CS 221: Artificial Intelligence

Lecture 4: Probabilistic Inference

Sebastian Thrun and Peter Norvig

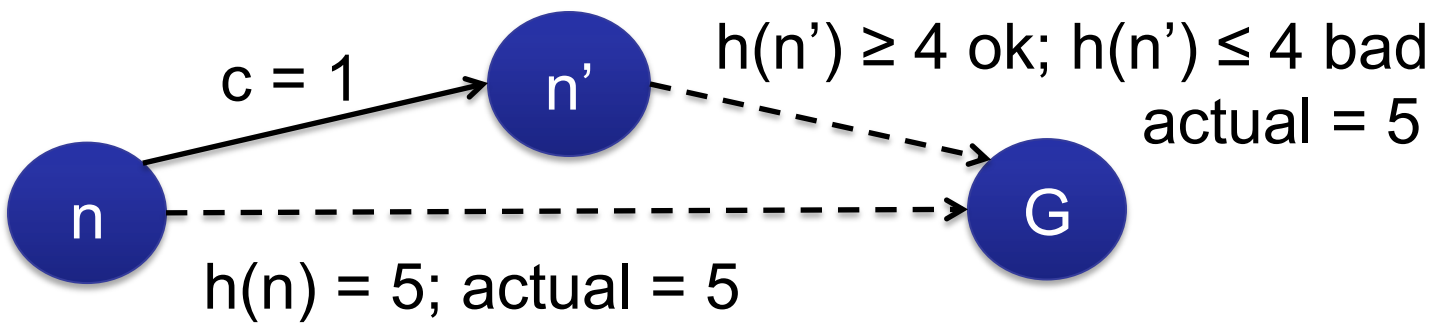
Slide credit: Dan Klein

Announcements

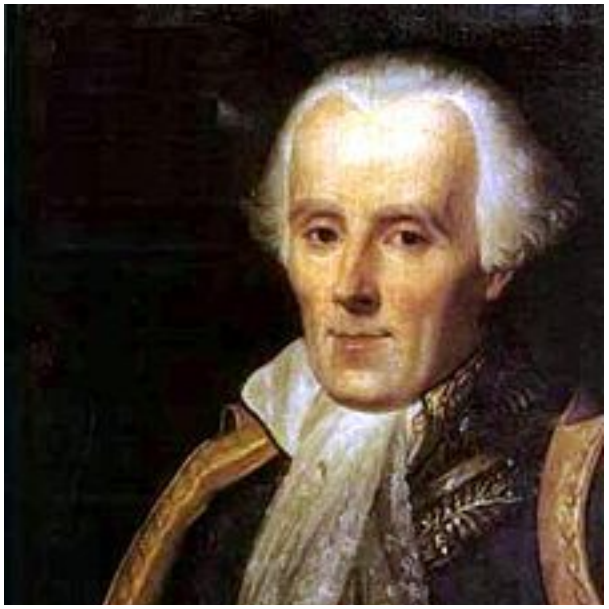
- Homework Assignment 2 Due Sunday
  - stanford.ai-class.com
- Programming Assignment 2 Due Today
  - cs221.stanford.edu
- TA Review Session Next Tuesday

Consistency

- Admissible: $h(n) \leq actual(n)$
  - Makes A* optimal for tree search
- Consistent: $h(n) \leq c(n, a, n') + h(n')$
  - Makes A* optimal for graph search
  - Or you can check explored/frontier for dups.



Probability



"Probability theory is nothing But common sense reduced to calculation."
- *Pierre Laplace, 1819*

