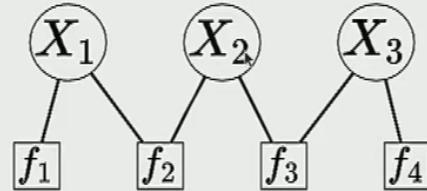


Bayesian Networks 1 - Inference

Review: definition



Definition: factor graph

Variables:

$X = (X_1, \dots, X_n)$, where $X_i \in \text{Domain}_i$

Factors:

f_1, \dots, f_m , with each $f_j(X) \geq 0$

$$\text{Weight}(x) = \prod_{j=1}^m f_j(x)$$

Stanford

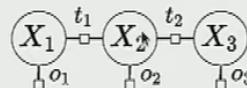
Specify locally (Factors) and optimize globally (Weight) Last time we talked about different algo for finding the maximum weight

Review: object tracking



Problem: object tracking

Sensors report positions: 0, 2, 2. Objects don't move very fast and sensors are a bit noisy. What path did the object take?



- Variables X_i : location of object at time i
- Transition factors $t_i(x_i, x_{i+1})$: incorporate physics
- Observation factors $o_i(x_i)$: incorporate sensors

[demo: maxVariableElimination()]

Stanford

4

CS221 / Autumn 2019 / Liang & Sadigh

The framework is already good, with this you can come up with a lot. But what are these factors mean and how do we come up with them? philosophically we might be quite bothered by this

So the goal of this lecture is to give more meaning to the factors, and bayesian network is the way to do that. In a line, bayesian networks are factor graphs plus probability

Previously we already talked about a lot of modeling: search games, MDPs And then we talked about modeling when the order of actions does not matter so much, and it is more nature to think it as variables and assignment And now we will move on to bayesian networks where it will be a higher level of abstraction

Basics



Review: probability

Random variables: sunshine $S \in \{0, 1\}$, rain $R \in \{0, 1\}$

Joint distribution:

s	r	$\mathbb{P}(S = s, R = r)$
0	0	0.20
0	1	0.08
1	0	0.70
1	1	0.02

Notations: Capital letters (S, R): random variables smaller letters (s,r): values that random variables can take

Notation P with Capital letters $P(S, R)$: the whole probability distributions, eg $P(S, R)$ is the distribution table

Notation P with $=s, =r$ assignments $P(S=s, R=r)$: represents a single number, which is a probability, eg: $P(S=1, R=0) = 0.7$



Review: probability

Random variables: sunshine $S \in \{0, 1\}$, rain $R \in \{0, 1\}$

Joint distribution:

s	r	$\mathbb{P}(S = s, R = r)$
0	0	0.20
0	1	0.08
1	0	0.70
1	1	0.02

Marginal distribution:

s	$\mathbb{P}(S = s)$
0	0.28
1	0.72

(aggregate rows)

Conditional distribution:

s	$\mathbb{P}(S = s R = 1)$
0	0.8
1	0.2

(select rows, normalize)

Marginal distribution: eg I don't care about r , just wanna focus on s > Sum up probs when $s = 0, s = 1$ and you get the table

Conditional: Select rows based on condition, normalize them and let them sum to 1. normalize (a,b) means $a' = a/(a+b)$, $b' = b/(a+b)$

Probabilistic inference

Joint distribution (probabilistic database):

$$\mathbb{P}(S, R, T, A)$$

Probabilistic inference:

- **Condition** on evidence (traffic, autumn): $T = 1, A = 1$
- Interested in **query** (rain?): R

$$\mathbb{P}(\underbrace{R}_{\text{query}} \mid \underbrace{T = 1, A = 1}_{\text{condition}})$$

(S is marginalized out)

Think of joint distribution as a database

Probabilistic inference: You observe some evidence, that's what we know and what we like to find out is weather it's raining

Is that reaning under condition of we know it is autumn and traffic

Challenges: the joint distibution table can he really huge Cool thing about bayesian networks is that it allows us to define joint distibution using the language of factor graphs

Challenges

Modeling: How to specify a joint distribution $\mathbb{P}(X_1, \dots, X_n)$ **compactly**?

Bayesian networks (factor graphs to specify joint distributions)

Inference: How to compute queries $\mathbb{P}(R \mid T = 1, A = 1)$ **efficiently**?

