Introduction to Artificial Intelligence: Principles and Techniques

Lecture 21: Bayesian networks---Coping with uncertainty

November 17, 2025

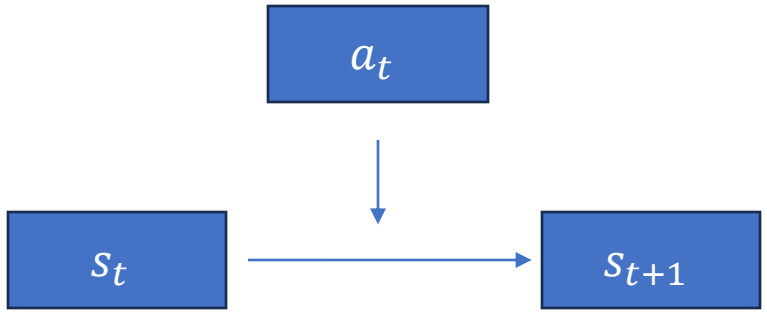
Lecture Plan

- Bayesian networks
  - **Overview and definitions**
  - Probabilistic inference
  - Learning Bayesian networks from data

What is a Bayesian network?

- Bayesian network: a directed probabilistic network that represents a joint distribution using a directed acyclic graph and conditional probability distributions

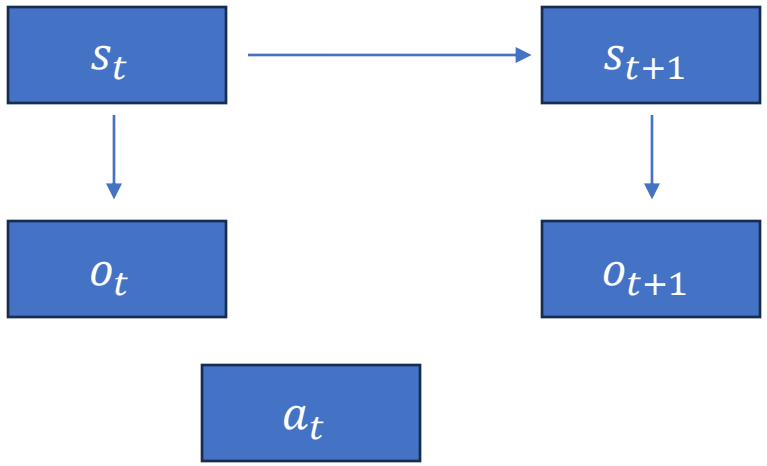
- Example (Markov decision process)



  - Markov transition model: $\Pr(s_{t+1}|s_t, a_t)$

Partially-observed MDP (POMDP)

- A partially-observed MDP has
  - Hidden state s_t
  - Action a_t
  - Observation o_t



  - Belief update

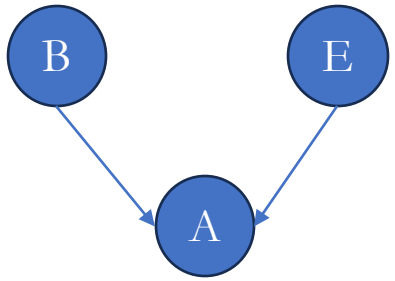
$$b_{t+1}(s') = \eta \Pr(o_{t+1}|s') \sum_{s} \Pr(s'|, b_t) \cdot b_t(s)$$

A toy example for factor analysis

- Problem: earthquakes, burglaries, and alarms
  - **Earthquakes** and **burglaries** are independent events (happening with probability ϵ)
  - Either event will cause the **alarm** to go off
- Suppose you get an **alarm**
  - Does hearing that there is an **earthquake** increase, decrease, or keep the probability of a **burglary**?
- Joint distribution: $\Pr(E, B, A)$
- Questions:
  - $\Pr(B = 1 \mid A = 1)$?
  - $\Pr(B = 1 \mid A = 1, E = 1)$?

Apply Bayesian networks to analyze alarms

- Dependency diagram and table of all possible outcomes
  - Either $B = 1$ or $E = 1$ triggers $A = 1$



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

- Joint distribution:

$$\Pr(B = b, E = e, A = a)$$
$$= \Pr(B = b)\Pr(E = e)\Pr(A = a|B = b, E = e)$$

Calculating the probabilities

- We can calculate the joint probabilities
- Recall that each event happens w. p. ϵ

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

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

Formal definition of Bayesian networks

- 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:

$$\Pr(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} p\left(X_i = x_i \mid X_{parents(i)} = x_{parents(i)} \right)$$

