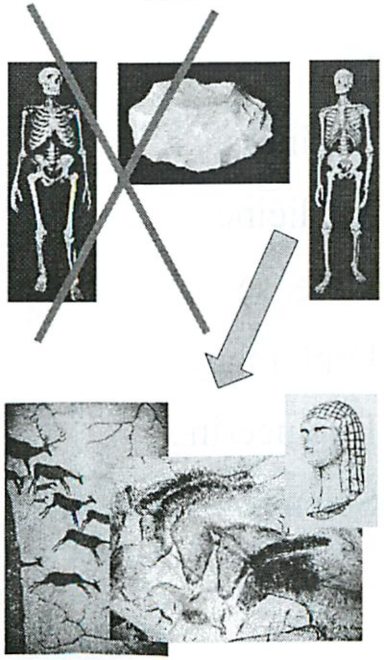
We make our selves smarter by talking
looking
drawing
Collebrate - engages everything

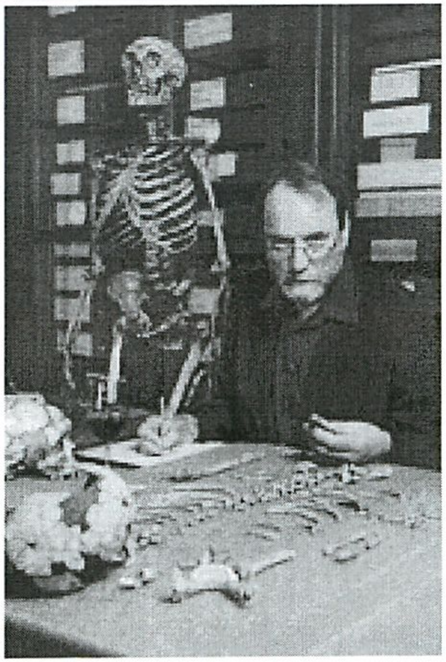
what I do
↓
take notes
(you won't look at)

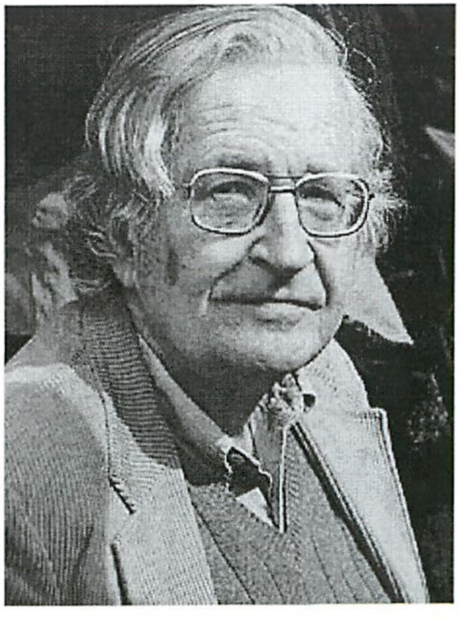
This subject makes you smarter. Do these things!

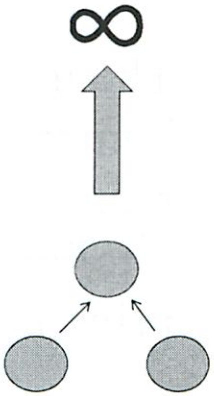
The Right Way: Five Hypotheses



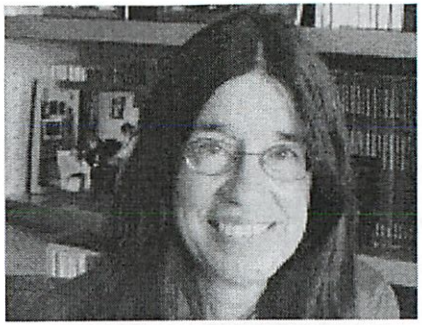


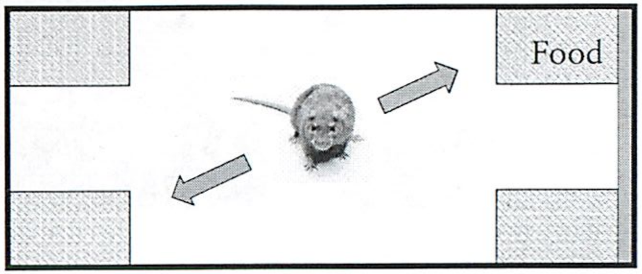






12/21





The Inner Language Hypothesis

We are different because we have a symbolic inner language

The Strong Story Hypothesis

The mechanisms that enable us humans to tell, understand, and recombine stories separate our intelligence from that of other primates.

Fairy and folk tales
Religious parables
Ethnic narratives
History
Literature
Experience
News
…

Law
Business
Medicine
Defense
Diplomacy
Engineering
Science
…

The Strong Story Hypothesis

The mechanisms that enable us humans to tell, understand, and recombine stories separate our intelligence from that of other primates.

A thane is a kind of noble. Macbeth and Macduff are thanes. Lady Macbeth is Macbeth's wife and Lady Macbeth is greedy. Duncan, who is Macduff's friend, is the king, and Macbeth is Duncan's successor. Macbeth defeated a rebel. Macbeth's success made Duncan become happy. Witches had visions and talked with Macbeth. Duncan rewarded Macbeth because Duncan became happy. Lady Macbeth is greedy. Lady Macbeth is Macbeth's wife. Macbeth wants to become king because Lady Macbeth persuaded Macbeth to want to become the king. Macbeth murders Duncan. Then, Lady Macbeth becomes crazy. Lady Macbeth kills herself. Dunsinane is a castle and Burnham Wood is a forest. Burnham Wood goes to Dunsinane. Then, Macduff fights with Macbeth. Then, Macduff kills Macbeth. Macduff had unusual birth. The witches's predictions came true.

- Commonsense level:

```
If someone kills you, then you become dead.
```

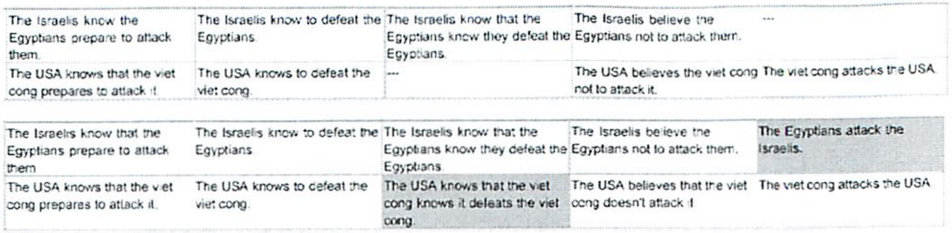
- Reflective level:

```
Description of "revenge":
xx's harming yy leads to yy's harming xx.
```

A thane is a kind of noble. Macbeth and Macduff are thanes. Lady Macbeth is Macbeth's wife and Lady Macbeth is greedy. Duncan, who is Macduff's friend, is the king, and Macbeth is Duncan's successor. Macbeth defeated a rebel. Macbeth's success made Duncan become happy. Witches had visions and talked with Macbeth. Duncan rewarded Macbeth because Duncan became happy. Lady Macbeth is greedy. Lady Macbeth is Macbeth's wife. Macbeth wants to become king because Lady Macbeth persuaded Macbeth to want to become the king. Macbeth murders Duncan. Then, Lady Macbeth becomes crazy. Lady Macbeth kills herself. Dunsinane is a castle and Burnham Wood is a forest. Burnham Wood goes to Dunsinane. Then, Macduff fights with Macbeth. Then, Macduff kills Macbeth. Macduff had unusual birth. The witches's predictions came true.