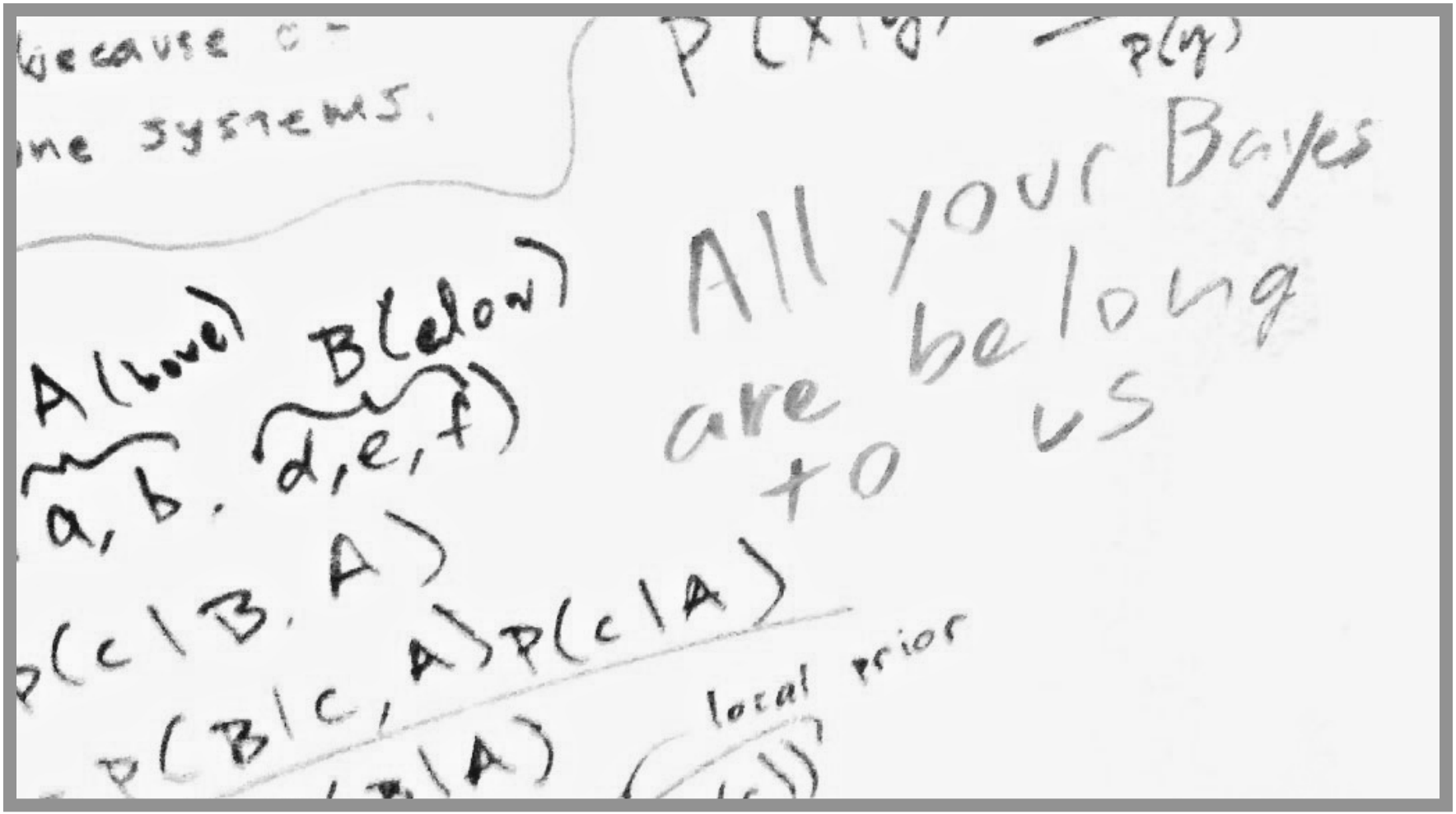
The true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind."
- *James Maxwell, 1850*

Probabilistic Inference

- Joel Spolsky:
  A very senior developer who moved to Google told me that Google works and thinks at a higher level of abstraction...

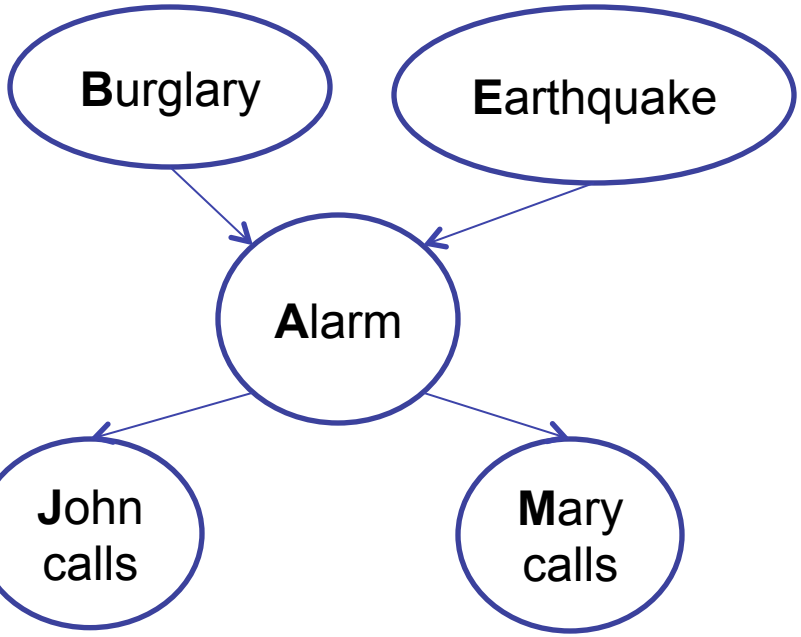
  "Google uses Bayesian filtering the way [previous employer] uses the if statement," he said.

Google Whiteboard



Example: Alarm Network

B	P(B)
+b	0.001
¬b	0.999



E	P(E)
+e	0.002
¬e	0.998

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	¬a	0.05
+b	¬e	+a	0.94
+b	¬e	¬a	0.06
¬b	+e	+a	0.29
¬b	+e	¬a	0.71
¬b	¬e	+a	0.001
¬b	¬e	¬a	0.999

A	J	P(J|A)
+a	+j	0.9
+a	¬j	0.1
¬a	+j	0.05
¬a	¬j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	¬m	0.3
¬a	+m	0.01
¬a	¬m	0.99

