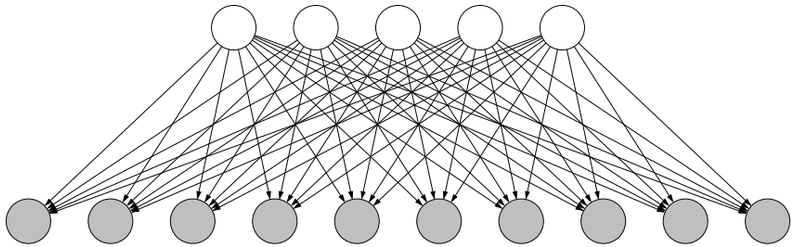
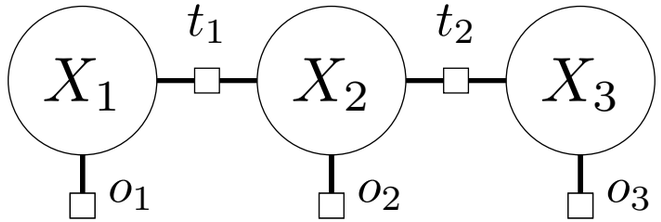
Bayesian networks: overview

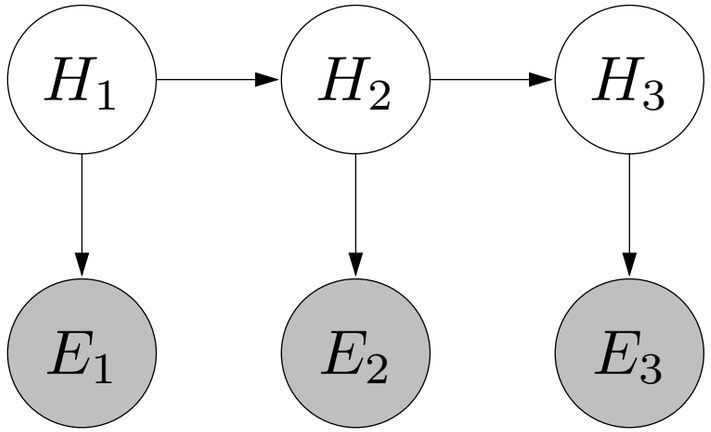


☆ factors have special meaning

Markov networks versus Bayesian networks

Both define a joint probability distribution over assignments





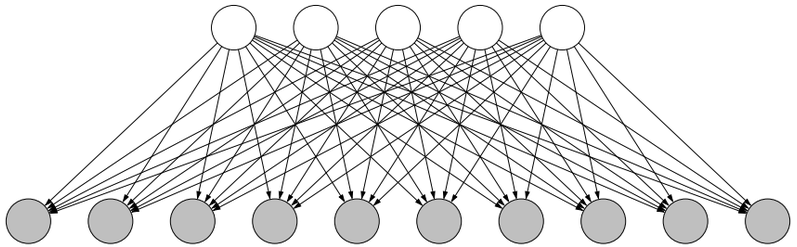
Markov networks	Bayesian networks
arbitrary factors	local conditional probabilities
set of preferences	generative process

- Before defining Bayesian networks, it is helpful to compare and contrast Markov networks and Bayesian networks at a high-level.
- Both define a joint probability distribution over assignments, and in the end, both are backed by factor graphs.
- But the way each approaches modeling is different. In Markov networks, the factors can be arbitrary, so you should think about being able to write down an arbitrary set of preferences and constraints and just throw them in. In the object tracking example, we slap on observation and transition factors.
- Bayesian networks require the factors to be a bit more coordinated with each other. In particular, they should be local conditional probabilities, which we'll define in the next module.
- We should think about a Bayesian network as defining a generative process represented by a directed graph. In the object tracking example, we think of an object as moving from position H_{i-1} to position H_i and then yielding a noisy sensor reading E_i.

Why Bayesian networks?