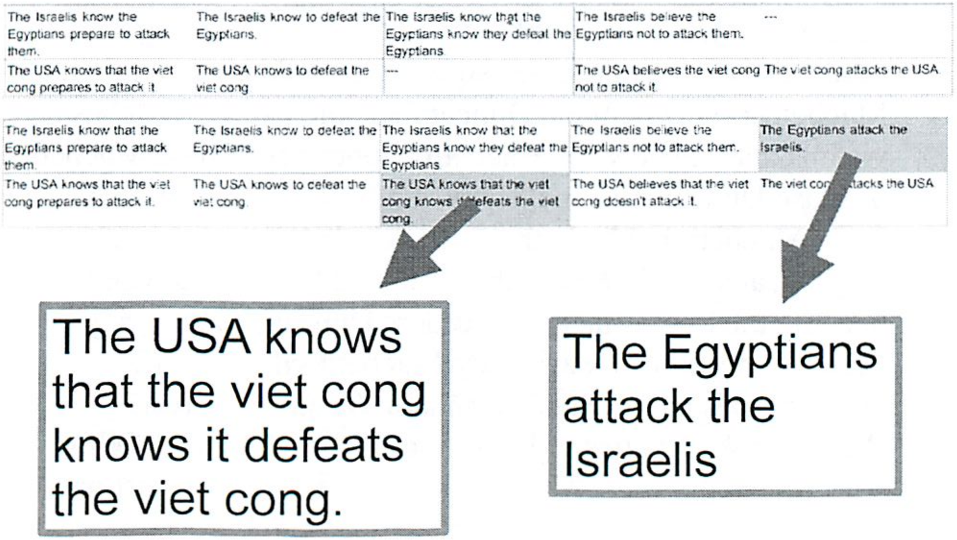
What's New

✓ Story alignment and analogy



The Israelis know the Egyptians prepare to attack them.	The Israelis know to defeat the Egyptians.	The Israelis know that the Egyptians know they defeat the Egyptians.	The Israelis believe the Egyptians not to attack them.	---
The USA knows that the viet cong prepares to attack it.	The USA knows to defeat the viet cong.	---	The USA believes the viet cong not to attack it.	The viet cong attacks the USA.

The Israelis know that the Egyptians prepare to attack them	The Israelis know to defeat the Egyptians.	The Israelis know that the Egyptians know they defeat the Egyptians.	The Israelis believe the Egyptians not to attack them.	The Egyptians attack the Israelis.
The USA knows that the viet cong prepares to attack it.	The USA knows to defeat the viet cong.	The USA knows that the viet cong knows it defeats the viet cong.	The USA believes that the viet cong doesn't attack it.	The viet cong attacks the USA.



The Israelis know the Egyptians prepare to attack them.	The Israelis know to defeat the Egyptians.	The Israelis know that the Egyptians know they defeat the Egyptians.	The Israelis believe the Egyptians not to attack them.	---
The USA knows that the viet cong prepares to attack it.	The USA knows to defeat the viet cong.	---	The USA believes the viet cong not to attack it.	The viet cong attacks the USA.

The Israelis know that the Egyptians prepare to attack them.	The Israelis know to defeat the Egyptians.	The Israelis know that the Egyptians know they defeat the Egyptians.	The Israelis believe the Egyptians not to attack them.	The Egyptians attack the Israelis.
The USA knows that the viet cong prepares to attack it.	The USA knows to defeat the viet cong.	The USA knows that the viet cong knows it defeats the viet cong.	The USA believes that the viet cong doesn't attack it.	The viet cong attacks the USA

The USA knows that the viet cong knows it defeats the viet cong.

The Egyptians attack the Israelis

What's New

- Story alignment and analogy

✓ Story telling story

Duncan is a person. Lady Macbeth is a person. Macduff is a person. Macbeth is a person. A thane is a noble. Macbeth is a thane. Macduff is a thane. Lady Macbeth is Macbeth's wife. Lady Macbeth is greedy. Duncan is the king. Macbeth is Duncan's successor. Duncan is Macduff's friend. Macbeth defeats a rebel. Appear is a success. Macbeth has a success. Witches talk with Macbeth. Witches have visions. Duncan rewards Macbeth because Duncan becomes happy. Macbeth wants to become king because Lady Macbeth persuades Macbeth to want to become king. Macbeth murders Duncan.

Duncan becomes dead. Macbeth becomes king.

Duncan is a person. Lady Macbeth is a person. Macduff is a person. Macbeth is a person. A thane is a noble. Macbeth is a thane. Macduff is a thane. Lady Macbeth is greedy. Macbeth defeats a rebel. Appear is a success. Macbeth has a success. Witches talk with Macbeth. Witches have visions. Duncan rewards Macbeth because Duncan becomes happy. Macbeth wants to become king because Lady Macbeth persuades Macbeth to want to become king.

Duncan becomes dead because Macbeth murders Duncan. Macbeth becomes king because Duncan becomes dead, Duncan is king, and Macbeth is Duncan's successor.

Duncan is a person. Lady Macbeth is a person. Macduff is a person. Macbeth is a person. A thane is a noble. Macbeth is a thane. Macduff is a thane. Lady Macbeth is greedy. Macbeth defeats a rebel. Appear is a success. Macbeth has a success. Witches talk with Macbeth. Witches have visions. Duncan rewards Macbeth because Duncan becomes happy. Macbeth wants to become king because Lady Macbeth persuades Macbeth to want to become king. Macbeth murders Duncan.

Duncan becomes dead because if a person murders another person, the other person becomes dead.

- Spoon feeding
- Explanation
- Explanation with intervention
- X intervenes to prevent Y from acting
- X understands Y’s point of view
- X negotiates with Y
- X explains situation to Y in Y’s terms
- X teaches Y how to interpret situation
- X shapes Y’s reaction

What's New

- Story alignment and analogy
- Story telling story
- ✓ Concept discovery

In 1998, Afghan terrorists bombed the U.S.'s embassy in Cairo, killing over 200 people and 12 Americans. Two weeks later, The U.S. retaliated for the bombing with cruise missile attacks on the terrorist's camps in Afghanistan, which were largely unsuccessful. The terrorists claimed that the bombing was a response to America torturing Egyptian terrorists several months earlier.

In early 2010, Google's servers were attacked by Chinese hackers. As such, Google decided to withdraw from China, removing its censored search site and publically criticizing the Chinese policy of censorship. In response, a week later China banned all of Google's search sites.