Variable elimination, Gibbs sampling, particle filtering
(analogue of algorithms for finding maximum weight assignment)



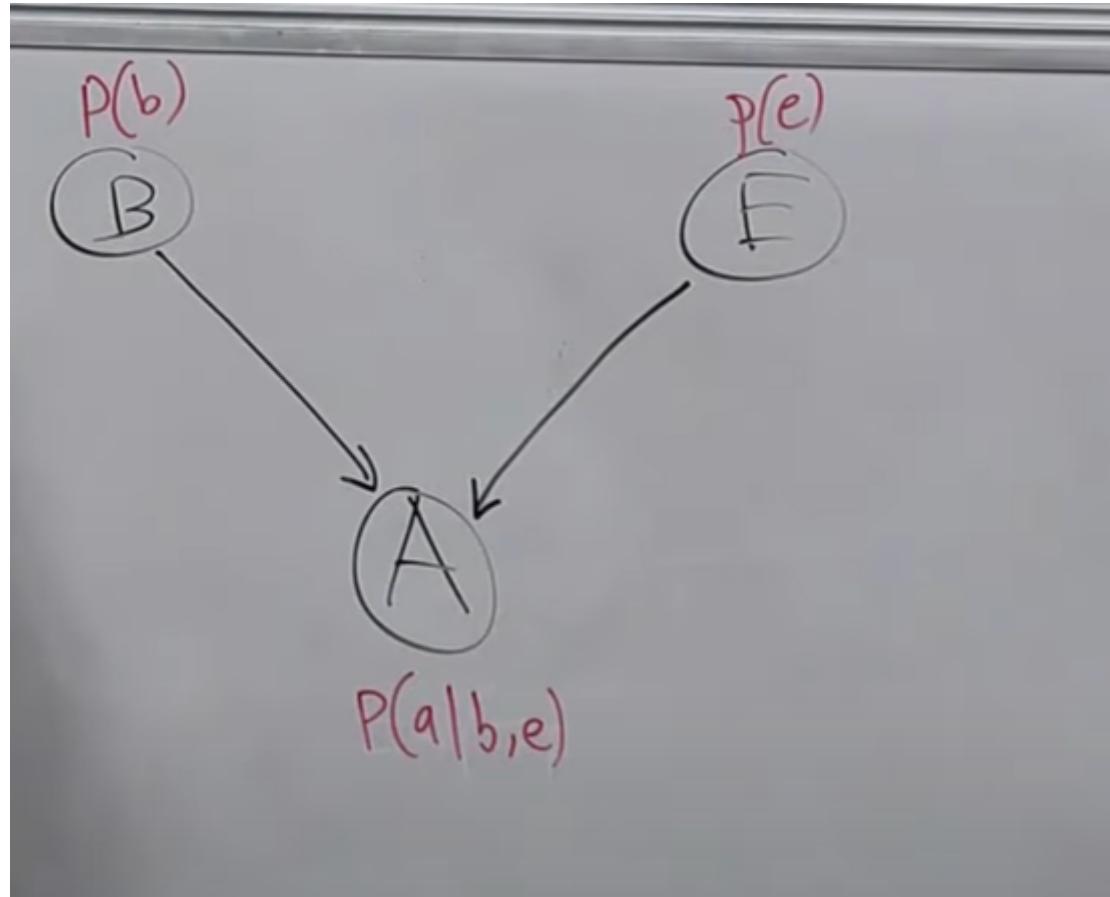
Question

Earthquakes and burglaries are independent events that will cause an alarm to go off. Suppose you hear an alarm. How does hearing on the radio that there's an earthquake change your beliefs about burglary?