- Handle **heterogenously** missing information, both at training and test time
  - First, in traditional machine learning (e.g., linear models or neural networks), the input is usually of a fixed size (homogenous). With Bayesian networks, the types of inputs one can handle can be **hetereogenous** (e.g., missing features), both during training and test times.
- Incorporate **prior** knowledge (e.g., Mendelian inheritance, laws of physics)
  - Second, Bayesian networks offer most leverage when you have rich **prior knowledge** (e.g., Mendelian inheritance, laws of physics). Th allows one to often learn from very few samples and extrapolate beyond distribution of the training data. In contrast, deep neural networl generallly requires much more data to be effective.
- Can **interpret** all the intermediate variables
  - Third, because Bayesian networks are often carefully constructed based on prior knowledge, the variables in the Bayesian network are **interpretable** (more so that hidden units in a neural network), and you can ask questions about any of them via the laws of probability.
- Precursor to **causal** models (can do interventions and counterfactuals)

Bayesian networks: definitions



Review: probability

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

Joint distribution (probabilistic database):

$\mathbb{P}(S, R) =$

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:

(aggregate rows)

$\mathbb{P}(S) =$

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

0.2 + 0.08

Conditional distribution:

(select rows, normalize)

$\mathbb{P}(S \mid R = 1) =$

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

$\frac{0.08}{0.08 + 0.02}$

Review: probability

Variables: S (sunshine), R (rain), T (traffic), A (autumn)

Joint distribution (probabilistic database):

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

Marginal conditional distribution (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**)

A puzzle

Problem: earthquakes, burglaries, and alarms

Earthquakes and **burglaries** are independent events (probability ϵ).

Either will cause an **alarm** to go off.

Suppose you get an **alarm**.

Does hearing that there's an **earthquake** increase, decrease, or keep constant the probability of a **burglary**?

★ conditional independence

Joint distribution:

$$\mathbb{P}(E, B, A)$$

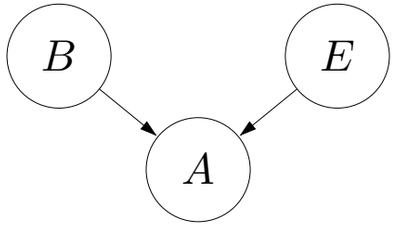
Questions:

$$\mathbb{P}(B = 1 \mid A = 1) \quad ? \quad \mathbb{P}(B = 1 \mid A = 1, E = 1)$$

burglary given alarm burglary given alarm & earthquake

Bayesian network (alarm)

B & E are independent



b	$p(b)$
1	ϵ
0	$1-\epsilon$

e	$p(e)$
1	ϵ
0	$1-\epsilon$

b	e	a	$p(a \mid b,e)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

indicator

$$p(b) = \epsilon \cdot [b = 1] + (1 - \epsilon) \cdot [b = 0]$$

$$p(e) = \epsilon \cdot [e = 1] + (1 - \epsilon) \cdot [e = 0]$$

$$p(a \mid b, e) = [a = (b \vee e)]$$

bc independent

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

- Now let us define the joint distribution. Recall the first step was just to define the three variables, B (burglary), E (earthquake), and A (alarm).
- Second, we connect up the variables to model the dependencies. Unlike in factor graphs, these dependencies are represented as **directed edges**. You can intuitively think about the directionality as representing causality, though what this actually means is a more complex issue and beyond the scope of this module.
- Third, for each variable, we specify a **local conditional distribution** of that variable given its parent variables. In this example, B and E have no parents while A has two parents, B and E. This local conditional distribution is what governs how a variable is generated.
- Fourth, we define the joint distribution over all the random variables as the product of all the local conditional distributions.
- Note that we write the local conditional distributions using p, while $\mathbb{P}$ is reserved for the joint distribution over all random variables, which is defined as the product.

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

Questions:

$$\mathbb{P}(B=1) = \epsilon(1-\epsilon) + \epsilon^2 = \epsilon$$

$$\mathbb{P}(B=1 \mid A=1) = \frac{\epsilon(1-\epsilon)+\epsilon^2}{\epsilon(1-\epsilon)+\epsilon^2+(1-\epsilon)\epsilon} = \frac{1}{2-\epsilon}$$

$$\mathbb{P}(B=1 \mid A=1, E=1) = \frac{\epsilon^2}{\epsilon^2+(1-\epsilon)\epsilon} = \epsilon$$

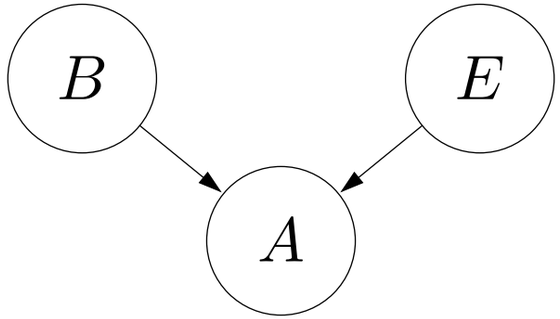
$\frac{1}{2-\epsilon} \geq \epsilon$

[demo]

News flash: earthquakes decrease burglarlies!*

*This is not a causal statement!

