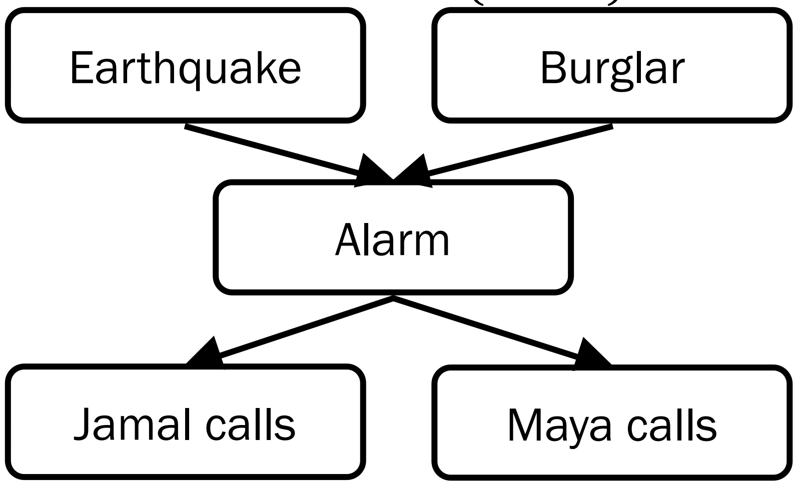
FUDAN SUMMER SCHOOL INTRODUCTION TO AI

Probabilistic Reasoning and Decision Making

Day 7 – variable elimination exercise, MLE learning

Inference in Bayes Nets: Variable Elimination

$P(E=1) = 0.001$ $\quad$ $P(B=1) = 0.005$



B	E	$P(A=1 \mid B,E)$
0	0	0.002
0	1	0.35
1	0	0.96
1	1	0.98

A	$P(J=1 \mid A)$
0	0.1
1	0.93

A	$P(M=1 \mid A)$
0	0.01
1	0.65

query variable(s)

evidence

Compute $P(B \mid J=1, M=1)$

$$= \frac{\sum_e \sum_a P(B, J=1, M=1, A=a, E=e)}{\sum_b P(B=b, J=1, M=1)}$$

E, A : non query, non-evidence

1. Create a factor for each CPT
2. Choose an elimination ordering for non-query variables (we'll use M, J, A, E)
3. Eliminate each variable in order by applying factor operations.
   1. Product
   2. Summing out
   3. Condition

Factor : { multiple, summing out, condition }

$P(B, J=1, M=1)$

Marginal

$\sum_e \sum_a P(B, J=1, M=1, A=a, E=e)$

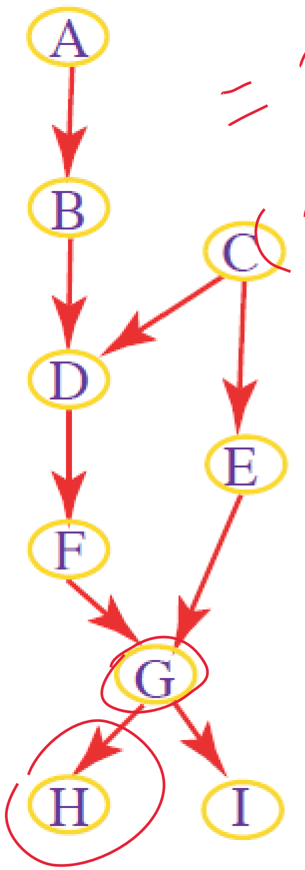
Calculate P(G | H = 1)