it increases the probability of burglary

it decreases the probability of burglary

it does not change the probability of burglary

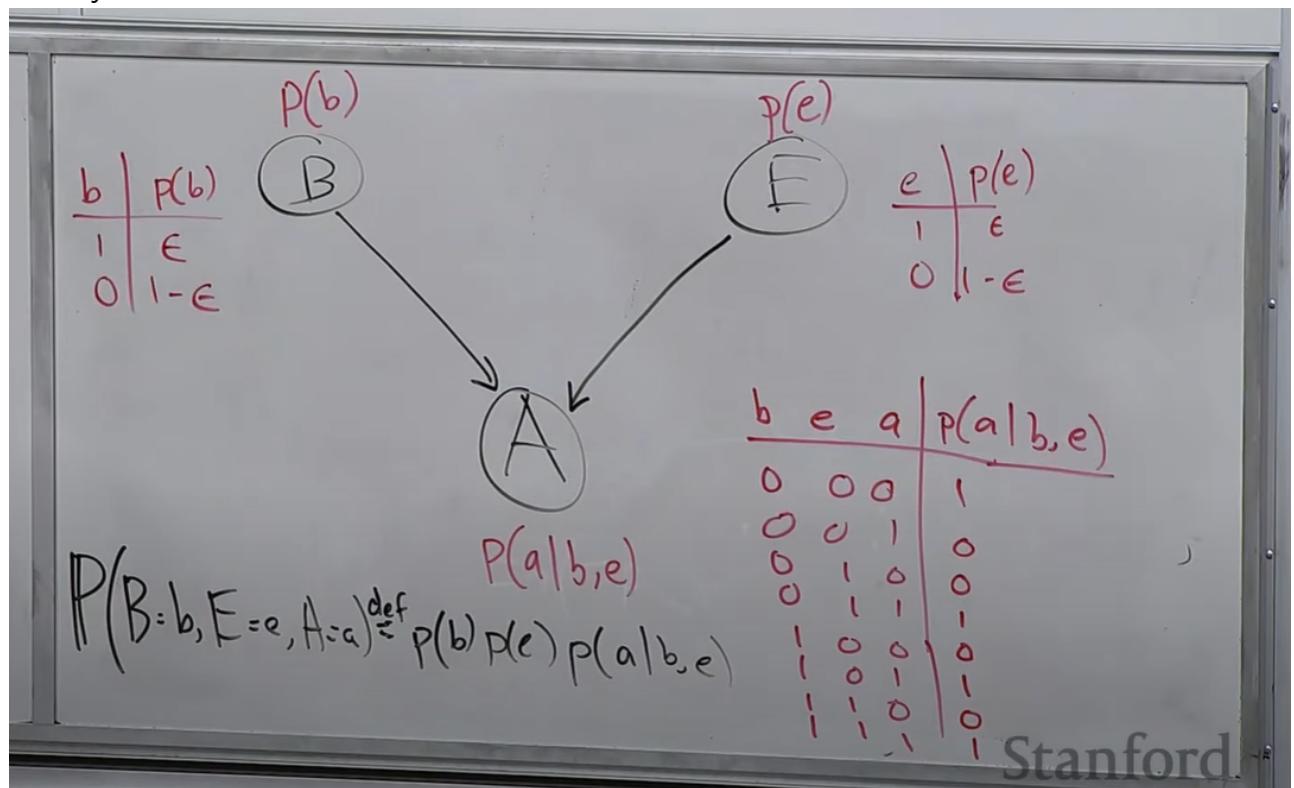


A: prob if alarm going off

2. Define local conditional distribution

a local conditional distribution is: P of whatever that variable is, given its parents, parents are the variable directly point into it In this example the parents of **a** is **b** and **e**

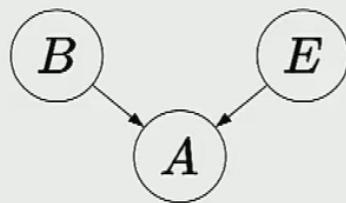
1. define joint distribution



Stanford

The small p stands for local conditional distribution, it's the thing we need to define (it is the ground truth cuz we defined it) The capital P stands for joint distribution

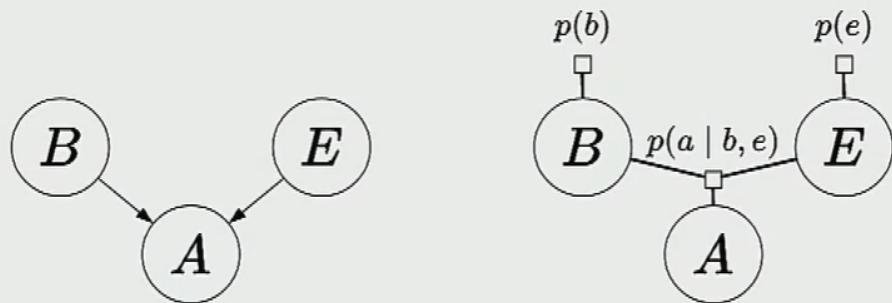
Bayesian network (alarm)



$$\mathbb{P}(B=b, E=e, A=a) = p(b)p(e)p(a | b, e)$$

Okay this is bayesian networks, and what's the connection between this and factor graphs This looks like
 weight = product of factors

Bayesian network (alarm)



$$\mathbb{P}(B = b, E = e, A = a) = p(b)p(e)p(a | b, e)$$

Bayesian networks are a special case of factor graphs!

Note: single factor that connects **all** parents!

In this special case: one factor per variable has a single factor connect all parents

Probabilistic inference (alarm)

Joint distribution:

b	e	a	$\mathbb{P}(B = b, E = e, A = a)$
0	0	0	$(1 - \epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1 - \epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1 - \epsilon)$
1	1	0	0
1	1	1	ϵ^2

Queries: $\mathbb{P}(B)$? $\mathbb{P}(B | A = 1)$? $\mathbb{P}(B | A = 1, E = 1)$?