Explaining away



Key idea: explaining away

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

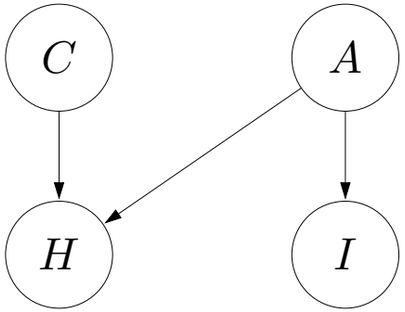
$$\mathbb{P}(B = 1 \mid A = 1, E = 1) < \mathbb{P}(B = 1 \mid A = 1)$$

Note: happens even if causes are independent!

Medical diagnosis

Problem: cold or allergies?

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



step 2)

1) Random variables:

cold C, allergies A, cough H, itchy eyes I

4) Joint distribution:

step 3)

$$\mathbb{P}(C = c, A = a, H = h, I = i) = p(c)p(a)p(h \mid c, a)p(i \mid a)$$

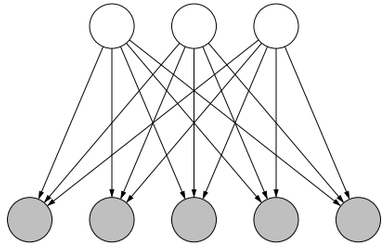
Questions:

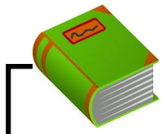
$$\mathbb{P}(C = 1 \mid H = 1) = 0.28 \qquad \mathbb{P}(C = 1 \mid H = 1, I = 1) = 0.13$$

[demo]

- Here is another example (a cartoon version of Bayesian networks for medical diagnosis).
- Step 1: identify all the relevant variables.
- Step 2: draw arrows between them, using prior knowledge. Using our simplistic medical knowledge, suppose that a cough can be either because of a cold or because of allergies, but itchy eyes are generally only caused by allergies.
- Step 3: define a local conditional distribution for each variable.
- Step 4: multiply all the local conditional distributions to form the joint distribution.
- Now we have our probabilistic database and we can ask questions about it. Our motivating question is $\mathbb{P}(C, A \mid H = 1, I = 1)$.
- You can try the demo to get a quantitative answer. Note that $\mathbb{P}(C = 1 \mid H = 1) = 0.28$, which is another example of explaining away. Observing itchy eyes provides evidence for A, which explains away the cough ($H = 1$), resulting in a reduced probability of cold ($C = 1$).
- Note that even qualitatively reasoning about even a four-node Bayesian network can be quite subtle, let alone getting quantitative answers on large Bayesian networks. But we can rest at ease since the laws of probability make sure that all these calculations are internally consistent provided we defined the Bayesian network correctly (which in practice is an admittedly hard modeling task).

Bayesian network (definition)





Definition: Bayesian network

Let $X = (X_1, \ldots, X_n)$ 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:

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

- Second, we have a directed **acyclic** graph over the variables that captures the qualitative dependencies.
- Third, we specify a local conditional distribution for each variable X_i, which is a function that specifies a distribution over X_i given an assignment $x_{\text{Parents}(i)}$ to its parents in the graph (possibly no parents).
- Finally, the joint distribution is simply **defined** to be the product of all of the local conditional distributions.
- Notationally, we use lowercase p (in $p(x_i \mid x_{\text{Parents}(i)})$) to denote a local conditional distribution, and uppercase $\mathbb{P}$ to denote the induced joint distribution over all variables. While we will see that the two coincide, it is important to keep these things separate in your head!