$P(G=0 \mid H=1)$, $P(G=1 \mid H=1)$ ← $P(G=0, H=1)$ & $P(G=1, H=1)$

2^7 terms, each term has 2^3 *

$= 2^{10} * 2$

$(2^7-1) + * 2$



$$P(G=1 \mid H=1) = \frac{P(G=1, H=1)}{P(G=0, H=1) + P(G=1, H=1)}$$

$$P(G=0, H=1) = \sum_a \sum_b \sum_c \sum_d \sum_e \sum_f \sum_i P(G=0, H=1, A=a, B=b, C=c, D=d, E=e, F=f, I=i)$$

$$= \sum_a \sum_b \cdots \sum_i P(A=a)\,P(B=b \mid A=a)\,P(C=c)\,P(D=d \mid B=b, C=c)\,P(E=e \mid C=c)\,P(F=f \mid D=d)\,P(G=0 \mid F=f, E=e)\,P(H=1 \mid G=0)\,P(I=i \mid G=0)$$

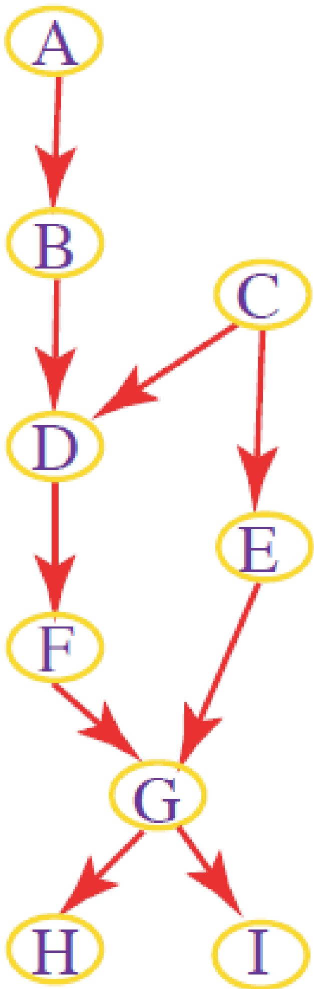
If I simply use joint probability to calculate the result, what is the number of ops that we need to do? Assume each variable is binary and we have all the CPTs

A. 2^14 multiplications and 2^8 additions
B. 2^15 multiplications and 2^7 additions
C. 2^11 multiplications and 2^9 additions — 2^8
D. 2^12 multiplications and 2^10 additions
E. None of the above

2^{11} *

2^8 additions

Calculate P(G | H = 1)



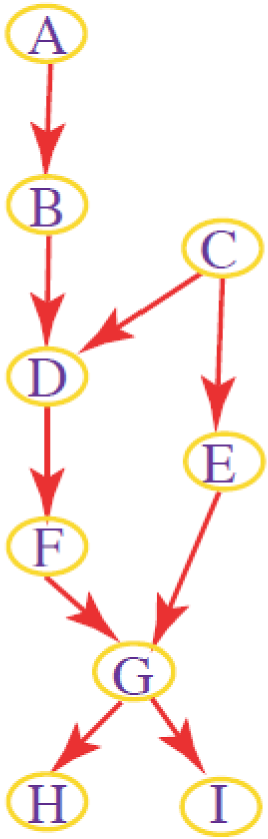
$$P(G, H=1)$$

$$= \sum_{a,b,c,d,e,f,i} P(A=a, B=b, C=c, D=d, E=e, G, H=1, I=i)$$

$$= \sum_{abcdefi} P(A=a)\, P(B=b \mid A=a)\, P(C=c)\, P(D=d \mid B=b, C=c)\, P(E=e \mid C=c)\, P(F=f \mid D=d)\, P(G \mid F=f, E=e)\, P(H=1 \mid G)\, P(I=i \mid G)$$

elimination order:

I, H, A, B, C, D, E, F



$$= \sum_{abcdefi} P(A=a)\,P(B=b|A=a)\,P(C=c)\,P(D=d|B=b,C=c)\,P(E=e|C=c)\,P(F=f|D=d)\,P(G|F=f,E=e)\,P(H=1|G)\,P(I=i|G)$$

elimination order: I, H, A, B, C, D, E, F

$$\sum_b \left(P(D=d|B=b,C=c) \sum_a \left(P(A=a)\,P(B=b|A=a) \right) \right) \cdot \underbrace{P(H=1|G)}_{\text{condition}} \cdot \underbrace{\sum_i P(I=i|G)}_{F(G)\,:\,2*}$$