[demo: $\epsilon = 0.05$]

Examples: [vote] [csp] [pair] [chain] [track] [alarm] [med] [dep] [delay] [mln] [new]
[\[Background\]](#) [\[Documentation\]](#)

```
// Alarm network (a Bayesian network)
// (B)urglary, (E)arthquake, (A)larm
variable('B', [0, 1])
variable('E', [0, 1])
variable('A', [0, 1])

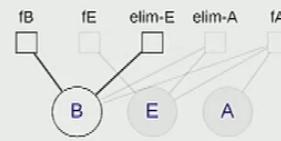
// Small probability of something happening
factor('fB', 'B', {0: 0.95, 1: 0.05})
factor('fE', 'E', {0: 0.95, 1: 0.05})
// If something happens, alarm
factor('fA', 'B E A', function(b, e, a) {
    return (b || e) == (a == 1);
})

// Alarm went off: B,E no longer independent
condition('A', 1)

// No earthquake: explaining away phenomenon
condition('E', 1)

query('B') // Was there a burglary?
sumVariableElimination({order: 'A E'})
```

Query: $\mathbb{P}(B)$
Algorithm: **variable elimination (sum)**



Algorithm done.
Final factor: final

B	final(B)	$\mathbb{P}(B)$
0	0.95	0.95
1	0.05	0.05

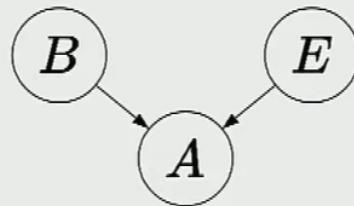
←

B	fB(B)	$\mathbb{P}(B)$
0	0.95	0.95
1	0.05	0.05

B	elim-E(B)	$\mathbb{P}(B)$
0	0.95	0.95
1	0.05	0.05



Explaining away



Key idea: explaining away

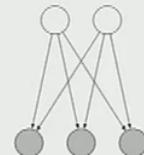
Suppose two causes positively influence an effect. Conditioned on the effect, conditioning on one cause reduces the probability of the other cause.

So given the alarms goes off and there's a bulargruy, decrease the prob of earthquake

We might doubt that, **E** and **B** are independent. That's true but when we are conditioning on **A**, we already changed the independent structure of the model, which means when alarm goes off, **E** and **B** are no longer independent

Definition

Definition



Definition: Bayesian network

Let $X = (X_1, \dots, X_n)$ be random variables.

A **Bayesian network** is a directed acyclic graph (DAG) that specifies a joint distribution over X as a product of local conditional distributions, one for each node:

$$\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) \stackrel{\text{def}}{=} \prod_{i=1}^n p(x_i | x_{\text{Parents}(i)})$$

$x_{\text{parents}(i)}$ means the value assigned to parents of i

Special properties

Key difference from general factor graphs:



Key idea: locally normalized

All factors (local conditional distributions) satisfy:

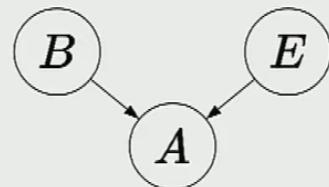
$$\sum_{x_i} p(x_i \mid x_{\text{Parents}(i)}) = 1 \text{ for each } x_{\text{Parents}(i)}$$

Implications:

- Consistency of sub-Bayesian networks
- Consistency of conditional distributions

if we summarize all possible values X_i can take on, it should be 1, and it is true for every settings of X_{parents}

Consistency of sub-Bayesian networks



A short calculation:

$$\mathbb{P}(B = b, E = e) \stackrel{\text{def}}{=} \sum_a \mathbb{P}(B = b, E = e, A = a)$$

$$\stackrel{\text{def}}{=} \sum_a p(b)p(e)p(a \mid b, e)$$

$$= p(b)p(e) \sum_a p(a \mid b, e)$$

$$= p(b)p(e)$$

Stanford

def means by the laws of probability So by the laws of probability we can take $p(b)p(e)$ out side

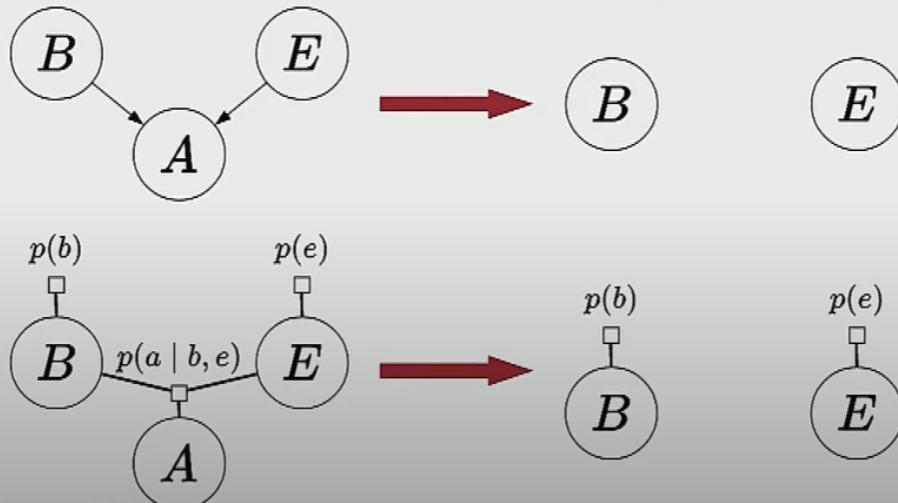
We've made an algebraic operation graphically

Consistency of sub-Bayesian networks



Key idea: marginalization

Marginalization of a leaf node yields a Bayesian network without the node.