Probabilistic inference (definition)

Input

Bayesian network: $\mathbb{P}(X_1, \ldots, X_n)$

Evidence: $E = e$ where $E \subseteq X$ is subset of variables

Query: $Q \subseteq X$ is subset of variables

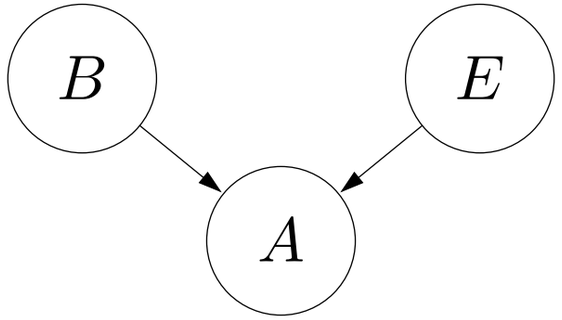
Output

$\mathbb{P}(Q \mid E = e)$  $\mathbb{P}(Q = q \mid E = e)$ for all values q

Example: if coughing and itchy eyes, have a cold?

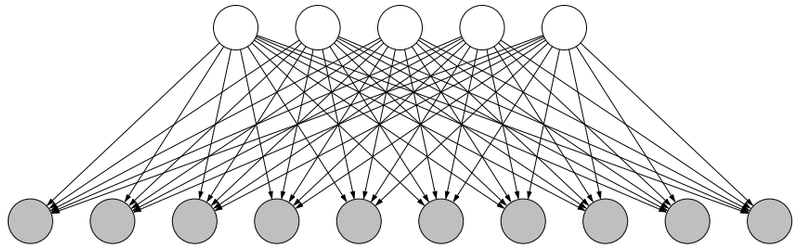
$$\mathbb{P}(C \mid H = 1, I = 1)$$

Summary



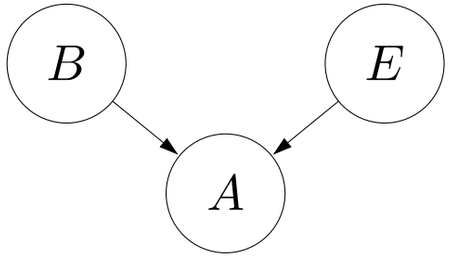
- Random variables capture state of world
- Directed edges between variables represent dependencies
- Local conditional distributions $\Rightarrow$ joint distribution
- Probabilistic inference: ask questions about world
- Captures reasoning patterns (e.g., explaining away)

Bayesian networks: probabilistic programming



- In this module, I will talk about probabilistic programming, a new way to think about defining Bayesian networks through the lens of writing programs, which really highlights the generative process aspect of Bayesian networks.

Probabilistic programs



Joint distribution:

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

Probabilistic program: alarm

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

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

Key idea: probabilistic program

A randomized program that sets the random variables.

Probabilistic program: example

Probabilistic program: object tracking

$X_0 = (0, 0)$

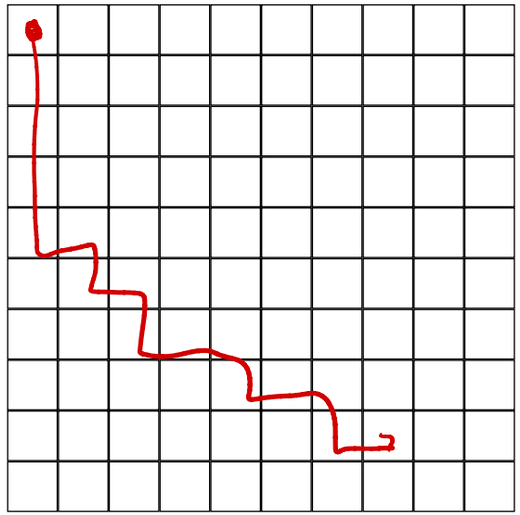
For each time step $i = 1, \ldots, n$:

  if Bernoulli(α): go right w/ p α

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

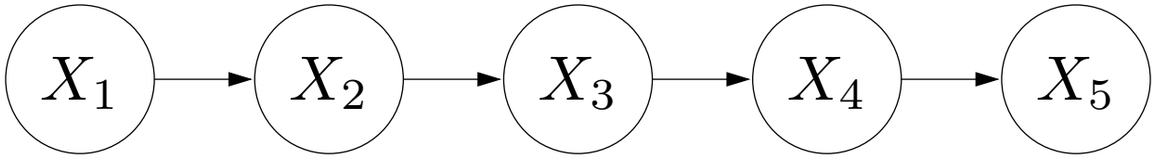
  else: $1 - \alpha$

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



(press ctrl-enter to save)