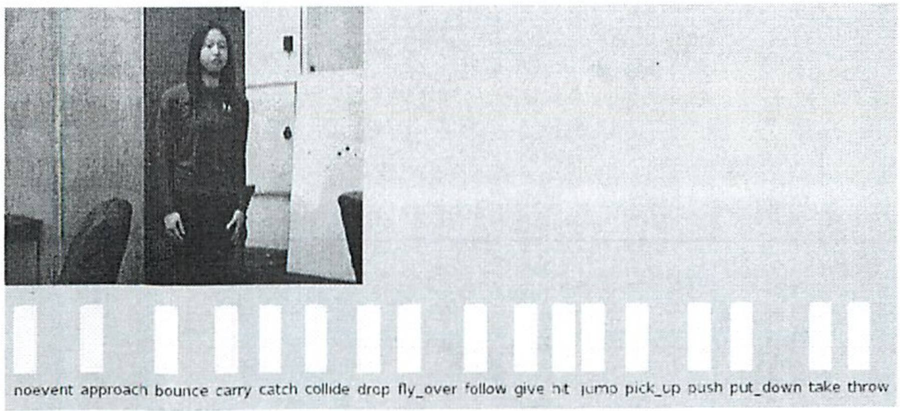
The Social Animal Hypothesis

We developed an outer language because we are social animals

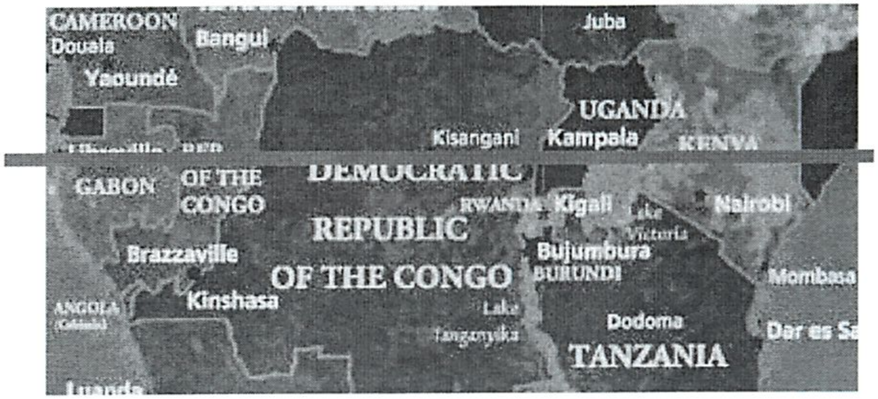
The Directed Perception Hypothesis

The mechanisms that enable us humans to direct and hallucinate with our perceptual faculties separate our intelligence from that of other primates.

The perceptual level



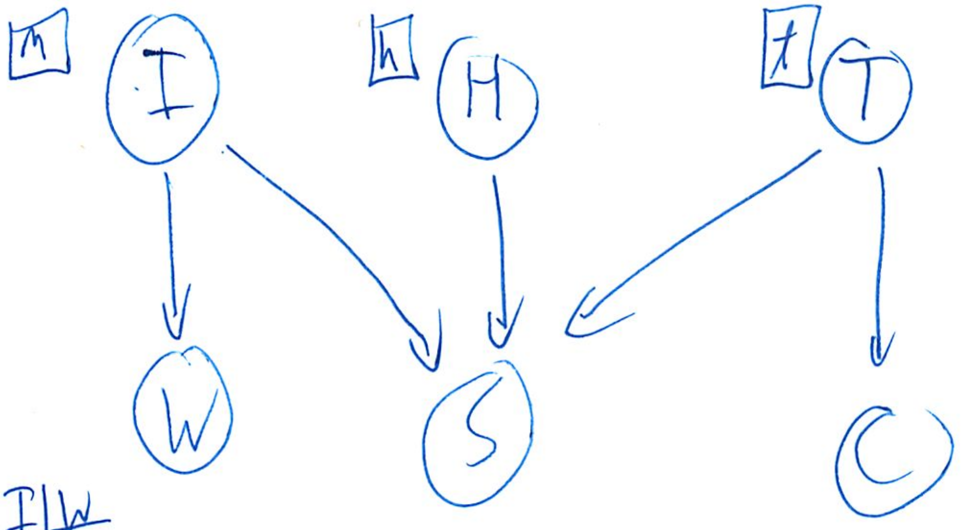
Thinking about the Equator



6.034 Tutorial

12/12

Quiz: people thought was fair
Final: each section will cover same lectures ~~set~~ - might be a diff one



I	W
F	v
T	w

I	H	T	S
F	F	F	a
F	F	T	b
F	T	F	c
F	T	T	d
T	F	F	e
T	F	T	f
T	T	F	g
T	T	T	h

T	C
F	j
T	k

TA: I'm not very good at probability

②

a1) $P(I, H, W, \overline{T}, S, C) =$

~~$= P(I)\, P(H)\, P$~~

$= P(I)\, P(H)\, P(\overline{T})\, P(W \mid I)\, P(S \mid I, \overline{T}, H)\, P(C, \overline{T})$

$= m\, h\, (1-t)\, w\, g\, j$

a2) $P(S \mid I) =$

$$=\cancel{\frac{P(S, I)}{P(I)}}$$ is a simpler way

~~$= e + f + g + s$~~ but Bayes rule - need parents ...

$= P(S, T, H) +$

$P(S, \overline{T}, H) +$

$P(S, T, \overline{H}) +$

$P(S, \overline{T}, \overline{H})$

③

$$= P(S \mid T, H)\, P(T)\, P(H) +$$
$$P(S \mid \overline{T}, H)\, P(\overline{T})\, P(H) +$$
$$P(S \mid T, \overline{H})\, P(T)\, P(\overline{H}) +$$
$$P(S \mid \overline{T}, \overline{H})\, P(\overline{T})\, P(\overline{H})$$

since we are assuming I is always true

c) $P(H \mid W) = \dfrac{P(W \mid H)\, P(H)}{P(W)}$

but they're ind.

$= P(H)$

each node only dependent on its decendents

but if wanted to prove

$$= \frac{P(W)\, P(\overline{H})}{P(W)}$$

4

New problem

S	M	L	MW	Dos

PC	.4	F=.1
		Sch=.7
		CW=.2
			K=.3
			C=.3
			MWF=.2
			PE=.2
				.3

MBC	.3	F=.4
		Sch=.4
		CW=.3
			K=.4
			C=.3
			MF=.4
			PE=.1
				.1

TDLC	.1	F=.3
		Sch=.3
		CW=.4
			K=0
			C=.5
			MF=0
			PE=.5
				.4

ST	.9	F=.8
		Sch=.1
		CW=.1
			K=.2
			C=.5
			MF=.1
			PE=.2
				.2

$\boxed{5}$

$P(S \mid \bar{M}, F, C) =$

Very similar to Zombie, Tatered clothes, Ate brains example
but not just true, false

$PC \begin{array}{|c} M \\ \hline .4 \end{array}$ → Means $T = .4$, $F = .6$ (Ohhhh!)

; all ind of S
No S is dep on each of the vaiables
but each of M, F, C are ind <u>of each other</u>
That is what nieve Bayes means!

We can't just read it off
Have all the probabilites in the other direction

So do Bayes rule

$$= \frac{P(\bar{M}, F, C \mid S)\, P(S)}{P(\bar{M}, F, C)}$$

We still can't pull #s off chart

$$= \frac{P(\bar{M} \mid S)\, P(F \mid s)\, P(C \mid S)\, P(S)}{P(\bar{M})\, P(\bar{F})\, P(C)}$$

Do for each
take the maximum

$$P(\bar{M}, F, C) =$$

$$= P(\bar{M}, F, C \mid S = PC) +$$

$$P(\bar{M}, F, C \mid S = MBC) +$$

$$P(\bar{M}, F, C \mid S = TDLC) +$$

$$P(\bar{M}, F, C \mid S = ST)$$

$$= P(\bar{M} \mid s)\, P(\bar{F} \mid S)\, P(C \mid S)\, P(S)$$

$$+$$

⋮ for each s

6.034 Lecture

12/14

Fini

(last lecture)