Probabilistic Inference

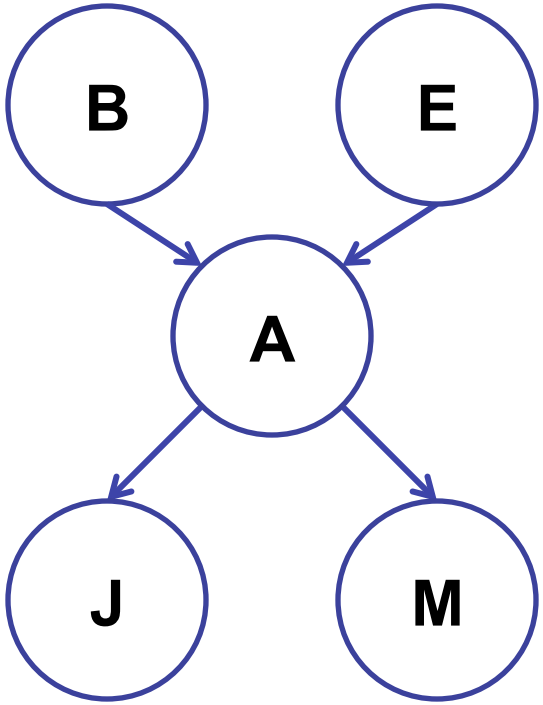
- Probabilistic Inference: calculating some quantity from a joint probability distribution
  - Posterior probability:

    $$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$

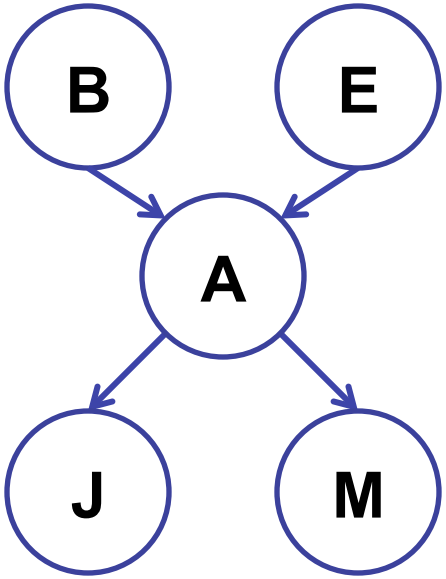
  - In general, partition variables into *Query (Q or X)*, *Evidence (E)*, and *Hidden (H or Y)* variables



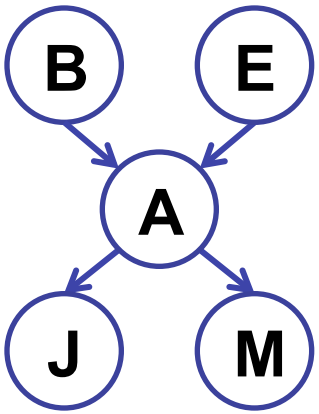
Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the unconditional probabilities you need
  - Enumerate *all* the atomic probabilities you need
  - Calculate sum of products
- Example:

$$P(+b|+j,+m) = \frac{P(+b,+j,+m)}{P(+j,+m)}$$



Inference by Enumeration

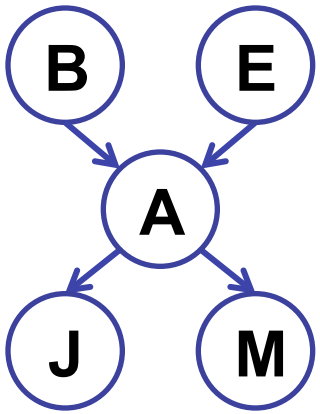


$$P(+b, +j, +m)$$
$$= \sum_e \sum_a P(+b, +j, +m, e, a)$$
$$= \sum_e \sum_a P(+b)\, P(e)\, P(a|+b,e)\, P(+j|a)\, P(+m|a)$$

$$\begin{aligned} = \quad & P(+b)P(+e)P(+a|+b,+e)P(+j|+a)P(+m|+a)+ \\ & P(+b)P(+e)P(-a|+b,+e)P(+j|-a)P(+m|-a)+ \\ & P(+b)P(-e)P(+a|+b,-e)P(+j|+a)P(+m|+a)+ \\ & P(+b)P(-e)P(-a|+b,-e)P(+j|-a)P(+m|-a) \end{aligned}$$