Applications of Bayesian networks

- Natural language processing:
  - **Part-of-speech tagging**: assigning each word in a sentence its **syntactic category**
  - **Parsing**: figuring out the syntactic tree structure of a sentence
  - **Topic modeling**: an unsupervised text mining task that takes a corpus of documents and discovers abstract topics within that corpus (latent Dirichlet allocation)
  - **Grammatical error correction**: encode grammatical rules to correct the grammar within text

- Robotics
  - **Decision-making under uncertainty**
  - **Markov decision processes**

Review: probability

- Random variables: sunshine $S \in \{0,1\}$, rain $R \in \{0,1\}$
- Joint distribution (look at historical records)

$$\mathbb{P}(S,R) = \begin{array}{|cc|c|} \hline s & r & \mathbb{P}(S=s,R=r) \\ \hline 0 & 0 & 0.20 \\ 0 & 1 & 0.08 \\ 1 & 0 & 0.70 \\ 1 & 1 & 0.02 \\ \hline \end{array}$$

- Marginal distribution (sum over all possible R)

$$\mathbb{P}(S) = \begin{array}{|c|c|} \hline s & \mathbb{P}(S=s) \\ \hline 0 & 0.28 \\ 1 & 0.72 \\ \hline \end{array}$$

Review: probability

- Conditional probability (normalize by $\Pr(R)$)

$$\mathbb{P}(S \mid R = 1) = \begin{array}{|cc|} \hline s & \mathbb{P}(S = s \mid R = 1) \\ \hline 0 & 0.8 \\ 1 & 0.2 \\ \hline \end{array}$$

- Additional variables: traffic T
  - Joint distribution: $\Pr(S, R, T)$
  - Marginal distribution $\Pr(T)$
  - Conditional distribution: $\Pr(R|T = 1, S = 0)$ and $\Pr(R|T = 0, S = 0)$

Lecture plan

- Bayesian networks
  - Overview and definitions
  - **Probabilistic inference**
  - Learning Bayesian networks from data

Problem setup

- Input:
  - Bayesian network: $P(X_1, \ldots, X_n)$
  - Evidence: $E = e$ where $E \subseteq X$ is subset of variables
  - Query: $Q \subseteq X$ is subset of variables

- Output:
  - $\Pr(Q \mid E = e)$ or $\Pr(Q = q \mid E = e)$ for all values q

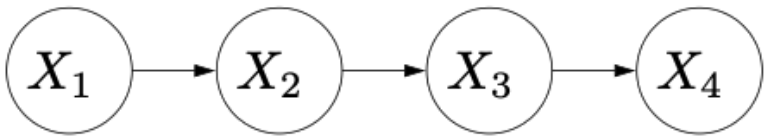
Probabilistic programs

- There is another way of writing down Bayesian networks as a **probabilistic program**, which is a probabilistic program
  - Executing this program will assign values to a collection of random variables $X_1, \ldots, X_n$
- Example (probabilistic program for alarm)
  - $B \sim \text{Bernoulli}(\epsilon)$
  - $E \sim \text{Bernoulli}(\epsilon)$
  - $A = B \vee E$

Example: sentence modeling

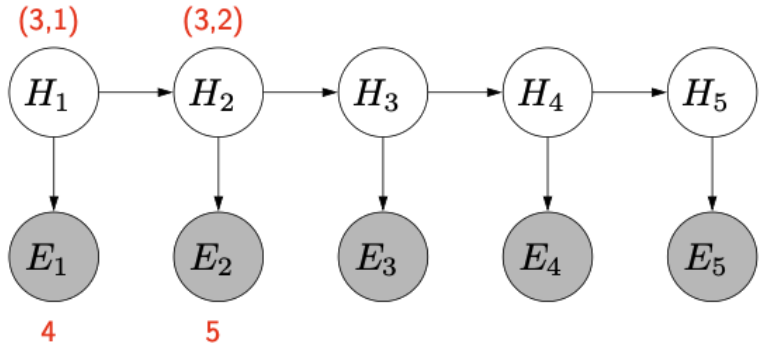
- Score sentences for spam detection, speech recognition, or machine translation

- Underlying probabilistic program: a Markov model for generating a sentence
  - For each position $i = 1,2, \ldots, n$:
    - Measure next-word generation probability $X_i \sim p(X_i \mid X_{i-1})$



Example: object trajectory tracking

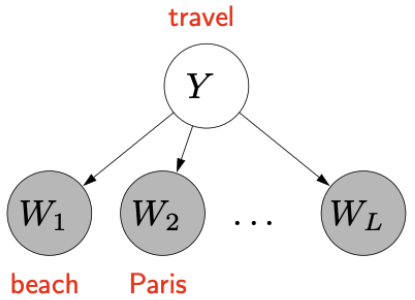
- Dependency diagram: hidden Markov model (HMM)
  - For each time step $t = 1, \ldots, T$:
    - Generate object location $H_t \sim p(H_t \mid H_{t-1})$
    - Generate sensor reading $E_t \sim p(E_t \mid H_t)$
    - After object location