□ The Fifth Hypothesis

□ The Subject

□ The Final

□ What Next

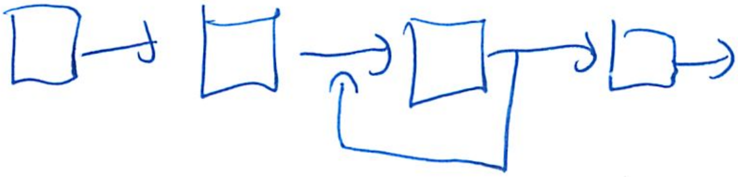
□ Powerful Idea

* Powerful Idea

Exotic Engineering Hyp

There is a kind of engineering in our heads which we are nearly not aware of

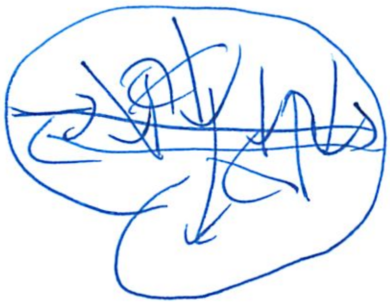
Normal systems



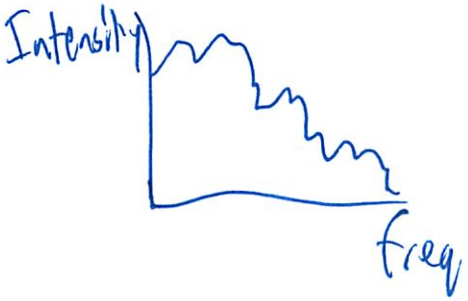
Textbooks

Simple model of brain

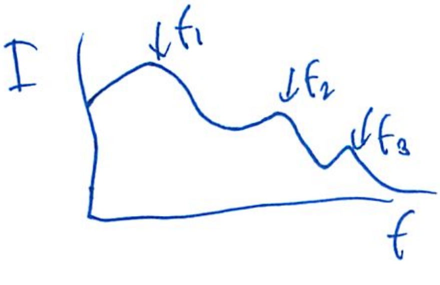
but stuff is all over the place



Speech

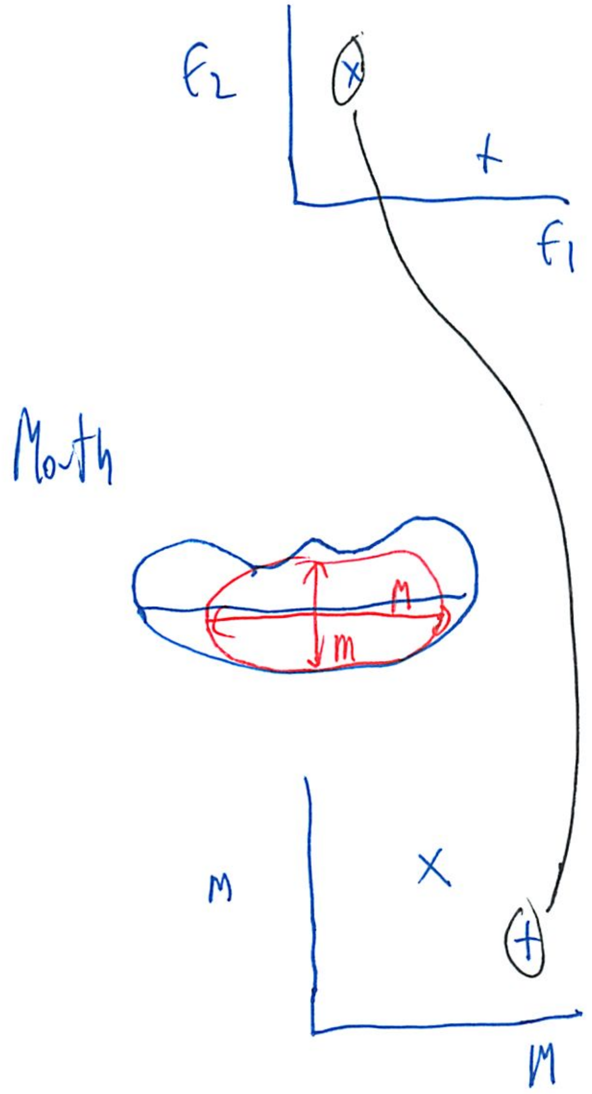


Smooth out



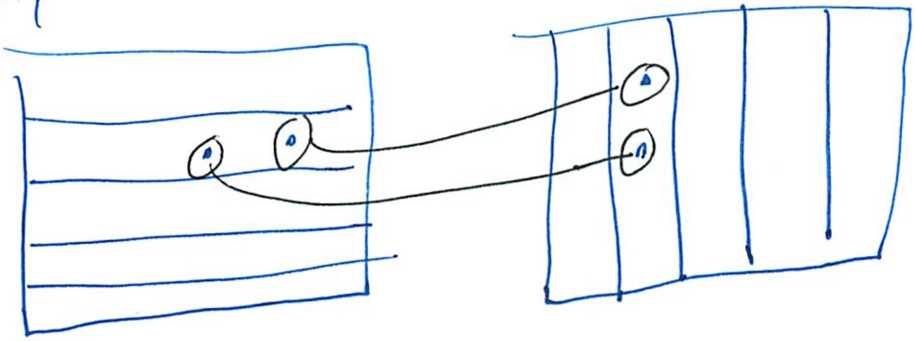
3

Point at where peaks occur



But when we were little we didn't learn in graphs
One side can assist the otherside

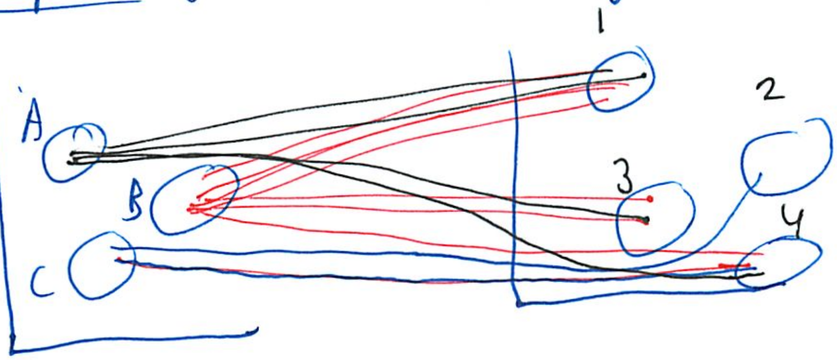
Basically



: Trying to cluster group

Using ~~[illegible]~~ distance metric to try and see how close other points

Technical part (could be on final)



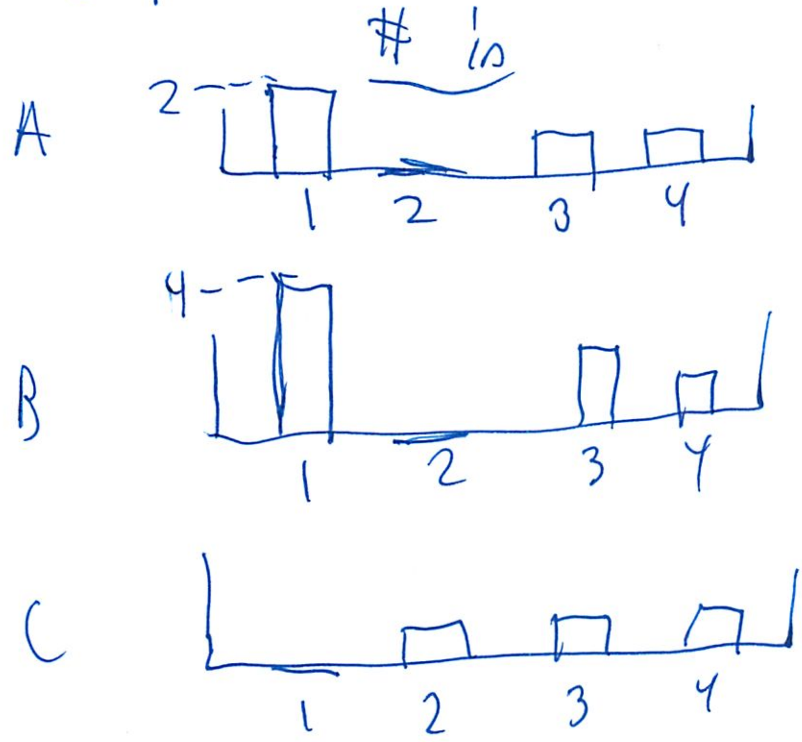
left

right

Some partitioning algorithm on right and left

Note the co-occurences

Components of vector



What would ya merge?