Inference by Enumeration

- An optimization: pull terms out of summations



$$P(+b, +j, +m)$$

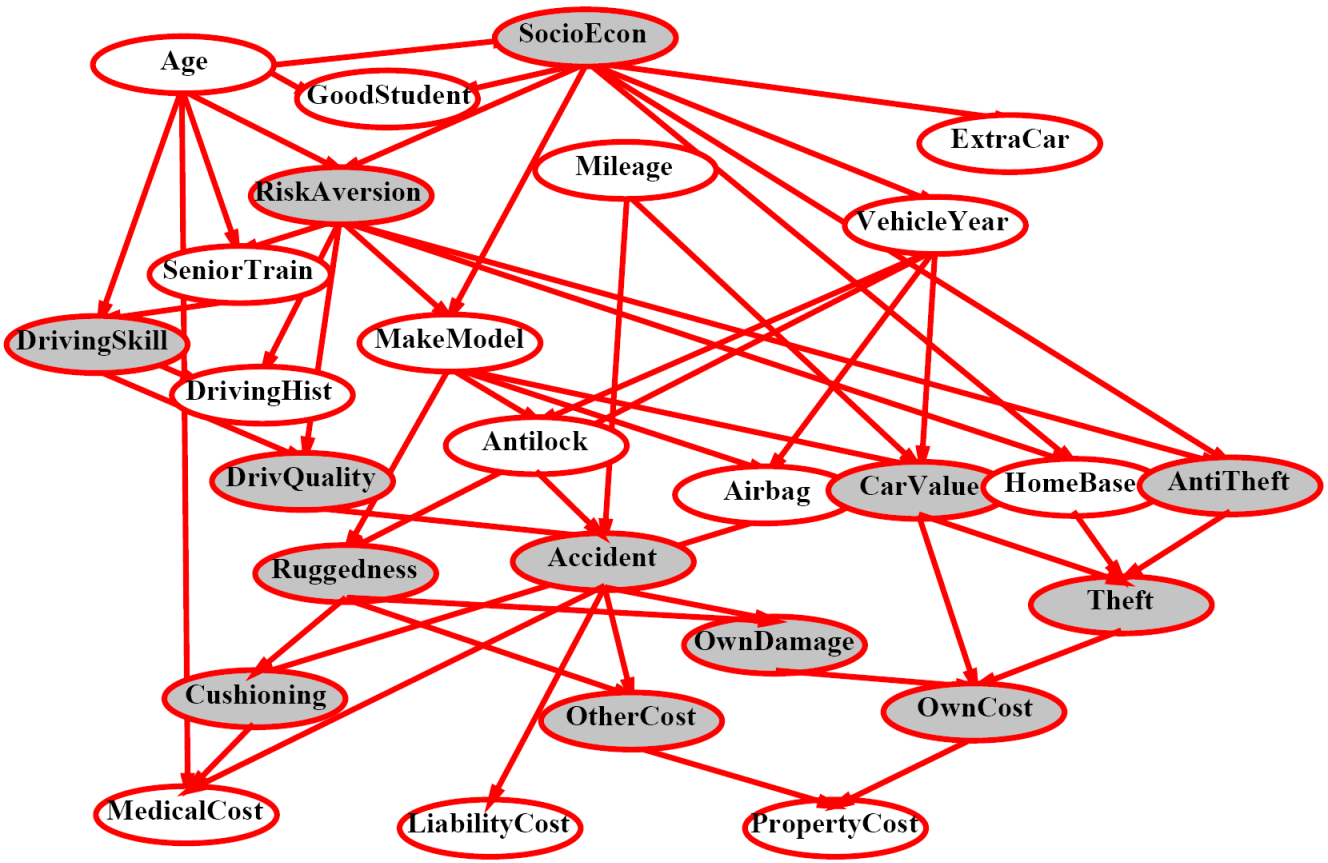
$$= \sum_e \sum_a P(+b, +j, +m, e, a)$$

$$= \sum_e \sum_a P(+b)\, P(e)\, P(a|+b,e)\, P(+j|a)\, P(+m|a)$$

$$= P(+b) \sum_e P(e) \sum_a P(a|+b,e)\, P(+j|a)\, P(+m|a) \qquad \textbf{or}$$

$$= P(+b) \sum_a P(+j|a)\, P(+m|a) \sum_e P(e)\, P(a|+b,e)$$

Inference by Enumeration



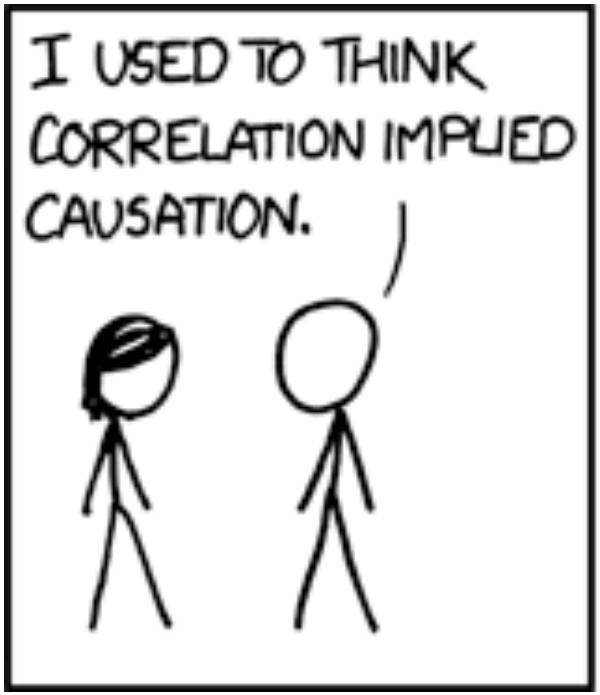
Problem?

Not just 4 rows; approximately 10^{16} rows!

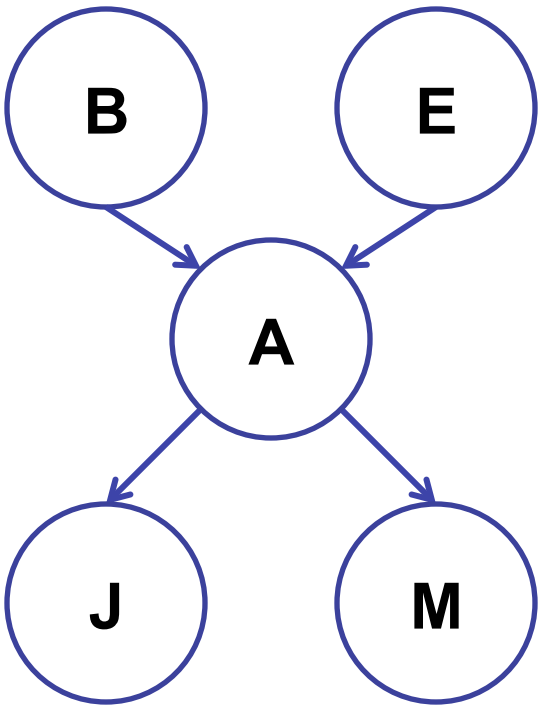
How can we make inference tractible?