  - Speech recognition: H_t would be the words and E_t would be the raw acoustic signal or wave

Example: sentiment recognition

- Probabilistic program: naïve Bayes model
  - Generate label $Y \sim p(Y)$
  - For each position $i = 1, \ldots, L$:
    - Conditional word generation probability $W_i \sim p(W_i \mid Y)$



- Inference: given a sentence, what sentiment is it?
  - One advantage of using naïve Bayes for classification is that training is extremely easy and fast and just requires counting as opposed to training a neural network

Summary of probabilistic programming

- Probabilistic programs specify a Bayesian network
  - Many different types of models
  - Common paradigm: define how the various random variables of interest (output) generate the observations (input)
  - This type of modeling is different from how we usually do supervised learning

Overall idea: Gibbs sampling

- Recall: given a Bayesian network, want output $\Pr(X \mid E = e)$ or $\Pr(X = q \mid E = e)$ for all values q
- The exact inference is often expensive
- Gibbs sampling gives an approximate way to estimate these conditional probabilities by generating many samples from the posterior distribution
- The key idea is to treat all non-evidence variables as unknowns that we repeatedly resample
  - **Markov blanket**: given a node, the parent nodes and the children of this node

Gibbs sampling

- Step 1: initialization
  - For each non-evidence variable Z, assign Z a random value from its domain
- Step 2: iterative sampling
  - For each Gibbs iteration:
    - Loop over variables in Z (the non-evidence variables)
    - Resample each variable from its conditional distribution given its Markov blanket
- Formally, for each non-evidence variable X_i:

$$Pr(X_i \mid MB(X_i)) \propto Pr(X_i \mid Parents(X_i)) \prod_{Y \in Children(X_i)} Pr(Y \mid Parents(Y))$$

Example: Wet grass

- Graph:
  - Rain → Wet Grass ← Sprinkler
- Evidence variables:
  - Evidence: W
  - W = true
- Non-evidence variables:
  - Non-evidence: R, S
- Goal:
  - Compute $Pr(R = T \mid W = T)$

Example: Wet grass

- Rain:
  - $Pr(R = true) = 0.2, Pr(R = false) = 0.8$
- Sprinkler | Rain

R	$P(S = true \mid R)$	$P(S = false \mid R)$
True	0.01	0.99
False	0.40	0.60

- Wet grass | Rain, Sprinkler

R	S	$P(W = true \mid R, S)$	$P(W = false \mid R, S)$
True	True	0.99	0.01
True	False	0.80	0.20
False	True	0.90	0.10
False	False	0.00	1.00

Exact inference

- First compute the joint probability

$$P(R = T, W = T) = \sum_{s} P(R = T, S = s, W = T)$$

For $s = T$:

$$Pr(R = T, S = T, W = T) = 0.2 \cdot 0.01 \cdot 0.99 = 0.00198$$

For $s = F$:

$$Pr(R = T, S = F, W = T) = 0.2 \cdot 0.99 \cdot 0.80 = 0.1584$$

So

$$Pr(R = T, W = T) = 0.00198 + 0.1584 = 0.16038$$

Exact inference

- Now compute

$$P(W = T) = \sum_r \sum_s P(r, s, W = T)$$

We already have the two terms with $R = T$. For $R = F$:

$$Pr(R = F, S = T, W = T) = 0.8 \cdot 0.4 \cdot 0.9 = 0.288$$
$$Pr(R = F, S = F, W = T) = 0.8 \cdot 0.6 \cdot 0.0 = 0$$

So

$$Pr(W = T) = 0.16038 + 0.288 = 0.44838$$

- Finally,

$$Pr(\, R = T \mid W = T \,) = \frac{Pr(R = T, W = T)}{Pr(W = T)} = \frac{0.16038}{0.44838} \approx 0.3577$$