A and B - not C

Then alternate sides merging regions

Can also cross gestures, etc

Try to find out Zebra Finch lean maiting call?

Not known if it actually works

Review

We've seen a lot of methods from different people

classic Integration program to unpublished work

Can think of AI as

- Eng displire - building stuff
- Sci displire - understanding how

Biz - about making new things possible

Why is AI different?

- lang for procedures
- new ways to make models
- enforced detal
- oppertunity to expriment
- upper bounds

7

Diff than other subjects because its computational

Can quantify knowledge needed to solve problem

How do you do it?

- Characterize behavior
- Formulate computational problems
- Propose " solutions
- Implement exploratory systems
- Crystallze at the principles

Final

1. Rules + Search
2. Games + Constraints
3. NN, NN, ID trees
4. SVM, Boosting
5. Prob Inferences

Part 3s on all sections (except perhaps 1)
└might be from a diff lecture

8

Open book, calculator, etc as always

Bring a clock

No computers

What's Next

Esp few AI classes next semester

Do you have any UROPs?

No unless you are persistent

Berwick's class to evolution

Winston's possible Class

- reading primary sources
- Communications
- Underground Guide says lots of quizzes
  - hacking the underground guide

How to Speak talk IAP

Feb 3 11 AM

9

Still some issues

Lots of schools you can go
- go for the prof you like
- apprenticeship
- Grad schools only care about them
- How will you contribute to their program
- People interested often in stuff not good at

Big Qu
- Is AI useful?
  - yes
- What are the powerful ideas?
- Can they be truly smart?
- Are we close?

Chinese Room Argument some guy translating Eng => Chinese

Computers like this?

Is system not smart?

Winstons; But humans are like this too!

Homunculus Fallacy

~~All~~ (see slides)

Powerful Ideas

- Good representations make you smater

(missed rest)

Really Powerful Ideas

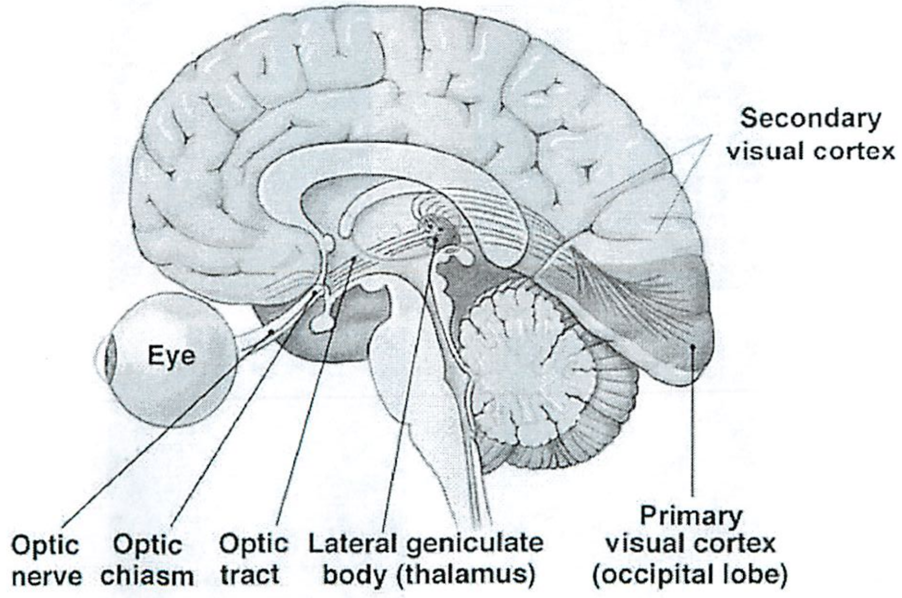
- You can change the world
- Only you can do it
- You can't do it alone
- Your obliged to do it
  - 7-8 people tried to get in instead

6.034
Farewell Address

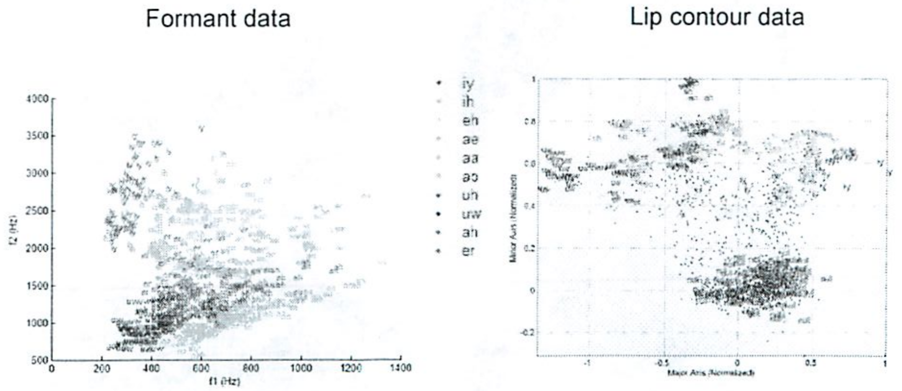
2011

The Exotic Engineering Hypothesis

There is a kind of engineering in our heads about which we are nearly clueless.



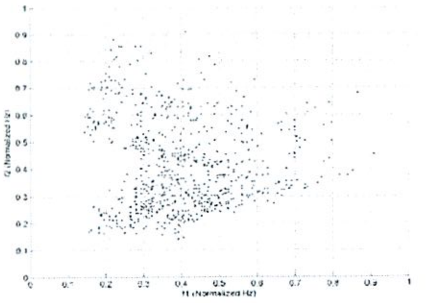
Secondary visual cortex
Eye
Optic nerve
Optic chiasm
Optic tract
Lateral geniculate body (thalamus)
Primary visual cortex (occipital lobe)


Formant data

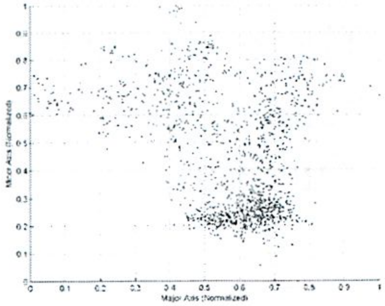
Lip contour data

12/14

Formant data

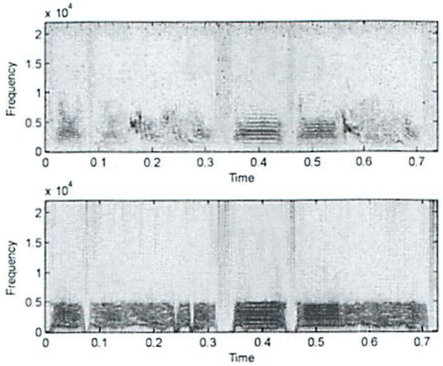


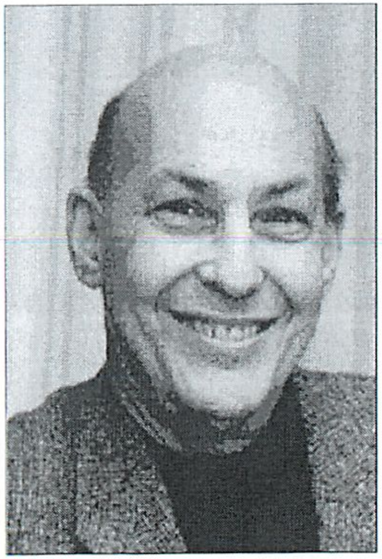
Lip contour data

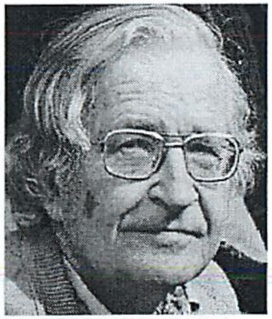


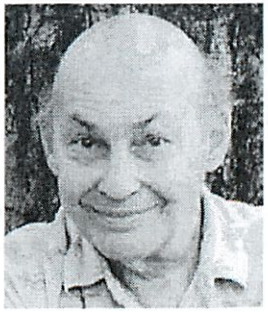
Samba

"Samba's son"









$$\int \frac{x^4}{(1-x^2)^{5/2}}dx = \frac{1}{3}\tan^3(\arcsin x) - \tan(\arcsin x) + \arcsin x$$