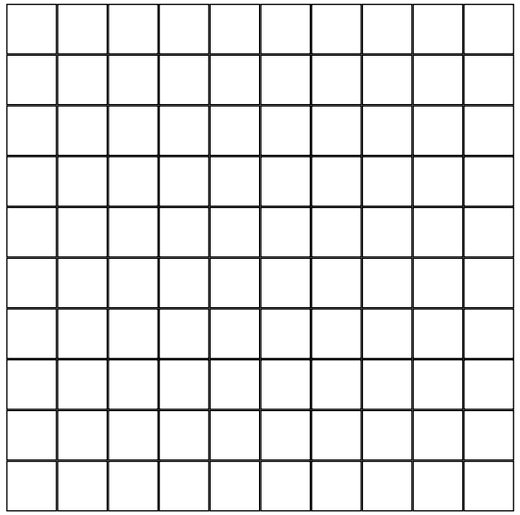
Run



Probabilistic inference: example

Question: what are possible trajectories given **evidence** $X_{10} = (8, 2)$?

Sample w/ condition
s $X_{10} = (8, 2)$

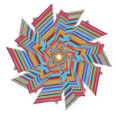


(press ctrl-enter to save)

Run

Application: language modeling

Can be used to score sentences for speech recognition or machine translation

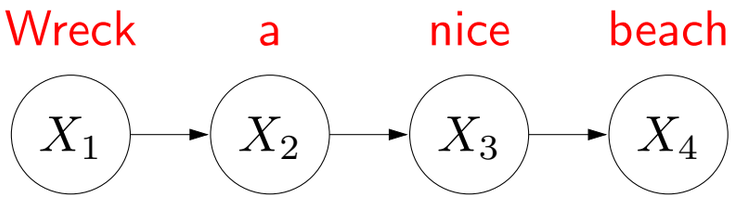


Probabilistic program: Markov model

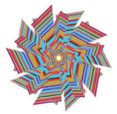
For each position $i = 1, 2, \ldots, n$:

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

generate word based on previous



Application: object tracking

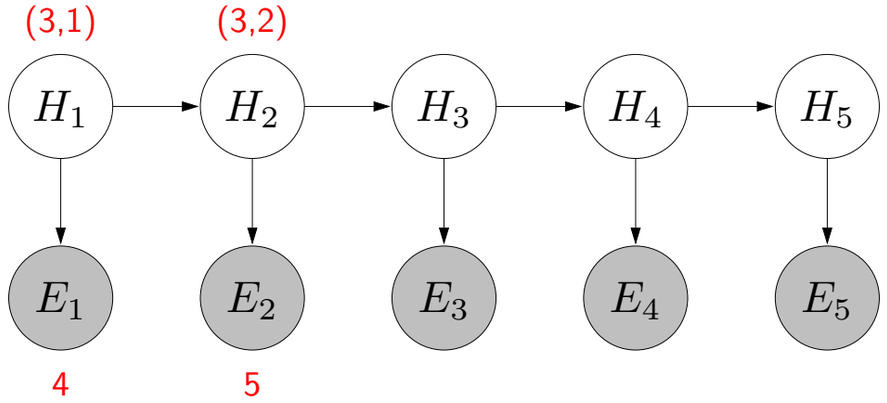


Probabilistic program: hidden Markov model (HMM)

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

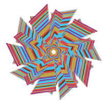
Generate object location $H_t \sim p(H_t \mid H_{t-1})$

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



Inference: given sensor readings, where is the object?