Stanford

You marginalize a leaf node in Bayesian network, you can just drop it from the graph

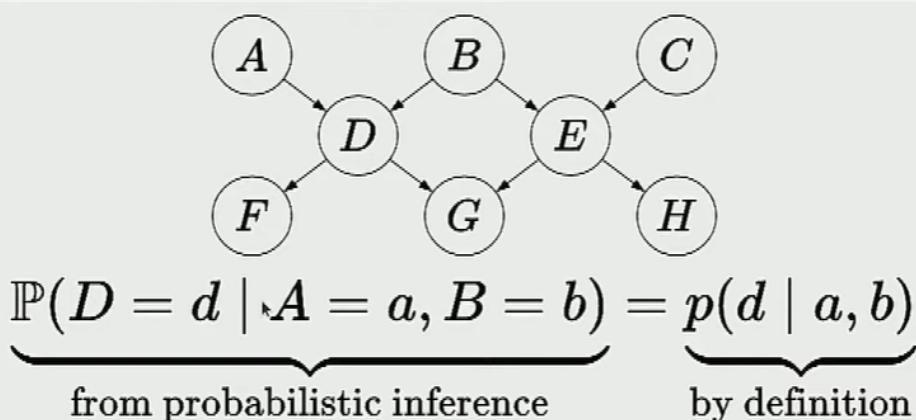
Consistency of local conditionals

Consistency of local conditionals



Key idea: local conditional distributions

Local conditional distributions (factors) are the true conditional distributions.



They are equal



Medical diagnosis



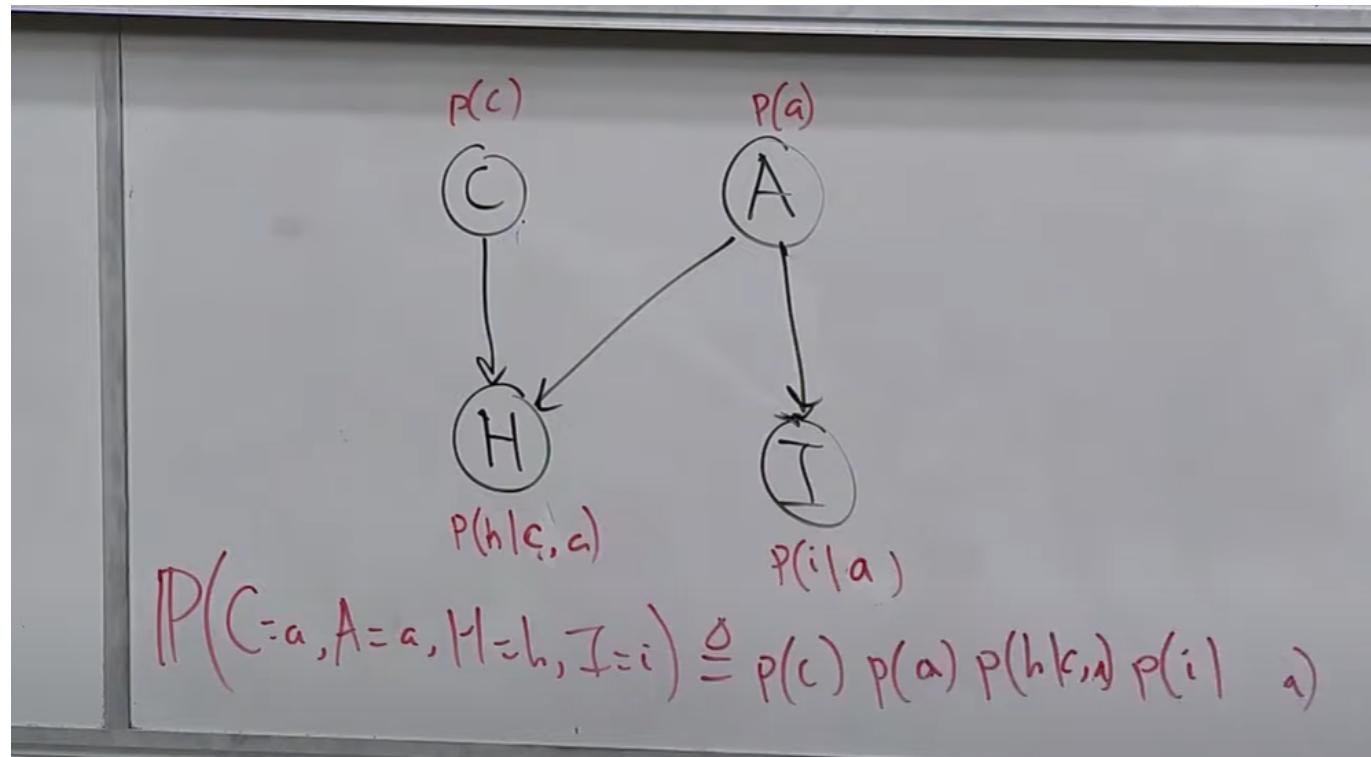
Problem: cold or allergies?