Duncan is a person. Lady Macbeth is a person. Macduff is a person. Macbeth is a person. A thane is a noble. Macbeth is a thane. Macduff is a thane. Lady Macbeth is greedy. Macbeth defeats a rebel. Appear is a success. Macbeth has a success. Witches talk with Macbeth. Witches have visions. Duncan rewards Macbeth because Duncan becomes happy. Macbeth wants to become king because Lady Macbeth persuades Macbeth to want to become king. Macbeth murders Duncan.

Duncan becomes dead because if a person murders another person, the other person becomes dead.

Engineering Perspective

Artificial Intelligence is about building stuff with

Representations

Methods

Architectures

Scientific Perspective

Artificial Intelligence is about understanding stuff with

Representations

Methods

Architectures

The Business Perspective

	Saves Money	Creates New Opportunity
Information Gatherers		✓
Blunder Stoppers		✓
Novice Workers		
Expert Workers	✗	

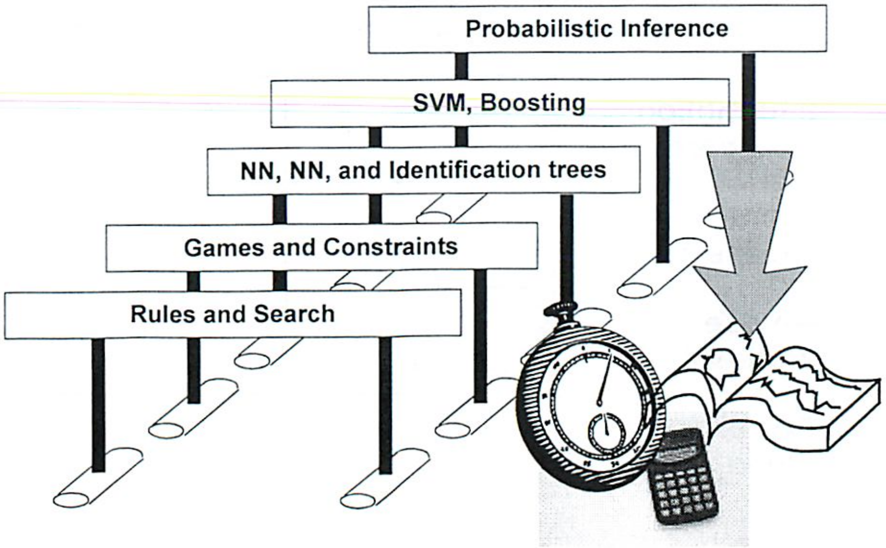
What Does AI Offer That Is Different

- A language for procedures
- New ways to make models
- Enforced detail
- Opportunities to experiment
- Upper bounds

How do you do it?

- Characterize behavior
- Formulate computational problems
- Propose computational solutions
- Implement exploratory systems
- Crystallize out the principles

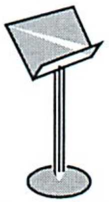
What Might Be on the Final



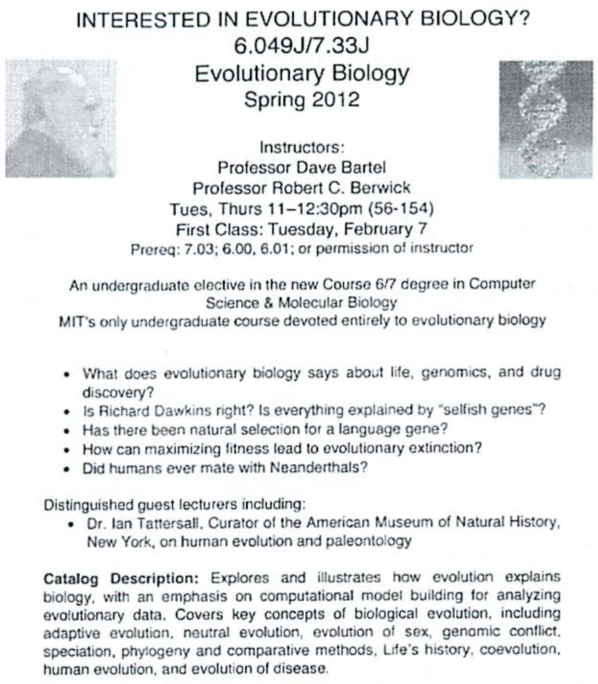
Winston's Picks

6.034

6.868	Minsky	Society of Mind ?	☹
6.891	Berwick	Evolution	
6.UAT	Davis	Communication	☹
6.945	Sussman	Large Scale Symbolic Systems	☹
9.71 (F)	Kanwisher	Functional MRI Investigations	☹
...	Richards	☹	
...	Tenenbaum	...	
...	Sinha	☹	
...	Battlecode	...	
...	UROP	...	
6.xxx	Winston	Human Intelligence Enterprise	



How to Speak
Friday, February 3, 11am



INTERESTED IN EVOLUTIONARY BIOLOGY?
6.049J/7.33J
Evolutionary Biology
Spring 2012

Instructors:
Professor Dave Bartel
Professor Robert C. Berwick
Tues, Thurs 11–12:30pm (56-154)
First Class: Tuesday, February 7
Prereq: 7.03; 6.00, 6.01; or permission of instructor

An undergraduate elective in the new Course 6/7 degree in Computer Science & Molecular Biology
MIT's only undergraduate course devoted entirely to evolutionary biology

- What does evolutionary biology says about life, genomics, and drug discovery?
- Is Richard Dawkins right? Is everything explained by "selfish genes"?
- Has there been natural selection for a language gene?
- How can maximizing fitness lead to evolutionary extinction?
- Did humans ever mate with Neanderthals?

Distinguished guest lecturers including:
- Dr. Ian Tattersall, Curator of the American Museum of Natural History, New York, on human evolution and paleontology

Catalog Description: Explores and illustrates how evolution explains biology, with an emphasis on computational model building for analyzing evolutionary data. Covers key concepts of biological evolution, including adaptive evolution, neutral evolution, evolution of sex, genomic conflict, speciation, phylogeny and comparative methods, Life's history, coevolution, human evolution, and evolution of disease.