Application: multiple object tracking



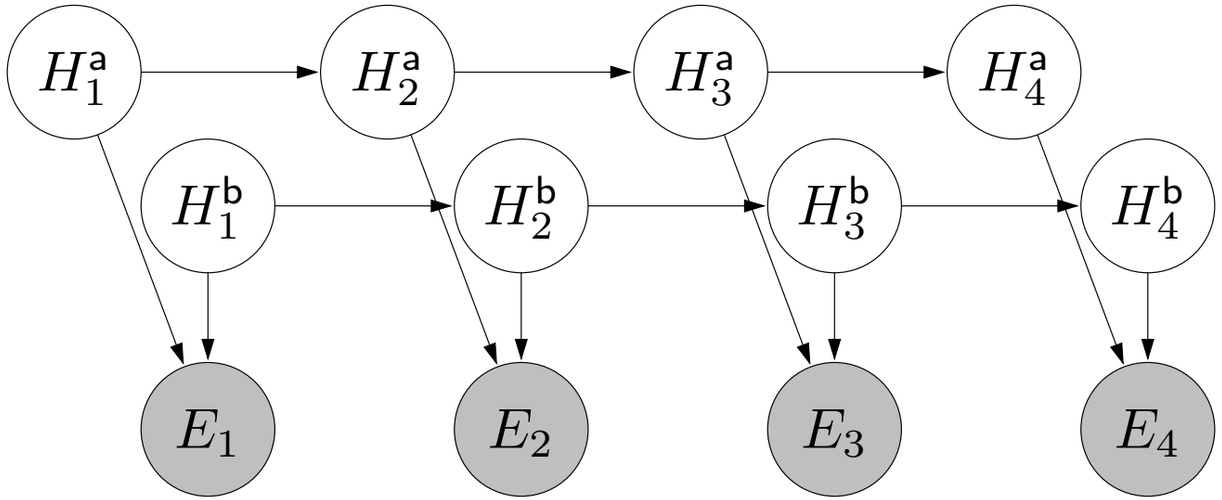
Probabilistic program: factorial HMM

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

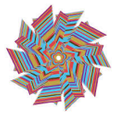
  For each object $o \in \{\mathsf{a}, \mathsf{b}\}$:

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

  Generate sensor reading $E_t \sim p(E_t \mid H_t^{\mathsf{a}}, H_t^{\mathsf{b}})$



Application: document classification

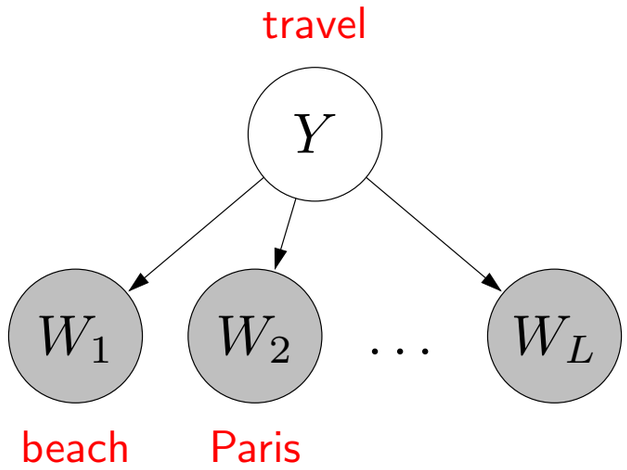


Probabilistic program: naive Bayes

Generate label $Y \sim p(Y)$

For each position $i = 1, \ldots, L$:

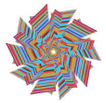
Generate word $W_i \sim p(W_i \mid Y)$



Inference: given a text document, what is it about?

given words, what is label

Application: topic modeling



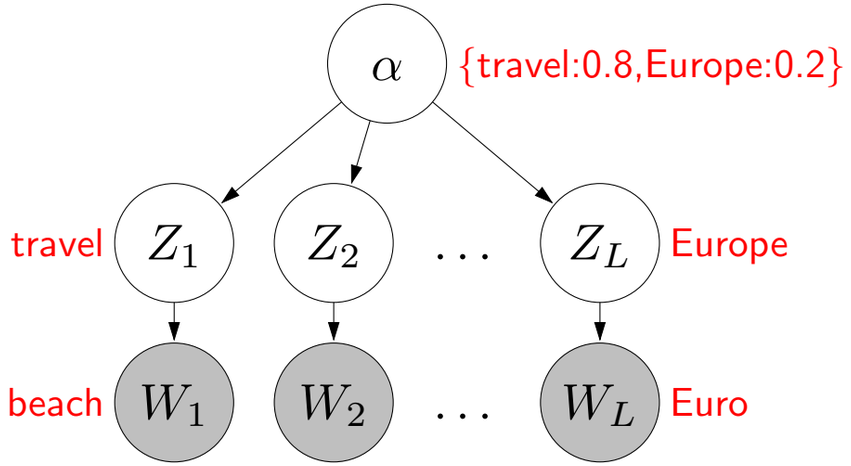
Probabilistic program: latent Dirichlet allocation

Generate a distribution over topics $\alpha \in \mathbb{R}^K$

For each position $i = 1, \ldots, L$:

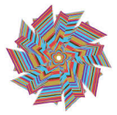
Generate a topic $Z_i \sim p(Z_i \mid \alpha)$

Generate a word $W_i \sim p(W_i \mid Z_i)$



Inference: given a text document, what topics is it about?