Gibbs sampling steps

- Each iteration alternates between:
  - Resample R from $Pr(\, R \mid S, W \,)$
  - Resample S from $Pr(\, S \mid R, W \,)$
- Record the value of R, S each iteration
- Iteration 1
  - Resample R
    - Case $R = T$

$$Pr(R = T, S = T, W = T) = 0.2 \cdot 0.01 \cdot 0.99 = 0.00198$$

    - Case $R = F$

$$Pr(R = F, S = T, W = T) = 0.8 \cdot 0.4 \cdot 0.9 = 0.288$$

  - Normalize:

$$P(\, R = T \mid S, W \,) = \frac{0.00198}{0.00198 + 0.288} \approx 0.0068$$

$$P(\, R = F \mid S, W \,) \approx 0.9932$$

Gibbs sampling steps

- Iteration 1
  - Resample S
  - Unnormalized:
    - $S = T: 0.4 \cdot 0.9 = 0.36$
    - $S = F: 0.6 \cdot 0.0 = 0$
  - Normalize:

$$Pr(S = T) = 1, Pr(S = F) = 0$$

Gibbs sampling steps

- Iterations 2-5
  - The same conditional distribution repeats:

$$P(R = F \mid S, W) \approx 0.9932, P(S = T) = 1$$

  - Thus, for iterations 1-5:

$$R = [F, F, F, F, F]$$

  - Posterior estimate:

$$\hat{P}(R = T \mid W = T) = \frac{0}{5} = 0$$

This is biased because the starting point $S = T$ strongly explains $W = T$

Gibbs sampling steps

- Restart with better initialization
  - New initial state:

$$R = T, S = F, W = T$$

Run 10 iterations (exact computation omitted---due to previous slides)

Iter	1	2	3	4	5	6	7	8	9	10
R	T	T	F	F	F	T	F	F	T	F
S	F	F	T	T	T	F	T	T	F	T

Recorded R values:

$$[T, T, F, F, F, T, F, F, T, F]$$

Counts:

- $R = T: 4$
- $R = F: 6$

$$\hat{P}(R = T \mid W = T) = \frac{4}{10} = 0.4$$

Gibbs sampling steps

- Final result:
  - Gibbs sampling gives:

$$Pr(R = T \mid W = T) \approx 0.40\ (empirical)$$

  - Exact value:

$$0.3577$$

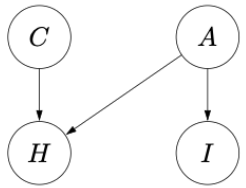
- This example shows that Gibbs sampling can approximate the posterior using only local Markov blanket computations

Lecture Plan

- Bayesian networks
  - Overview and definitions
  - Probabilistic inference
  - **Learning Bayesian networks from data**

Estimate Bayesian network parameters

- Inference assumes that the local conditional distributions are known. But where do all the local conditional distributions come from?
  - These local conditional distributions are specified in the Bayesian network
  - Wet grass example



c	$p(c)$
1	?
0	?

a	$p(a)$
1	?
0	?

c	a	h	$p(h \mid c, a)$
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

a	i	$p(i \mid a)$
0	0	?
0	1	?
1	0	?
1	1	?

Estimate from data

- As with any learning algorithm, we start with the data
- We will focus on the fully-supervised setting, where each data point is a complete assignment to all the variables in the Bayesian network

- Training data: D_{train} (an example is an assignment to X)
- Parameters: θ (local conditional probabilities)
- Output: estimate $\hat{\theta}$

Example: one variable

- Setup:
  - One variable R representing the rating of a movie $\{1,2,3,4,5\}$

$$\Pr(R = r) = p(r)$$

- Parameters:

$$\theta = (p(1), p(2), p(3), p(4), p(5))$$

- Training data:

$$D_{train} = \{1,3,4,4,4,4,4,5,5,5\}$$

Example: one variable

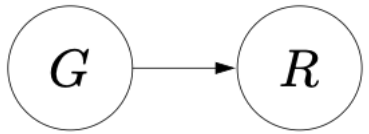
- Intuition: $p(r) \propto$ number of occurrences of r in $D_{train} = \{1,3,4,4,4,4,4,5,5,5\}$

θ:

r	count(r)	$p(r)$
1	1	0.1
2	0	0.0
3	1	0.1
4	5	0.5
5	3	0.3

Example: two variables

- Variables:
  - Genre $G \in \{drama, comedy\}$
  - Rating $R \in \{1,2,3,4,5\}$



$$\Pr(G = g, R = r) = p_G(g)p_R(r \mid g)$$

- $D_{train} = \{(d, 4), (d, 4), (d, 5), (c, 1), (c, 5)\}$
- Parameters: $\theta = (p_G, p_R)$