Winston's Picks

6.034

How to Speak
Friday, February 3, 11am

6.868	Minsky	Society of Mind ? ☹
6.891	Berwick	Evolution
6.UAT	Davis	Communication ☹
6.945	Sussman	Large Scale Symbolic Systems ☹
9.71 (F)	Kanwisher	Functional MRI Investigations ☹
...	Richards	☹
...	Tenenbaum	...
...	Sinha	☹
...	Battlecode	...
...	UROP	...
6.xxx	Winston	Human Intelligence Enterprise

6.XXX Benefits

- Understand the great ideas of the great thinkers and how they got them
- Learn how to extract and evaluate ideas from original, sometimes opaque sources
- Learn how to package your own ideas and expose their greatness

6.XXX Packaging Topics

Abstracts
Proposals
Slide presentations
Promotion letters
Letters of complaint
Trip reports
Elevator talks
Openings

Business plans
Press releases
Job interviews
Study briefs
Terms of reference
Panel discussions
How to threaten people

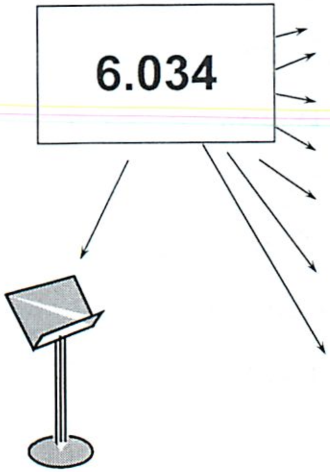
From the Underground Guide

Exams were described as “incredibly difficult,” “brutal,” and “frustrating.” They were graded harshly and “covered topics not taught in the class.”

From the Underground Guide

Officially, Winston has never confirmed or denied that there are quizzes for this class. His students seem to take after him --- comments were evenly split between complaints of brutal weekly 9:30AM quizzes and a "7-hour final", and denial of any and all testing. We at the UG aren't quite sure what to make of this.

Winston’s Picks

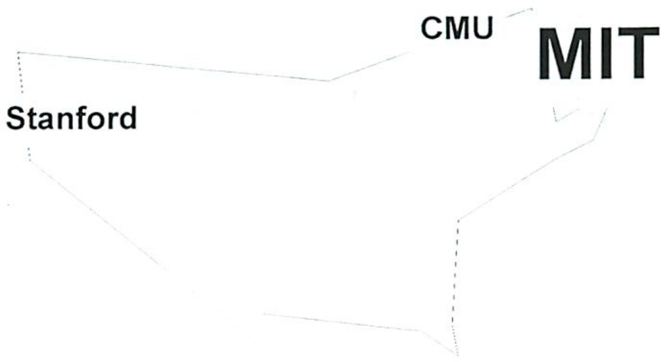


How to Speak
Friday, February 3, 11am

The Issues

- What can we know about the physical world?
- How do we handle abstract worlds?
- What can we imagine and why?
- How do we discover order in our perceptions?
- How do experience and culture guide thinking?
- How do symbols ground out in perception?
- How do our faculties learn to communicate?
- Why are human computers so robust?

Where Can You Go Next

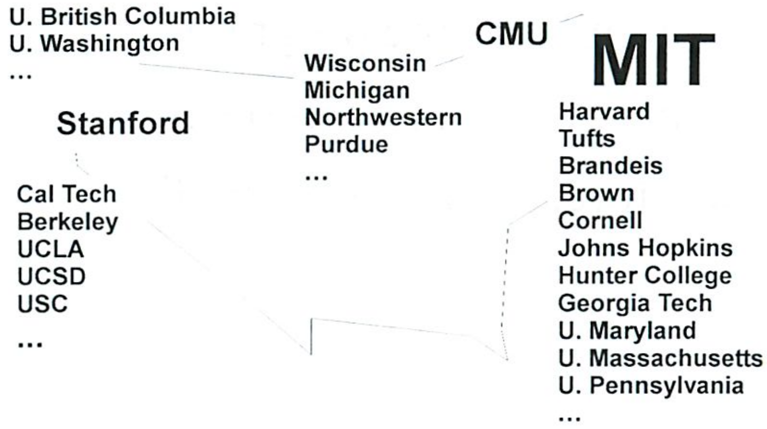


CMU

MIT

Stanford

Where Can You Go Next



U. British Columbia
U. Washington
...

Stanford

Cal Tech
Berkeley
UCLA
UCSD
USC
...

Wisconsin
Michigan
Northwestern
Purdue
...

CMU

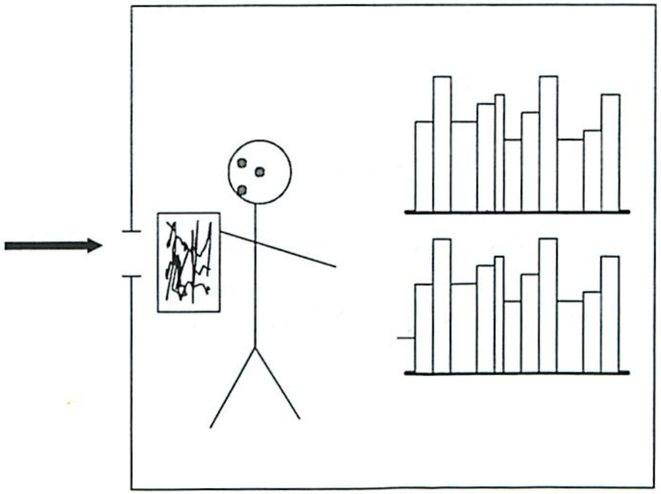
MIT

Harvard
Tufts
Brandeis
Brown
Cornell
Johns Hopkins
Hunter College
Georgia Tech
U. Maryland
U. Massachusetts
U. Pennsylvania
...

The Big Questions

- Is AI useful?
- What are the powerful ideas?
- Can they be truly smart?
- Are we close?

The Chinese-Room Argument



The Homunculus Fallacy

- It cannot be in the program
- It cannot be in the computer
- Therefore, it cannot be at all

The Biggest Issue

- Are people too smart?
- Are people smart enough?

The Powerful Ideas

- Good representations make you smarter
- Sleep makes you smarter
- You cannot learn unless you almost know
- You think with mouths, eyes, and hands
- The Strong Story Hypothesis

The Staff

Avril Kenney
Adam Mustafa
Caryn Krakauer
Erek Speed
Gary Planthaber
Mark Seifter
Peter Brin
Tanya Kortz

Bob Berwick
Randy Davis

David Broderick

The Rolling Stones
The Black Eyed Peas
…

A Really Powerful Idea

- You can change the world
- Only you can do it
- You can’t do it alone
- You are obliged to do it

Lab

Grades for Michael E Plasmeier:

Lab Average: 5.0

Labs Started/Completed: 6

lab5

Started: 2011-11-19 23:58:51
Ended: 2011-11-20 00:04:55
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 6

- lab5_theplaz_MIT_EDU_2011Nov19-221547.tar.bz2
- lab5_theplaz_MIT_EDU_2011Nov19-224123.tar.bz2
- lab5_theplaz_MIT_EDU_2011Nov19-233023.tar.bz2
- lab5_theplaz_MIT_EDU_2011Nov19-234937.tar.bz2
- lab5_theplaz_MIT_EDU_2011Nov19-235421.tar.bz2
- lab5_theplaz_MIT_EDU_2011Nov20-000217.tar.bz2

lab4

Started: 2011-10-31 19:44:41
Ended: 2011-10-31 19:50:34
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 8

- lab4_theplaz_MIT_EDU_2011Oct30-214528.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-173341.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-180921.tar.bz2

- lab4_theplaz_MIT_EDU_2011Oct31-182048.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-192142.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-192927.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-192954.tar.bz2
- lab4_theplaz_MIT_EDU_2011Oct31-194532.tar.bz2

lab3

Started: 2011-10-14 22:56:04
Ended: 2011-10-14 23:04:41
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 13

- lab3_theplaz_MIT_EDU_2011Oct03-211514.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct03-215856.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct03-223949.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct03-230221.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct10-165956.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct10-235302.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct11-012455.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-211926.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-224620.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-225526.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-225606.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-230736.tar.bz2
- lab3_theplaz_MIT_EDU_2011Oct14-231611.tar.bz2

lab2

Started: 2011-09-25 02:02:02
Ended: 2011-09-25 02:02:17
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 6