F(G)

G	F(G)
0	~
1	~

2+

F(I, G)

I	G	F(I, G)
0	0	
0	1	
1	0	
1	1	

4+ F(B, C, D)

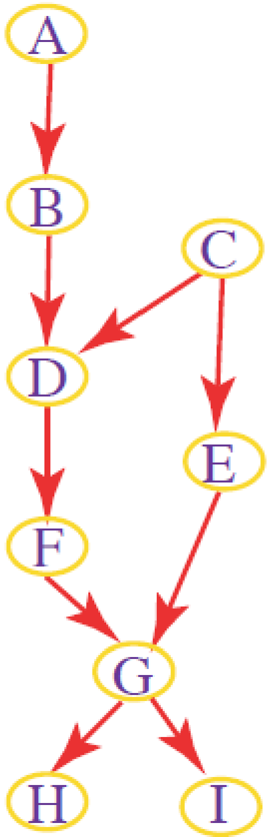
F(A, B) 4*

2+ F(B)

8* F(B, C, D)

f(H=1, G)

H	G	F(H=1, G)
1	0	~
1	1	~



$$= \sum_{abcdefi} \cancel{P(A=a)}\, \cancel{P(B=b|A=a)}\, \cancel{P(C=c)}\, \cancel{P(D=d|B=b,C=c)}$$

$$P(E=e|C=c)\, P(F=f|D=d)\, P(G|F=f,E=e)$$

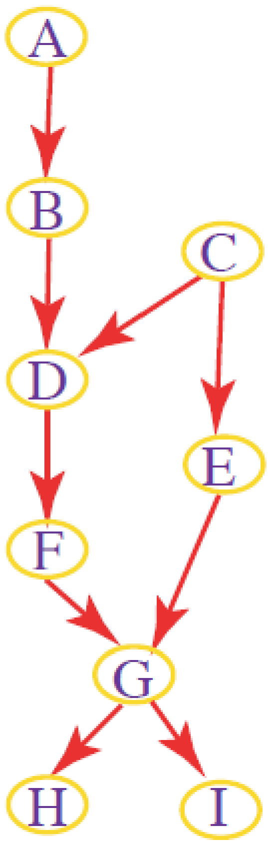
$$\cancel{P(H=1|G)}\, \cancel{P(I=i|G)}$$

elimination order:

I, H, A, B, C, D, E, F

$$\sum_{cdef} F(C,D) \cdot F(G)\, P(E=e|C=c)\, P(F=f|D=d)\, P(G|F=f,E=e)$$

Calculate P(G | H = 1)



$$\sum_{cdef} F(C,D) \cdot F(G)$$

$P(E=e \mid C=c)$

$P(F=f \mid D=d)$

$P(G \mid F=f, E=e)$

$$\sum_{cdef} F(C,D)\, F(G)\, F(C,E)\, F(D,F)\, F(E,F,G)$$

$$= F(G) \sum_f \sum_e F(E,F,G) \sum_d F(D,F) \sum_c F(C,D)\, F(C,E)$$

2* 2+ 4+ 8* 4+ 8* 4+ 8*

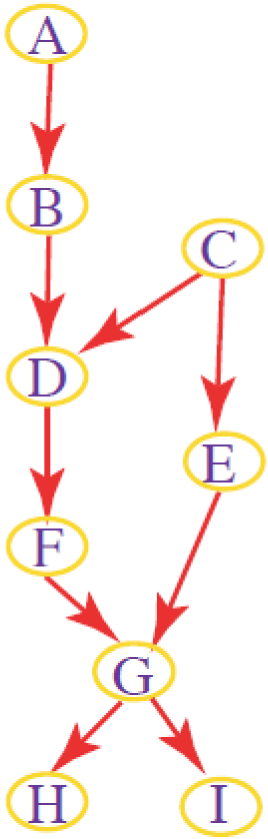
F(G) F(F,G) F(E,F) F(D,E) F(C,D,E)