You are coughing and have itchy eyes. Do you have a cold or allergies?

[whiteboard]

Let's define it with procedure

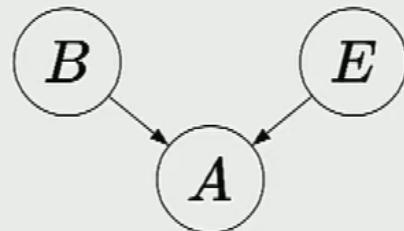
- ① Variables
- ② Edges
- ③ Local cond. distrib.
- ④ Joint distrib.



h: coughing



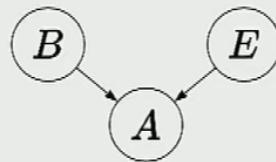
Summary so far



- Random variables capture state of world
- Edges between variables represent dependencies
- Local conditional distributions \Rightarrow joint distribution
- Probabilistic inference: ask questions about world
- Captures reasoning patterns (e.g., explaining away)
- Factor graph interpretation (for inference later)

Probabilistic programs

Probabilistic programs



Probabilistic program: alarm

$B \sim \text{Bernoulli}(\epsilon)$
 $E \sim \text{Bernoulli}(\epsilon)$
 $A = B \vee E$



Key idea: probabilistic program

A randomized program that sets the random variables.

```
def Bernoulli(epsilon):
    return random.random() < epsilon
```

Stanford

Bernoulli: returns true with prob epsilon

define a distribution?

Probabilistic program: example



Probabilistic program: object tracking

$$X_0 = (0, 0)$$

For each time step $i = 1, \dots, n$:

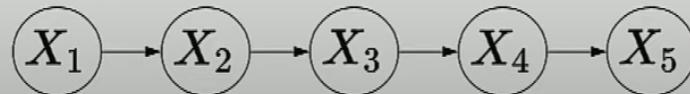
With probability α :

$$X_i = X_{i-1} + (1, 0) \text{ [go right]}$$

With probability $1 - \alpha$:

$$X_i = X_{i-1} + (0, 1) \text{ [go down]}$$

Bayesian network structure:



Stanford

this program reduces to a particular bayesian network structure, where each X_i is only connected to X_{i-1}

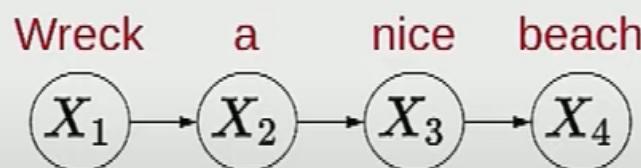
Application: language modeling



Probabilistic program: Markov model

For each position $i = 1, 2, \dots, n$:

Generate word $X_i \sim p(X_i | X_{i-1})$



For every position, generate a word given previous word

Application: object tracking

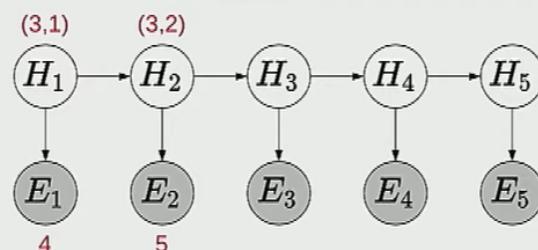


Probabilistic program: hidden Markov model (HMM)

For each time step $t = 1, \dots, T$:

Generate object location $H_t \sim p(H_t | H_{t-1})$

Generate sensor reading $E_t \sim p(E_t | H_t)$



Inference: given sensor readings, where is the object?

Convention: If shade a variable, means u observe it If not shade a variable, means u can't/don't observe it

Application: multiple object tracking



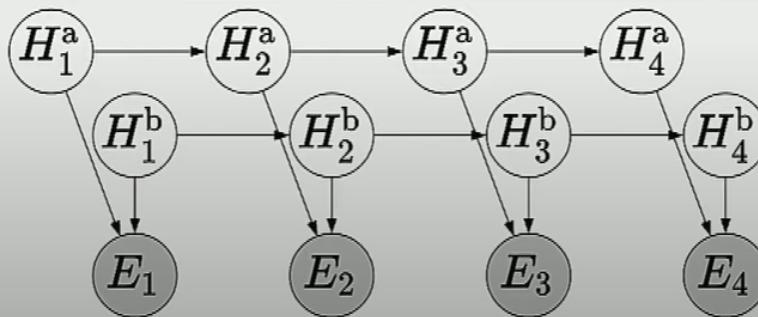
Probabilistic program: factorial HMM

For each time step $t = 1, \dots, T$:

For each object $o \in \{a, b\}$:

Generate location $H_t^o \sim p(H_t^o | H_{t-1}^o)$

Generate sensor reading $E_t \sim p(E_t | H_t^a, H_t^b)$

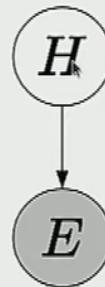


Stanford

We have two objects and two independent markov chains running. At each time step, I only observe one sensor reading, and that sensor reading is gonna be some combination of the actual objects a and b.



Summary so far



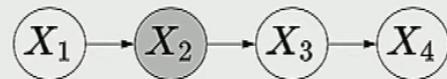
- Many many different types of models
- Mindset: come up with stories of how the data (input) was generated through quantities of interest (output)
- Opposite of how we normally do classification!

It's like another way around, we start with output. We first define the model.

A set of variables H, which you don't observe, that generates/causes a set of variables E, which you do observe.

Inference

Example: Markov model



Query: $\mathbb{P}(X_3 = x_3 \mid X_2 = 5)$ for all x_3

Tedious way:

$$\begin{aligned} &\propto \sum_{x_1, x_4} p(x_1)p(x_2 = 5 \mid x_1)p(x_3 \mid x_2 = 5)p(x_4 \mid x_3) \\ &\propto \left(\sum_{x_1} p(x_1)p(x_2 = 5 \mid x_1) \right) p(x_3 \mid x_2 = 5) \\ &\propto p(x_3 \mid x_2 = 5) \end{aligned}$$

Fast way:

[whiteboard]

Stanford

General strategy

Query:

$$\mathbb{P}(Q \mid E = e)$$



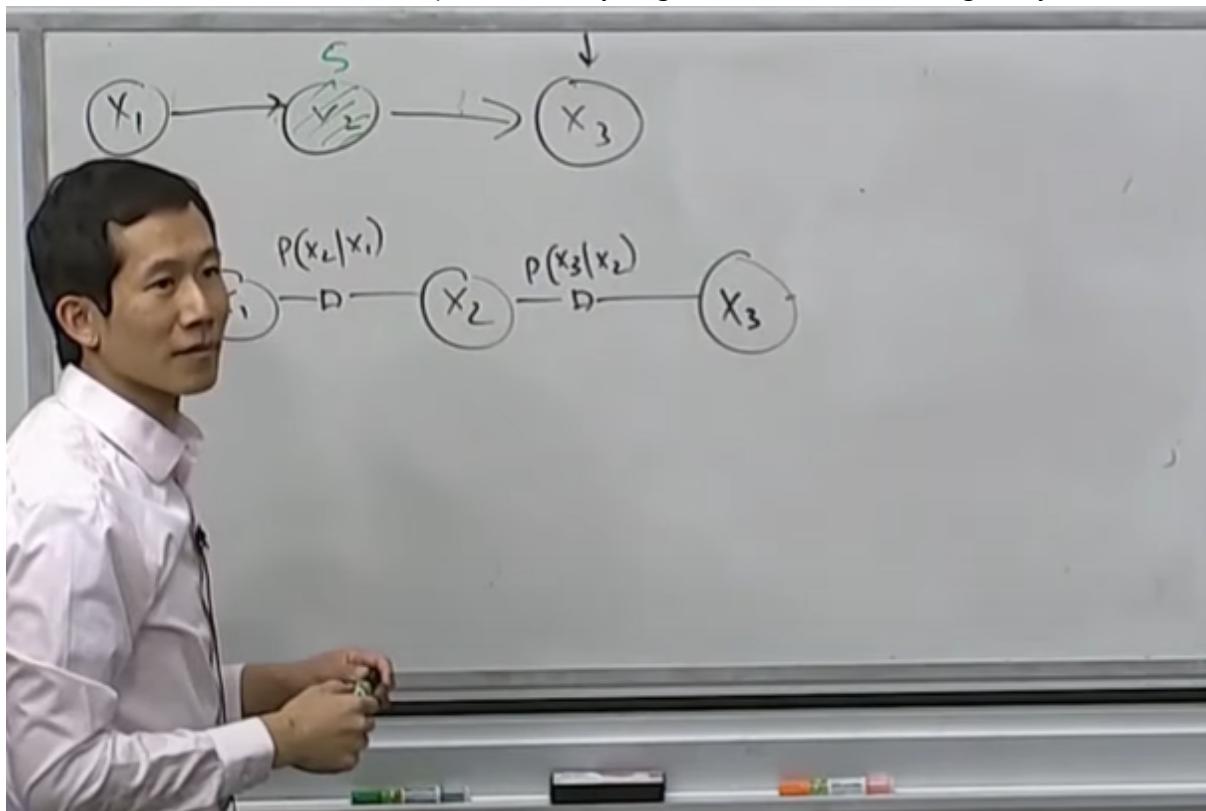
Algorithm: general probabilistic inference strategy

- Remove (marginalize) variables that are not ancestors of Q or E .
- Convert Bayesian network to factor graph.
- Condition on $E = e$ (shade nodes + disconnect).
- Remove (marginalize) nodes disconnected from Q .
- Run probabilistic inference algorithm (manual, variable elimination, Gibbs sampling, particle filtering).