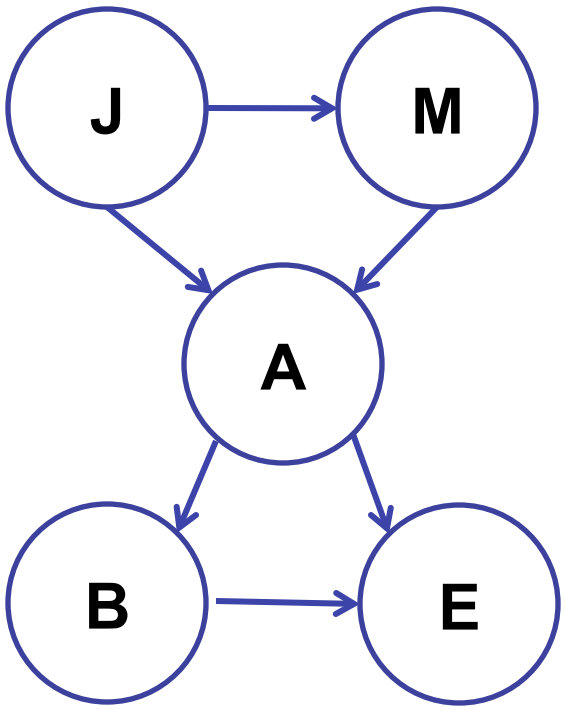
Causation and Correlation

xkcd

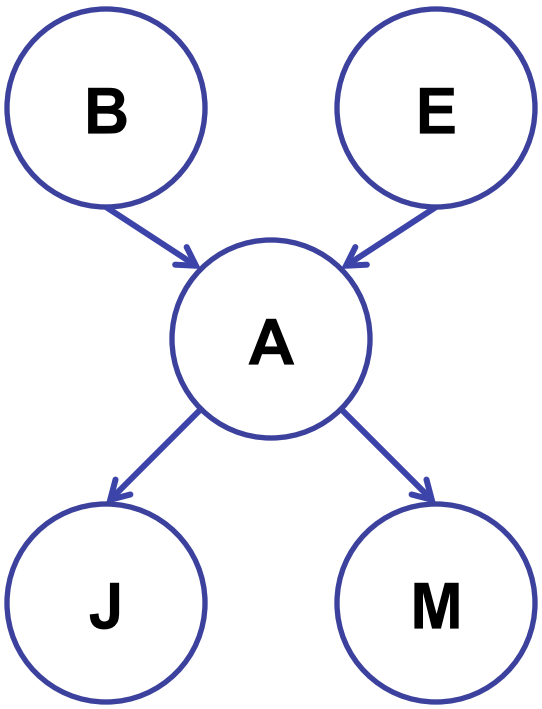


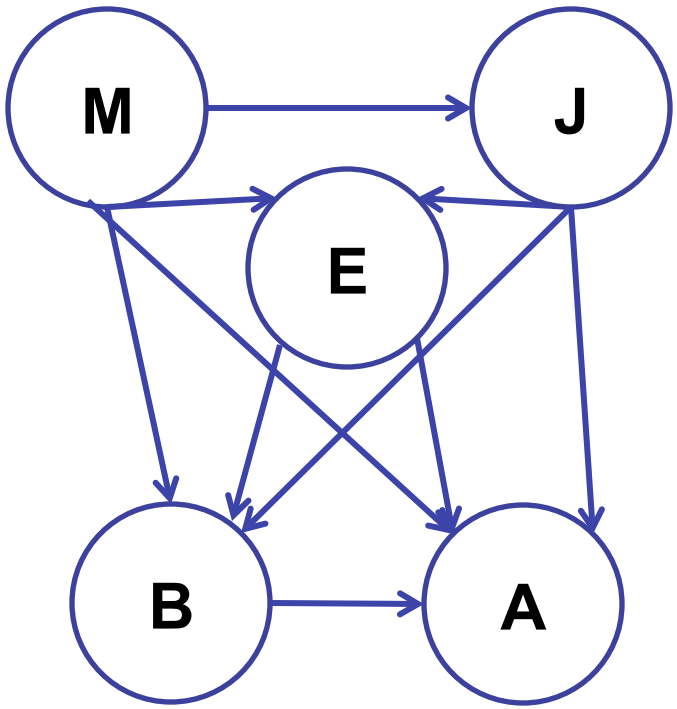
Causation and Correlation





Causation and Correlation





Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out (marginalize) the hidden variables
    $(\sum_e \sum_a P(+b)\, P(e)\, P(a|+b,e)\, P(+j|a)\, P(+m|a))$
  - You end up repeating a lot of work!

- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration
  - Requires an algebra for combining "factors" (multi-dimensional arrays)

Variable Elimination Factors

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1

$$P(T, W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Selected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for fixed x, all y
  - Sums to P(x)

$$P(cold, W)$$

T	W	P
cold	sun	0.2
cold	rain	0.3

Variable Elimination Factors

- Family of conditionals: P(X |Y)
  - Multiple conditional values
  - Entries P(x | y) for all x, y
  - Sums to |Y|
    (e.g. 2 for Boolean Y)

$$P(W|T)$$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

$$P(W|hot)$$

$$P(W|cold)$$

- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1

$$P(W|cold)$$

T	W	P
cold	sun	0.4
cold	rain	0.6

Variable Elimination Factors

- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$$P(rain|T)$$

T	W	P
hot	rain	0.2
cold	rain	0.6

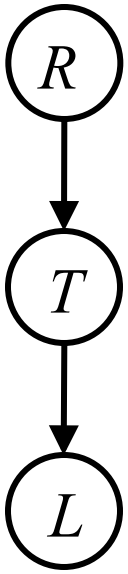
$P(rain|hot)$

$P(rain|cold)$

- In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$
  - It is a "factor," a multi-dimensional array
  - Its values are all $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
  - Any assigned X or Y is a dimension missing (selected) from the array

Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L \mid T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Variable Elimination Outline

- Track multi-dimensional arrays called factors
- Initial factors are local CPTs (one per node)

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know $L = +\ell$, the initial factors are

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

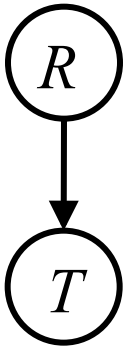
$P(+\ell|T)$

+t	+l	0.3
-t	+l	0.1

- VE: Alternately join factors and eliminate variables

Operation 1: Join Factors

- Combining factors:
  - Just like a database join
  - Get all factors that mention the joining variable
  - Build a new factor over the union of the variables involved
- Example: Join on R



$$P(R) \times P(T|R) \Rightarrow P(R,T)$$

+r	0.1
-r	0.9

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

- Computation for each entry: pointwise products

$$\forall r, t: \quad P(r,t) = P(r) \cdot P(t|r)$$

Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

$P(R,T)$

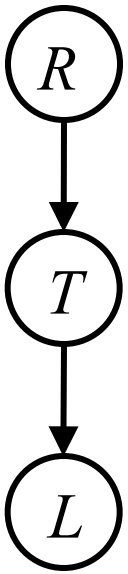
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum R →

$P(T)$

+t	0.17
-t	0.83

Example: Compute P(*L*)



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

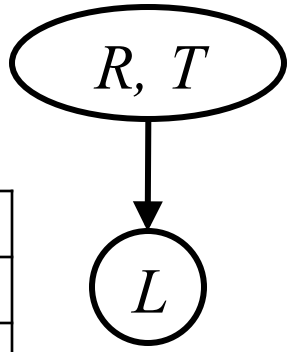
Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



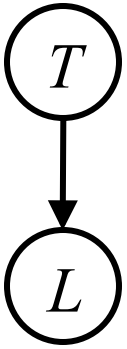
Sum out R

$P(T)$

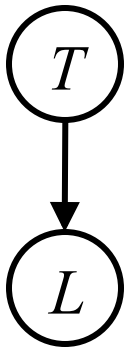
+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



Example: Compute P(*L*)



Join T

Sum out T

$P(T)$

+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

$P(L)$

+l	0.134
-l	0.886

Early marginalization is variable elimination

Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

  - Computing $P(L|+r)$, the initial factors become:

$P(+r)$

+r	0.1

$P(T|+r)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- We eliminate all vars other than query + evidence

Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for P(L | +r), we'd end up with:

$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize

$P(L| + r)$

+l	0.26
-l	0.74

- To get our answer, just normalize this!

- That's it!

General Variable Elimination

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H
- Join all remaining factors and normalize

Example

$P(B|j,m)$

$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$

Choose A

$P(A|B,E)$
$P(j|A)$
$P(m|A)$

$\times$ → $P(j,m,A|B,E)$ → Σ a → $P(j,m|B,E)$

$P(B)$	$P(E)$	$P(j,m\|B,E)$

Example

$P(B)$ $P(E)$ $P(j,m|B,E)$

Choose E

$P(E)$

$P(j,m|B,E)$

$\times$ $P(j,m,E|B)$ Σ e $P(j,m|B)$

$P(B)$ $P(j,m|B)$

Finish with B

$P(B)$

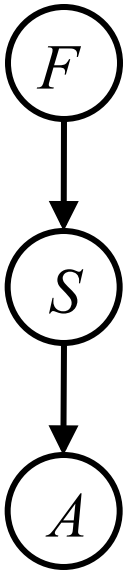
$P(j,m|B)$

$\times$ $P(j,m,B)$ Normalize $P(B|j,m)$

Pause for Questions

Approximate Inference

- Sampling / Simulating / Observing
- Sampling is a hot topic in machine learning, and it is really simple
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the exact answer (e.g. with variable elimination)



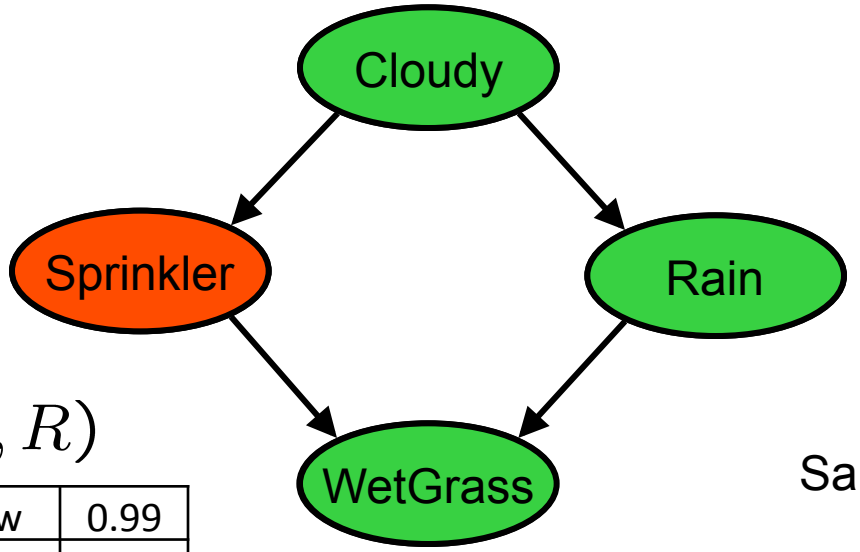
Prior Sampling

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, -s, +r, +w

-c, +s, -r, +w

…

Prior Sampling

- This process generates samples with probability:

$$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)) = P(x_1 \ldots x_n)$$

…i.e. the BN's joint probability

- Let the number of samples of an event be $N_{PS}(x_1 \ldots x_n)$

- Then $$\begin{aligned} \lim_{N\to\infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

- I.e., the sampling procedure is consistent

Example

- We'll get a bunch of samples from the BN:

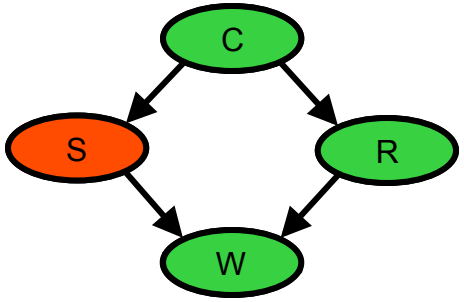
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

  +c, -s, +r, +w

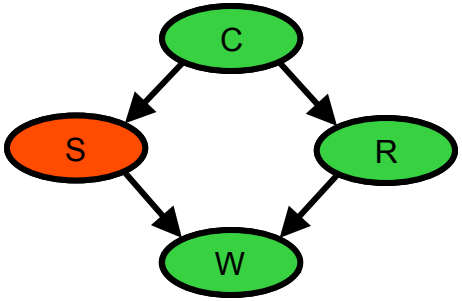
  -c, -s, -r, +w



- If we want to know P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - Fast: can use fewer samples if less time (what's the drawback?)

Rejection Sampling

- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go



- Let's say we want P(C| +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)

~~+c, -s, +r, +w~~
+c, +s, +r, +w
-c, +s, +r, -w
~~+c, -s, +r, +w~~
~~-c, -s, -r, +w~~

Sampling Example

- There are 2 cups.
  - First: 1 penny and 1 quarter
  - Second: 2 quarters

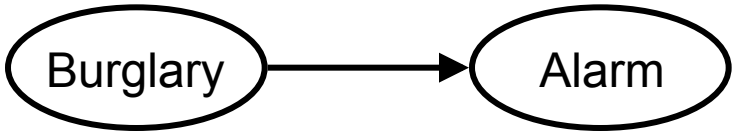
- Say I pick a cup uniformly at random, then pick a coin randomly from that cup. It's a quarter. What is the probability that the other coin in that cup is also a quarter?

25 25
~~1 25~~
25 25
25 1
25 25
~~1 25~~
25 25
25 1
25 25
25 25
25 1
~~1 25~~
~~1 25~~
25 25
~~1 25~~
~~1 25~~
25 25
25 1
~~1 25~~

747/
1000

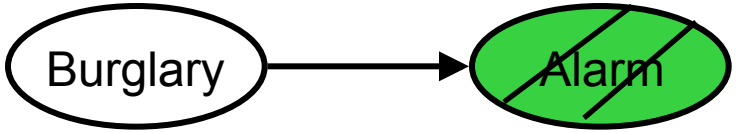
Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, you reject a lot of samples
  - You don't exploit your evidence as you sample
  - Consider P(B|+a)



-b, -a
-b, -a
-b, -a
-b, -a
+b, +a

- Idea: fix evidence variables and sample the rest



-b +a
-b, +a
-b, +a
-b, +a
+b, +a

- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents

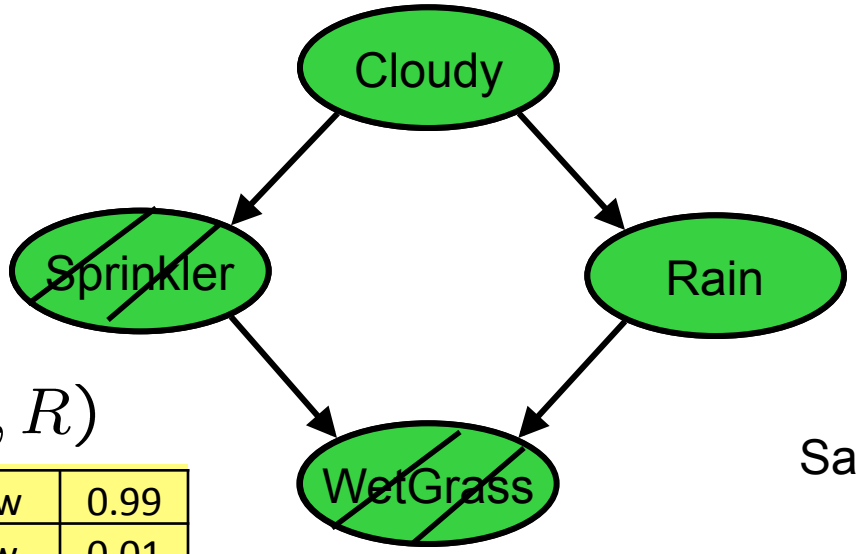
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, +s, +r, +w 0.099