F(E,F,G) F(D,E,F)

40 * ← 2048 *

20 + ← 256 +

Calculate P(G | H = 1)

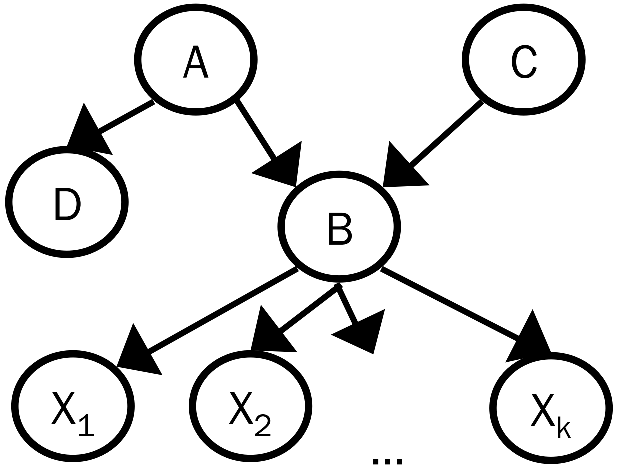


Does elimination order matter?

- In general, yes (but not in the trivial graphs we've been considering)
- Time and space of VE is dominated by the largest factor created
- Heuristic: Eliminate the variable that will lead to the smallest next factor being created
  - *In a polytree this leads to linear time inference (in size of largest CPT)*

DAG w/o undirected loops

no two undirected path between any two nodes

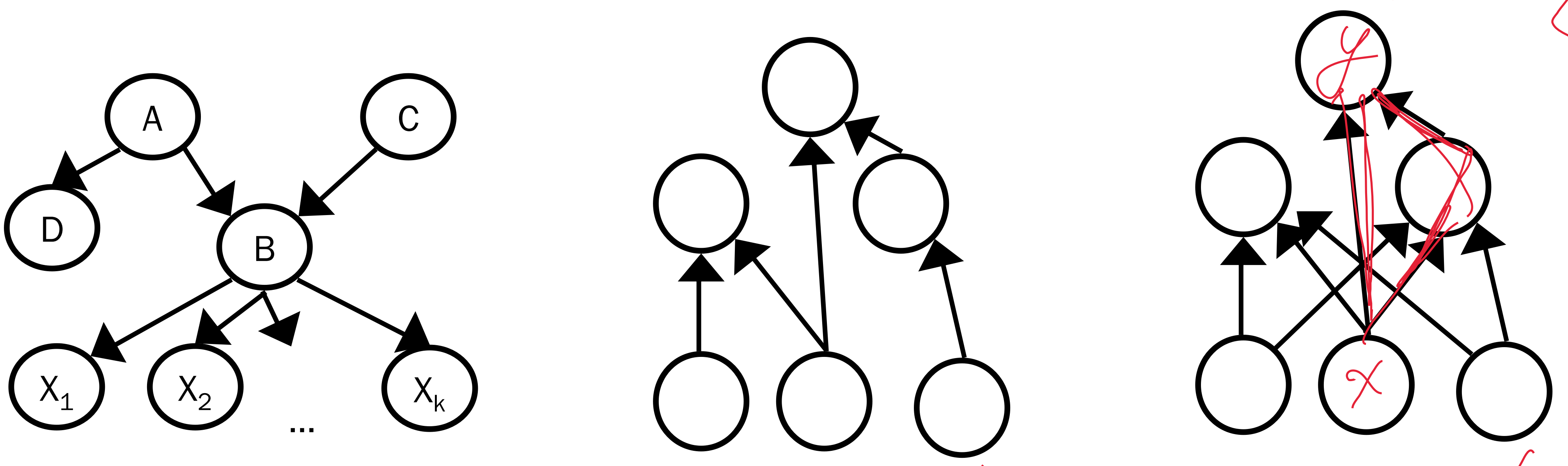


What is a polytree? A graph with no *undirected* loops

- Which are polytrees?