Application: medical diagnosis



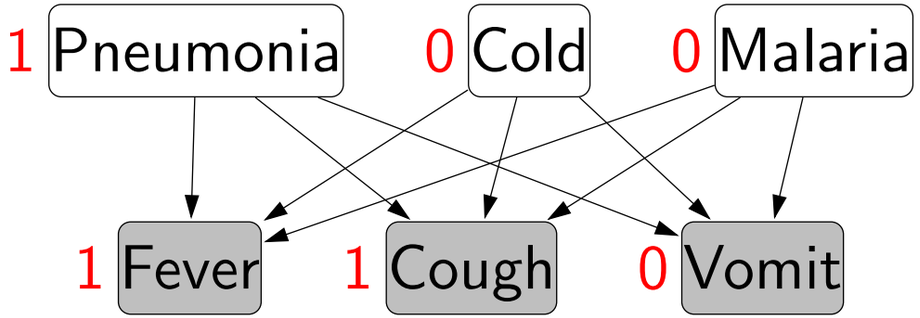
Probabilistic program: diseases and symptoms

For each disease $i = 1, \ldots, m$:

Generate activity of disease $D_i \sim p(D_i)$

For each symptom $j = 1, \ldots, n$:

Generate activity of symptom $S_j \sim p(S_j \mid D_{1:m})$



Inference: If a patient has some symptoms, what diseases do they have?

Application: social network analysis

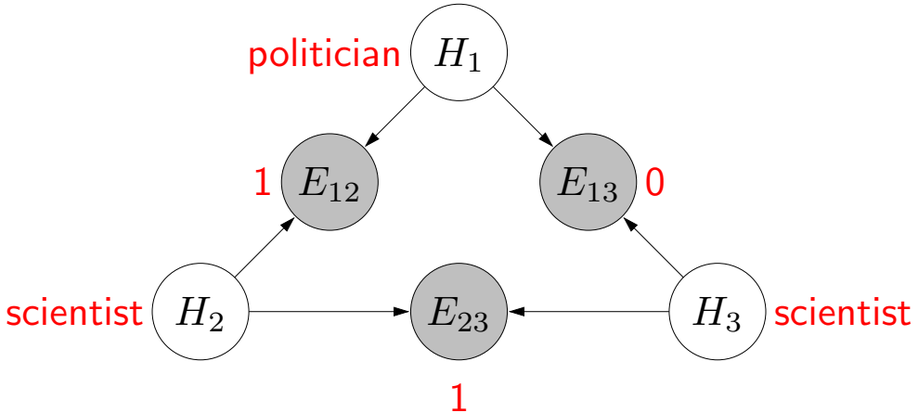
Probabilistic program: stochastic block model

For each person $i = 1, \ldots, n$:

Generate person type $H_i \sim p(H_i)$

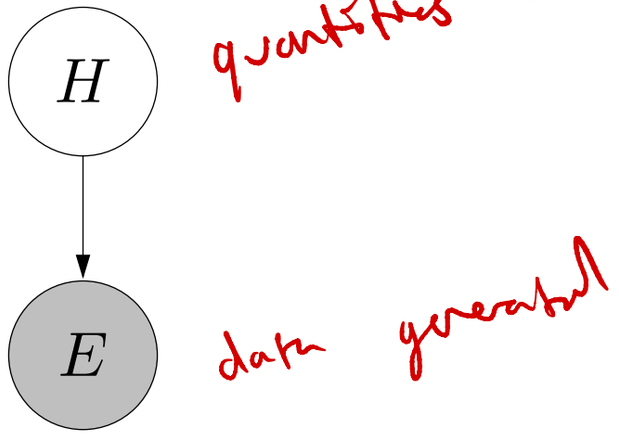
For each pair of people $i \neq j$:

Generate connectedness $E_{ij} \sim p(E_{ij} \mid H_i, H_j)$



Inference: Given a social network graph, what types of people are there?

Summary



- Probabilistic program specifies a Bayesian network
- Many different types of models
- Common paradigm: come up with stories of how the quantities of interest (output) generate the data (input)
- Opposite of how we normally do classification!