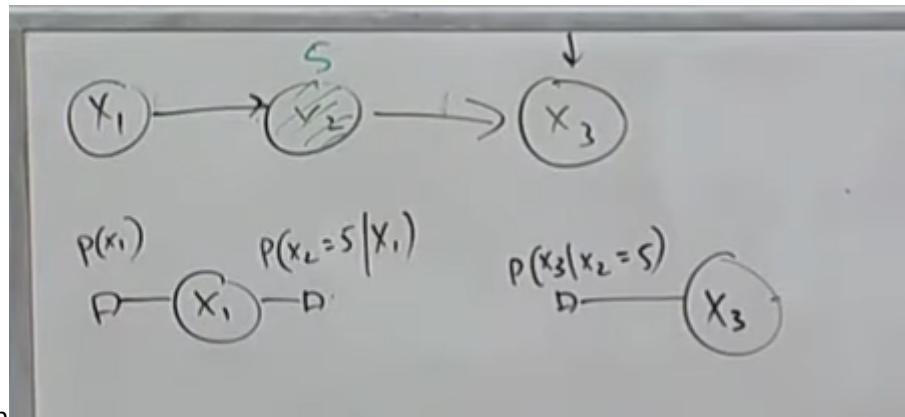
Stanford

1. We wanna remove as much as variables as I can
 1. Marginlize non-ancestors, meaning that anything upstream i can keep for now, anything down stream I can let go
2. Convert to a factor graph

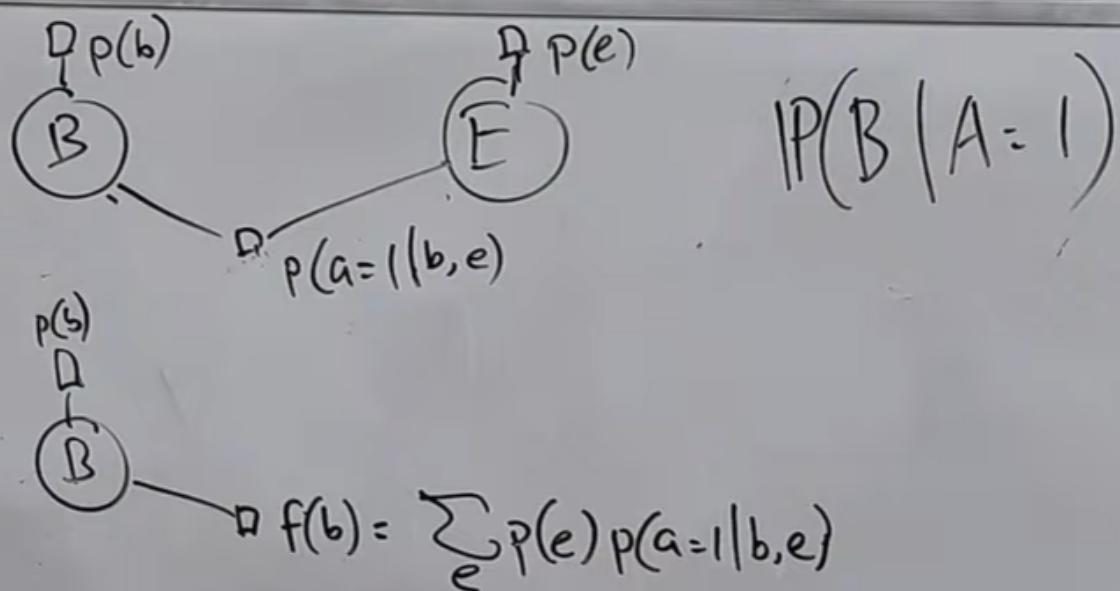
1. Remove the directions(cuz it's complex and easily to get confused), make things easy



3. Condition on evidence, we are conditioning on X_2 , so we wrap it up and change the factors to be a



4. Marginalize out the disconnected components (We care about X_3 only, since they are disconnected, we can drop it since it's not related)
5. Do work, you might not just left with one node, in this step we need to solve the factor graph



$$P(B=1 | A=1) = \frac{1}{2-\epsilon}$$