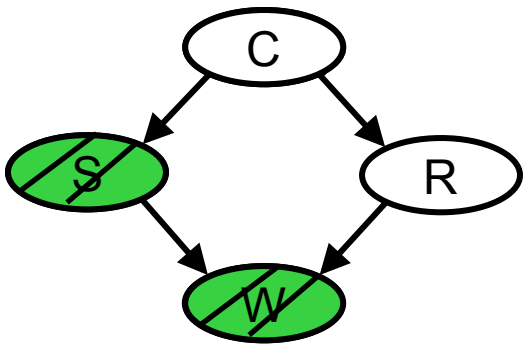
…

$$w = 1.0 \times 0.1 \times 0.99$$

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | \text{Parents}(Z_i))$$



- Now, samples have weights

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{m} P(e_i | \text{Parents}(E_i))$$

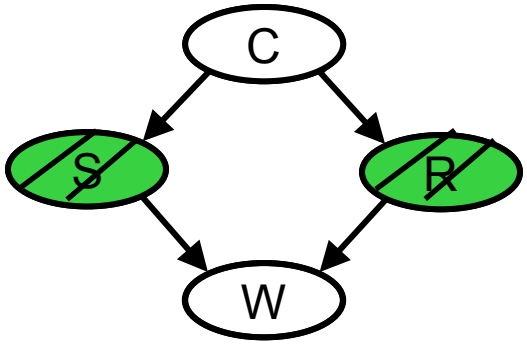
- Together, weighted sampling distribution is consistent

$$S_{\text{WS}}(z, e) \cdot w(z, e) = \prod_{i=1}^{l} P(z_i | \text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i | \text{Parents}(e_i))$$

$$= P(\mathbf{z}, \mathbf{e})$$

Likelihood Weighting

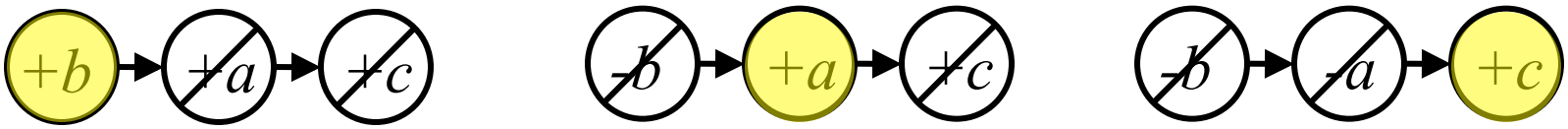
- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones (C isn't more likely to get a value matching the evidence)
- We would like to consider evidence when we sample *every* variable



Markov Chain Monte Carlo

- *Idea*: instead of sampling from scratch, create samples that are each like the last one.

- *Procedure*: resample one variable at a time, conditioned on all the rest, but keep evidence fixed. E.g., for P(b|c):



- *Properties*: Now samples are not independent (in fact they're nearly identical), but sample averages are still consistent estimators!

- *What's the point*: both upstream and downstream variables condition on evidence.

World’s most famous probability problem?

Monty Hall Problem

- Three doors, contestant chooses one.
- Game show host reveals one of two remaining, knowing it does not have prize
- Should contestant accept offer to switch doors?
- $P(+prize|\neg switch) = ?$
  $P(+prize|+switch) = ?$

Monty Hall on Monty Hall Problem

September 10, 1990

Mr. Lawrence A. Denenberg
Harvard University Center for
Research in Computing Technology
Aiken Computation Laboratory, Room 102
Harvard University
Cambridge, MA 02138

Dear Larry:

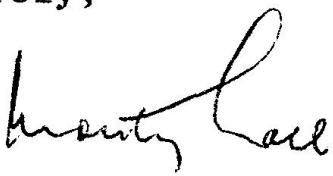
In sending you my okay for the use of "The Monty Hall Paradox," I should like to ask you a question. You mention that in part (a), the player should switch doors even without additional compensation -- indeed the player should be willing to pay Monty up to $21,845 for the privilege of switching.

Now, I am not well versed in algorithms; but as I see it, it wouldn't make any difference after the player has selected Door A, and having been shown Door C - why should he then attempt to switch to Door B? The major prize could only be in one of the three doors. He has made his selection of one of the doors. He has been shown one of the doors that contains a "booby"; ergo, the major prize will be either in the one he selected (Door A) or the one that

Monty Hall on Monty Hall Problem

question. You mention that in part (a), the player should switch doors even without additional compensation -- indeed the player should be willing to pay Monty up to $21,845 for the privilege of switching.

Now, I am not well versed in algorithms; but as I see it, it wouldn't make any difference after the player has selected Door A, and having been shown Door C - why should he then attempt to switch to Door B? The major prize could only be in one of the three doors. He has made his selection of one of the doors. He has been shown one of the doors that contains a "booby"; ergo, the major prize will be either in the one he selected (Door A) or the one that remains, Door B. Why would he be compelled to switch doors and even pay for the privilege? The chances of the major prize being behind Door A have not changed. He still has one of the two remaining doors. What makes Door B such an attraction? I would be pleased if you would write me, explaining this situation.

Best of luck with the book.

Sincerely,