- lab2_theplaz_MIT_EDU_2011Sep25-003250.tar.bz2

- lab2_theplaz_MIT_EDU_2011Sep25-010213.tar.bz2
- lab2_theplaz_MIT_EDU_2011Sep25-011148.tar.bz2
- lab2_theplaz_MIT_EDU_2011Sep25-012911.tar.bz2
- lab2_theplaz_MIT_EDU_2011Sep25-015649.tar.bz2
- lab2_theplaz_MIT_EDU_2011Sep25-020205.tar.bz2

lab0

Started: 2011-09-16 22:01:22
Ended: 2011-09-16 22:01:29
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 15

- lab0_theplaz_MIT_EDU_2011Sep16-172228.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-180550.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-180615.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-180641.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-193148.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-193543.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-193627.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-193728.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-194032.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-194827.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-195054.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-211514.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-214410.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-215138.tar.bz2
- lab0_theplaz_MIT_EDU_2011Sep16-220124.tar.bz2

lab1

Started: 2011-09-20 01:46:20
Ended: 2011-09-20 01:46:30
Lab Grade:
5.0

Test was Run to Completion:
YES

Submissions: 6

- lab1_theplaz_MIT_EDU_2011Sep19-220902.tar.bz2
- lab1_theplaz_MIT_EDU_2011Sep19-224442.tar.bz2
- lab1_theplaz_MIT_EDU_2011Sep20-001854.tar.bz2
- lab1_theplaz_MIT_EDU_2011Sep20-013703.tar.bz2
- lab1_theplaz_MIT_EDU_2011Sep20-014622.tar.bz2
- lab1_theplaz_MIT_EDU_2011Sep20-014905.tar.bz2

Reference material and playlist

From 6.034 Fall 2011

Final

Most of the readings come from Patrick Winston's AI textbook (third edition), which exists as a physical book (http://www.amazon.com/Artificial-Intelligence-3rd-Winston/dp/0201533774/) , but is also available on the internet (http://courses.csail.mit.edu/6.034f/ai3/) (and there's a table of contents here (http://people.csail.mit.edu/phw/Books/AITABLE.HTML)).

Topics and Playlist 2011

-->

September	Day	Topic	Quiz #	Playlist
7	Wed	What it's all about	1	This could be the last time, Stones
12	Mon	Goal trees and symbolic integration (http://courses.csail.mit.edu/6.034f/ai3/saint.pdf)	1	You can get it if you really want it, Jimmy Cliff
14	Wed	Goals and rule-based systems (pp.53-60) (http://courses.csail.mit.edu/6.034f/ai3/ch3.pdf)	1	Engineer's Song, Chorallaries
19	Mon	Basic search (http://courses.csail.mit.edu/6.034f/ai3/ch4.pdf)	1	Searchin', Stones
23	Fri	Optimal search (http://courses.csail.mit.edu/6.034f/ai3/ch5.pdf)	1	Route 66, Stones
26	Mon	Games (http://courses.csail.mit.edu/6.034f/ai3/ch6.pdf)	2	It's Only Rock and Roll, Stones
28	Wed	Quiz 1	-	-
October	**Day**	**Topic**	**Quiz #**	**Playlist**
3	Wed	Constraints in drawings (http://courses.csail.mit.edu/6.034f/ai3/ch12.pdf)	2	I Can't Get No Satisfaction, Stones
5	Wed	Constraints in maps and resource allocation	2	Paint it Black, Stones
12	Wed	Constraints in object recognition (http://courses.csail.mit.edu/6.034f/ai3/ch26.pdf)	2	The First Time I Saw your Face, Presley
14	Fri	Nearest neighbor learning (http://courses.csail.mit.edu/6.034f/ai3/ch19.pdf) /Sleep (http://courses.csail.mit.edu/6.034f/sleep.pdf)	3	ABC song, Ray Charles et al.

17	Mon	Identification tree learning (http://courses.csail.mit.edu/6.034f/ai3/ch21.pdf)	3	Romanian national anthem, Desteapta-te române!
19	Wed	Neural net learning (http://courses.csail.mit.edu/6.034f/ai3/netmath.pdf)	3	19th Nervous Breakdown, Stones
24	Mon	Genetic algorithms (http://courses.csail.mit.edu/6.034f/ai3/ch25.pdf)	3	Let's spend the night together, Stones
26	Wed	Quiz 2	-	-
31	Mon	Learning in sparse spaces (http://courses.csail.mit.edu/6.803/pdf/yip.pdf)	3	You talk too much, Peas
November	**Day**	**Topic**	**Quiz #**	**Playlist**
2	Wed	Support-vector machines (http://courses.csail.mit.edu/6.034f/ai3/SVM.pdf) , SVM (and Boosting) Notes (http://ai6034.mit.edu/fall11/images/SVM_and_Boosting.pdf)	4	Get a little help from my friends, Beatles
7	Mon	Learning from near misses (http://courses.csail.mit.edu/6.034f/ai3/ch16.pdf)	3	Imma Be Rocking that Body, Peas
9	Wed	Boosting (Winston and Ortiz notes) (http://courses.csail.mit.edu/6.034f/ai3/boosting.pdf) , Boosting (Shapiri paper) (http://courses.csail.mit.edu/6.034f/ai3/msri.pdf)	4	Workin' together, Ike and Tina Turner
14	Mon	Frames and representation (http://courses.csail.mit.edu/6.034f/ai3/ch9.pdf)	4	Selections from the Black Watch, aka The Ladies from Hell
16	Wed	Quiz 3	-	-
21	Mon	Slides (http://courses.csail.mit.edu/6.034f/ai3/Emotionmachine.pdf) GPS, SOAR (http://courses.csail.mit.edu/6.034f/ai3/SOAR.pdf) , Subsumption (http://courses.csail.mit.edu/6.034f/ai3/Subsumption.pdf) , Society of Mind (http://web.media.mit.edu/~minsky/eb5.html)	4	Thus spake Zarathustra, Strauss
23	Wed	The AI Business	-	Money, money, ABBA
28	Mon	Probabilistic inference I (http://courses.csail.mit.edu/6.034f/ai3/bayes.pdf)	5	Oh No, Not You Again, Stones
30	Wed	Probabilistic inference II (http://courses.csail.mit.edu/6.034f/ai3/bayes.pdf)	5	Tumbling Dice, Stones

December	Day	Topic	Quiz #	Playlist	
5	Mon	Watching the brain at work, less than you want to know (http://courses.csail.mit.edu/6.034f/ai3/Kanwisher2010.pdf) Watching the brain at work, more than you want to know (http://web.mit.edu/bcs/nklab/publications.shtml)	5	Happy, Stones	
7	Wed	Quiz 4	-	-	
12	Mon	Slides (http://courses.csail.mit.edu/6.034f/ai3/Rightway.pdf) Hypotheses: more than you want to know (http://courses.csail.mit.edu/6.034f/ai3/Submitted.pdf)	5	Ode to Joy, Ninth Symphony, Beethoven	
14	Wed	Slides (http://courses.csail.mit.edu/6.034f/ai3/Farewell2011.pdf) Cross-modal clustering: less and more than you want to know (http://courses.csail.mit.edu/6.034f/ai3/short-coen.pdf)	Cross modal clustering, remarks, discussion of the final	5	Don't stop, Stones

Retrieved from "http://ai6034.mit.edu/fall11/index.php?title=Reference_material_and_playlist"

- This page was last modified on 14 December 2011, at 21:09.
- *Forsan et haec olim meminisse iuvabit.*