A. None of these B. I only C. I and II D. I, II and III

no a polytree



I. ✓ II. ✓ III. ✗

$Y_1 \leftarrow X \rightarrow Y_2$ condition 2

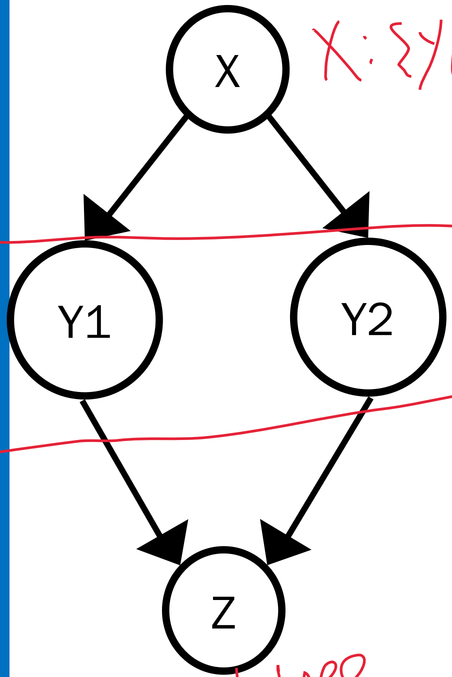
Making a polytree by collapsing nodes

$P(Y_1=0|X)$

$P(Y_1=0, Y_2=0|X) = P(Y_1=0|X) \cdot$

d-separation

$X: \{Y_1\},$ $E: \{X\}$, $Y = \{Y_2\}$



non-polytree

X	P(Y1=1|X)
0	0.1
1	0.5

.9
.5

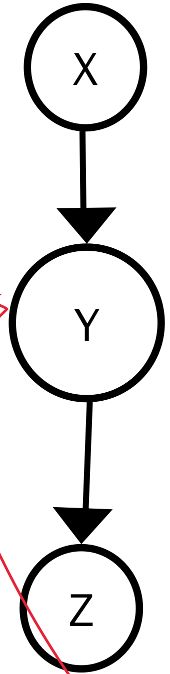
X	P(Y2=1|X)
0	0.9
1	0.7

.1
.3

condition 3

$Y_1 \rightarrow Z \leftarrow Y_2$

Y1	Y2	P(Z=1|Y1, Y2)
0	0	0.2
0	1	0.3
1	0	0.6
1	1	0.8



Y1	Y2	Y	X	P(Y|X)	P(Z=1|Y)
0	0	0	0	.9*.1	0.2
0	1	1	0	.9*.9	.3
1	0	2	0	.1*.1	.6
1	1	3	0	.1*.9	.8
0	0	0	1	.5*.3	
0	1	1	1	.5*.7	
1	0	2	1	.5*.3	
1	1	3	1	.5*.7	

$P(Z=1 | Y_1=0, Y_2=0)$

What we've learned so far

- Pause and spend 5 minutes writing down the main ideas we've learned so far, and how they connect.

What we've learned so far

- Basics of probability and how to use the product rule, Bayes rule and marginalization to do inference.
- How to represent relationships in the world with Bayes nets:
  - *Graph to represent variables and their direct dependencies*
  - *CPTs to represent strength of dependencies*

    parameters
- Noisy-OR model for representing CPTs
- Reasoning about conditional independence between variables in a Bayes' net using d-separation
- General algorithms for inference in Bayes nets:
  - *Enumeration (exponential in number of undefined variables)*
  - *Variable Elimination (linear in # of undefined variables and size of largest CPT for polytrees)*
- How to turn a graph into a polytree (at the cost of the size of the CPT)

learning

1) learn parameters (CPTs)

2) learn the structure (NP-hard)

Up next: Learning in Bayes nets (focused on learning CPTs)