Example: two variables

- Estimate each local conditional distribution (p_G and p_R) separately

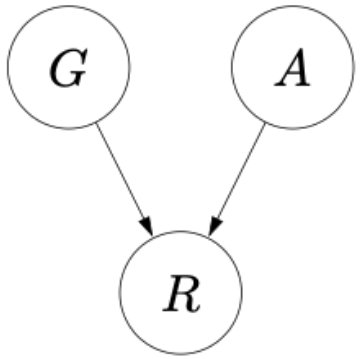
θ:

g	$\text{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

g	r	$\text{count}_R(g, r)$	$p_R(r \mid g)$
d	4	2	2/3
d	5	1	1/3
c	1	1	1/2
c	5	1	1/2

Example: two-level dependency diagram

- Variables:
  - $G \in \{drama, comedy\}$ (genre)
  - $A \in \{0,1\}$ (award)
  - $R \in \{1,2,3,4,5\}$ (rating)



$$\Pr(G = g, A = a, R = r) = p_G(g) \cdot p_A(a) \cdot p_R(r \mid g, a)$$

Example: two-level dependency diagram

$$D_{train} = \{(d, 0, 3), (d, 1, 5), (d, 0, 1), (c, 0, 5), (c, 1, 4)\}$$

- Parameters: $\theta = (p_G, p_A, p_R)$

θ:

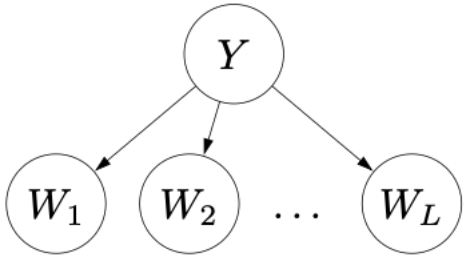
g	$\text{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

a	$\text{count}_A(a)$	$p_A(a)$
0	3	3/5
1	2	2/5

g	a	r	$\text{count}_R(g, a, r)$	$p_R(r \mid g, a)$
d	0	1	1	1/2
d	0	3	1	1/2
d	1	5	1	1
c	0	5	1	1
c	1	4	1	1

Parameter sharing in Naïve Bayes

- In some cases, the local conditional distributions of different variables can share the same parameters
- Variables:
  - Genre $Y \in \{comedy, drama\}$
  - Movie review (sequence of words): $W_1, \ldots, W_L$
  - Parameters: $\theta = (p_{genre}, p_{word})$

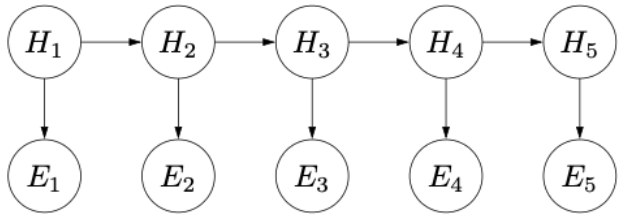


$$\mathbb{P}(Y = y, W_1 = w_1, \ldots, W_L = w_L) = p_{\text{genre}}(y) \prod_{j=1}^{L} p_{\text{word}}(w_j \mid y)$$

- Impact: more reliable estimates, less expressive model

Parameter sharing in hidden Markov models

- Variables:
  - $H_1, \ldots, H_n$ (e.g., actual positions)
  - $E_1, \ldots, E_n$ (e.g., sensor readings)



$$\mathbb{P}(H = h, E = e) = p_{\text{start}}(h_1) \prod_{i=2}^{n} p_{\text{trans}}(h_i \mid h_{i-1}) \prod_{i=1}^{n} p_{\text{emit}}(e_i \mid h_i)$$

- Parameters: $\theta = (p_{start}, p_{trans}, p_{emit})$
- D_{train} is a set of full assignments to (H, E)
- The HMM has K^2 transition parameters and KD emission parameters
  - These parameters are shared in the HMM
  - As a result, the number of parameters does not scale with length n

General case

- **Bayesian network:** variables $X_1, \ldots, X_n$
- **Parameters:** collection of distributions $\theta = \{p_d : d \in D\}$ (e.g., $D = \{start, trans, emit\}$)

- Each variable X_i is generated from distribution p_{d_i}:

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} p_{d_i}\left(x_i \mid x_{parents(i)} \right)$$

  - Parameter sharing: d_i could be same for multiple i

Expectation maximization (EM)

- **EM**: generalization of the K-means algorithm. Cluster centroids = parameters θ; Cluster assignments = hidden variables H
- **Variables**: H is hidden, $E = e$ is observed

EM algorithm

- Initialize θ randomly
- Repeat until convergence
  - E-step:
    - Compute $q(h) = P(H = h \mid E = e; \theta)$ for each h (probabilistic inference)
    - Create fully-observed weighted examples: (h, e) with weight $q(h)$
  - M-step:
    - Maximum likelihood (count and normalize) on weighted examples to get θ (the marginal and conditional probabilities)