Learning Bayes Nets : data

- Aspects of the Bayes Net might not be know or easy to elicit from experts. This could include:
  - *the structure of the DAG*
  - *the CPTs*
- In this course we will assume a given DAG (structure) and focus on learning CPTs from data:
  - *With complete data (all variables observed)* : MLE
  - *With incomplete data (some variables not observed)* -- LATER : EM

Learning from data via Maximum Likelihood (ML) Learning

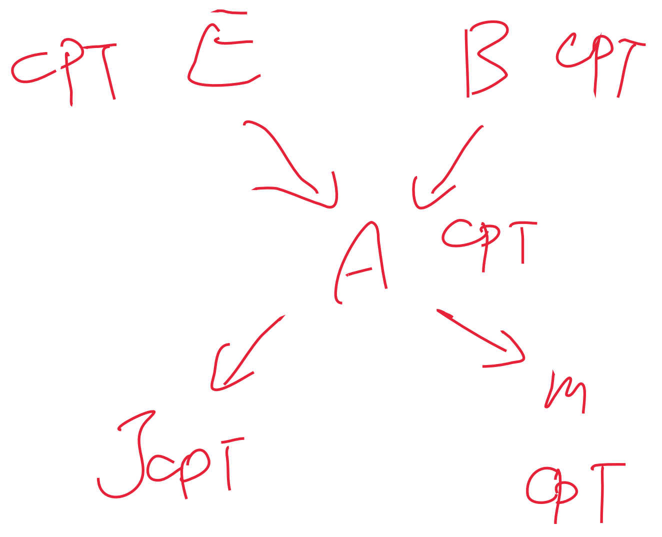
data:	E	B	A	J	M
sample 1 →	0	0	1	1	1
sample 2 →	1	0	1	1	1
sample 3 →	0	0	1	1	1

- ML is the simplest form of learning in BNs
- Idea: Choose (learn) the model (CPTs) that maximizes the probability of the data

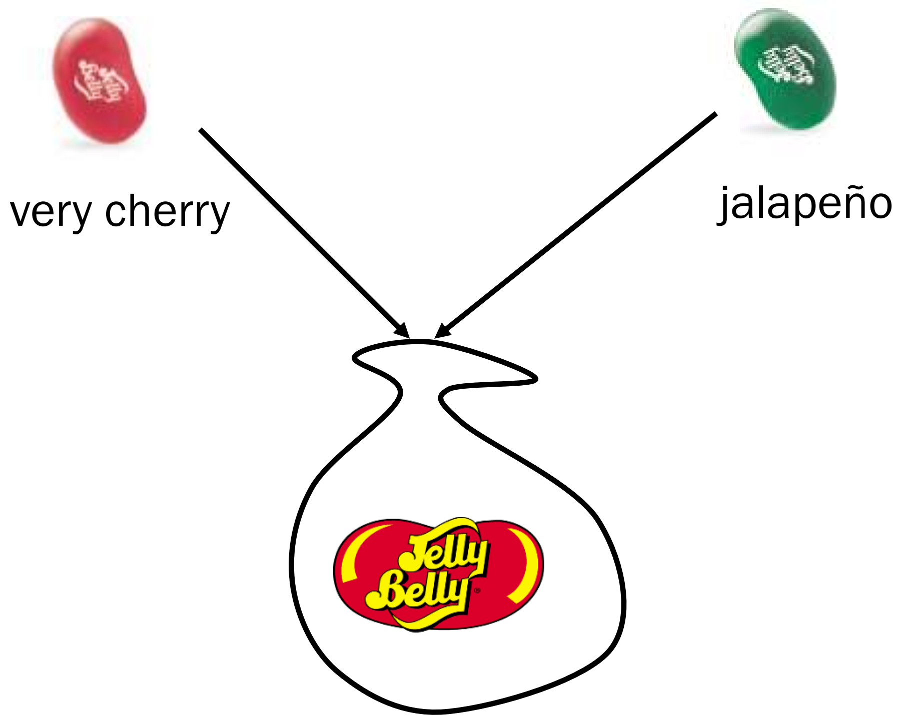
Given the observed data and the BN, ML is to find the values for CPTs that is the most likely to have produced the data.

$$P_{model}(observed\ data)$$

$$\underset{CPTs}{\arg\max}\ P(data \mid CPTs).$$



Simple Example: Estimating the proportion of candies in a bag



Goal: Estimate proportion of cherries/jalapenos by drawing samples

data:

drawing	type
1	cherry
2	jalapeno
3	cherry
4	cherry
⋮	⋮
T	

X CPT

$P(X = \text{cherry})$??

Simple Example: Estimating the proportion of candies in a bag

Goal: Estimate proportion of cherries/jalapenos by drawing samples

X

X: {cherry, jalapeno}

CPT: $P(X = \text{cherry}) = \underline{p}$ $P(X = \text{jalapeno}) = 1 - p$

data: T samples $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots x^{(T)}\}$

$$\arg\max_p P(\underline{\text{data}} \mid p)$$

$$= \arg\max_p P(\underbrace{x^{(1)}, x^{(2)} \dots, x^{(T)}}_{\prod} \mid p)$$

$$= \arg\max_p \prod_{i=1}^{T} P(x^{(i)} \mid p)$$

T samples

where $x^{(i)}$ is either cherry or jalapeno

Simple Example: Estimating the proportion of candies in a bag

Goal: Estimate $p = P(X = cherry)$ by drawing samples

Data: T samples $\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\}$ where each x^{t} is either cherry or jalapeño

The probability of selecting these samples given the parameter $p = P(X = cherry)$ is:

$$\underbrace{P(\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\} | p)}_{\text{The likelihood of } p \text{ for this data}}$$

Assumption: The samples are independently drawn and identically distributed (i.i.d.)
In other words, each sample is drawn using the same *p* and don't depend on each other:

$$P(\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\} | p) = P(X = x^{(1)} | p) P(X = x^{(2)} | p) \dots P(X = x^{(T)} | p) = \prod_{t=1}^{T} P(X = x^{(t)} | p)$$

Simple Example: Estimating the proportion of candies in a bag

Goal: Estimate $p = P(X = cherry)$ by drawing samples

__Data__: T i.i.d. samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}$ where each x^t is either cherry or jalapeño

$$likelihood(p) = P(\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\} \mid p) = \prod_{t=1}^{T} P(X = x^{(t)} \mid p)$$

For a given p, how many possible values can $P(X = x^{(t)})$ take?

A. 1
B. 2
C. 4
D. There is no way to know

cherry p
jalapeno $1-p$

Simple Example: Estimating the proportion of candies in a bag

Goal: Estimate $p = P(X = cherry)$ by drawing samples

<u>Data</u>: T i.i.d. samples $\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\}$ where each x^t is either cherry or jalapeño

$$likelihood(p) = P(\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\}) = \prod_{t=1}^{T} P(X = x^{(t)})$$

How many terms in $\prod_{t=1}^{T} P(X = x^{(t)})$ will equal p?

A. The number of cherry candies in the sample (known to us)
B. The number of jalapeno candies in the sample
C. The total number of candies in the sample
D. You cannot tell from the data given

Simple Example: Estimating the proportion of candies in a bag

Goal: Estimate $p = P(X = cherry)$ by drawing samples

$N_c + N_j = T$

N_c : # of cherries in sample

Data: T i.i.d. samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}$ where each x^t is either cherry or jalapeño

N_j : # of Jalapeno in sample

$$likelihood(p) = P(\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}) = \prod_{t=1}^{T} P(X = x^{(t)}) = p^{N_c}(1-p)^{N_j}$$

We want to choose p that maximizes this function

$$\arg\max_p \; p^{N_c} \cdot (1-p)^{N_j} = \arg\max_p \; \log\left(p^{N_c} \cdot (1-p)^{N_j}\right)$$

$$= \arg\max_p \left(N_c \log p + N_j \log(1-p)\right)$$

Log-Likelihood: An easier function

Goal: Estimate $p = P(X = cherry)$ by drawing samples

Data: T i.i.d. samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}$ where each x^t is either cherry or jalapeño

log-likelihood: $\mathcal{L}(p) = \log P(\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}) = \log \prod_{t=1}^{T} P(X = x^{(t)}) = N_c \log p + N_j \log(1-p)$

For this problem $\mathcal{L}(p) =$

$$\frac{d\ell(p)}{dp} = N_c \cdot \frac{1}{p} + N_j \frac{1}{1-p} \cdot (-1) \triangleq 0$$

$$\frac{N_c}{p} = \frac{N_j}{1-p}$$

$$\underbrace{(N_c + N_j)}_{T} p = N_c$$

$$\boxed{p = \frac{N_c}{T}}$$

Maximize the log-likelihood by taking the derivative

Goal: Estimate $p = P(X = cherry)$ by drawing samples

Data: T i.i.d. samples $\{x^{(1)}, x^{(2)}, \dots, x^{(T)}\}$ where each x^t is either cherry or jalapeño

log-likelihood: $\mathcal{L}(p) = N_c \log p + N_j \log(1-p)$

B.N.

flavor

CPT

flavor	P(flavor)
cherry	$\frac{N_c}{T}$

Marginal Probability

$$= \frac{\text{count of } X = x_i}{\text{total \# of samples}}$$

Estimating Parameters of a Bayes Net

Goal: Estimate $p = P(X = cherry)$ by drawing samples

Flavor

$$P(X = cherry) = p = \frac{N_c}{N_c + N_j}$$

$$P(X = jalapeno) = 1 - p$$

But what about a more complex Bayes Net? ; all data are observed

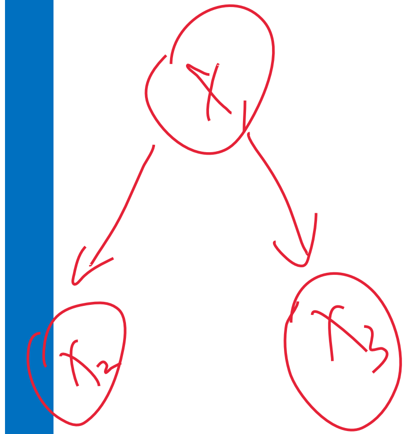
Estimating Parameters of a Bayes Net

Given: A fixed DAG with *n* discrete nodes $\{X_1, X_2, \dots, X_n\}$

Goal: Estimate the values in the CPTs to maximize probability of observed data

5 probabilities

n: 3



data

	X_1	X_2	X_3
1	0	0	0
2	0	0	0
3	0	0	1
4	0	0	0
⋮			
T	1	0	1

CPT

X_1	$P(X_1)$
1	??

X_1	$P(X_2=1 \mid X_1)$
0	??
1	??

X_1	$P(X_3=1 \mid X_1)$
0	??
1	??

Estimating Parameters of a Bayes Net

Parameters to estimate (CPTs of the network): $P(X_i = x|Pa(X_i) = \pi)$

Data set: $\{(x_1{}^{(t)}, x_2{}^{(t)}, \ldots, x_n{}^{(t)})\}_{t=1}^{T}$ where each sample is $\underbrace{\{x_1^{(t)}, x_2^{(t)}, \ldots x_n^{(t)}\}}_{t^{th}\text{ sample}}$

$$\mathcal{L} = \log P(data) \overset{??}{=} \log \boxed{\prod_{t=1}^{T}} P\left(X_1 = x_1{}^{(t)}, X_2 = x_2{}^{(t)}, \ldots, X_n = x_n{}^{(t)}\right)$$

Under what assumption(s) is the above log-likelihood equation true?

√ A. Data is i.i.d.

✗ *B.* $P(X_i|X_1, X_2, \ldots, X_{i-1}) = P(X_i|Parents(X_i))$

✗ C. T > 1 (you have more than one data point)

D. More than one of the above

E. None of the above (it is generally true)