PAI. Tutorial 3

Jonas Rothfuss

Jonas Rothfuss

ETH Zürich

October 11, 2019

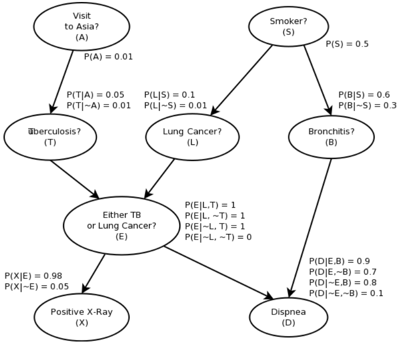
Probability

Today:

- Bayesian Networks
- d-Separation
- Inference in BN: Variable Elimination

Bayesian Network

- Structured representation of a joint probability distribution $P(X_1, \dots, X_n)$
- Favorable for locally structured (sparse) systems



Bayesian Network

Bayesian Network $(\mathcal{G}, \{P_1, ..., P_n\})$

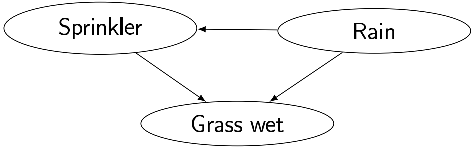
- **Directed acyclical graph (DAG)** $\mathcal{G}$, wherein each node/vertex corresponds to a random variable X_i
- **Conditional probability tables** $P_i = P(X_i \mid \text{Parents}(X_i))$

Joint Probability

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid \text{Parents}(X_i))$$

Bayesian Network: Example

	Sprinkler	
Rain	T	F
F	0.4	0.6
T	0.01	0.99



rain

?

Sprinkler	
T	F
0.2	0.8

		Grass wet	
Sprinkler	rain	T	F
F	F	0.4	0.6
F	T	0.01	0.99
T	F	0.01	0.99
T	T	0.01	0.99

Conditional Independence

Conditional Independence

$$X_1 \perp X_2 \mid X_3 \quad \Leftrightarrow \quad P(X_1, X_2 | X_3) = P(X_1 | X_3) P(X_2 | X_3)$$

Example:

$$\begin{tikzcd} X_1 \arrow[r] & X_2 \arrow[r] & X_3 \end{tikzcd}$$

$$X_1 \perp X_3 \mid X_2$$

$$X_1 \not\perp X_3$$

d-separation

Active Trail (informally: path along which information can flow)

Undirected path on $\mathcal{G}$ is called **active trail** for observed variables $\mathbf{O} \subset \{X_1, ..., X_n\}$, if for all consecutive triples (X, Y, Z) on the path:

- $X \rightarrow Y \rightarrow Z$ and Y unobserved
- $X \leftarrow Y \leftarrow Z$ and Y unobserved
- $X \leftarrow Y \rightarrow Z$ and Y unobserved
- $X \rightarrow Y \leftarrow Z$ and Y or any decendant(Y) is observed

d-separation

If there exists **no active trail** under observations $\mathbf{O}$ between X_1 and X_2, then X_1 and X_2 are called **d-separated** by $\mathbf{O}$. We write

$$\text{d-sep}(X_1; X_2 | \mathbf{O})$$

Conditional Independence & d-separation

d-sep implies cond. independence

$$\text{d-sep}(X_1; X_2 | \mathbf{O}) \quad \Rightarrow \quad X_1 \perp X_2 \mid \mathbf{O}$$

but not the other way around!

$$X_1 \perp X_2 \mid \mathbf{O} \quad \not\Rightarrow \quad \text{d-sep}(X_1; X_2 | \mathbf{O})$$

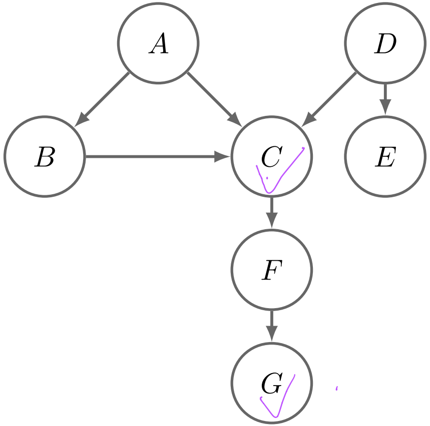
Linear time algorithm for d-separation

Algorithm: d-sep(X ; $\cdot$ $|\mathbf{O}$)

Input: Start node of interest X, observed variables $\mathbf{O}$
Output: All nodes reachable via active trails: $\{Y | Y \text{ not d-sep}(X; Y|\mathbf{O})\}$

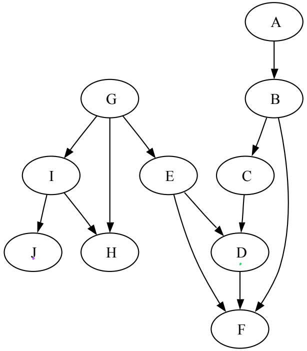
1. Mark nodes in $\mathbf{O}$ and their ancestors ($\Rightarrow$ to check $X \to Y \leftarrow Z$ structures)
2. Do breadth-first search starting from X and check if path is blocked $\Rightarrow$ check the 4 triplet cases

Exercise: d-sep algorithm



- d-sep(A ; · | {})?
- d-sep(E ; · | {G})? X

Homework Problem 1: d-separation



1. $A \perp F$ ✗
2. $A \perp G$ ✓
3. $B \perp I \mid F$ ✗
4. $D \perp J \mid G, H$ ✓
5. $I \perp B \mid H$ ✓
6. $J \perp D$ ✗
7. $I \perp C \mid H, F$ ✗

Homework Problem 3: d-sep algorithm

Given: Code skeleton for BNs
Task: Implement step 1 & 2 of d-sep algorithm

Algorithm: d-sep(X ; $\cdot$ $|\mathbf{O}$)

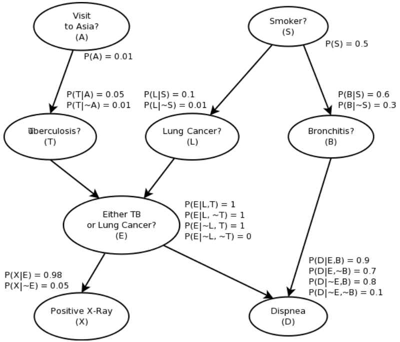
1. Mark nodes in **O** and their ancestors ($\Rightarrow$ to check $X \leftrightarrow Y \leftrightarrow Z$ structures)
2. Do breadth-first search starting from X and check if path is blocked $\Rightarrow$ check the 4 triplet cases
   - Node to expand: Y
   - Cache how you reached Y:
     - case 1: $\rightarrow Y$
     - case 2: $Y \leftarrow$
   - Check how you reach adjacent node: Z
     - case a: $\rightarrow Z$
     - case b: $Z \leftarrow$

Inference in Bayes Nets

Maximum a posteriori (MAP) query

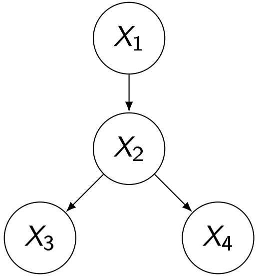
$$\arg\max_{(t,l,b)} P(T = t, l = l, b = b | \neg a, d)$$

How to efficiently compute
$P(T, L, B | A, D)$?



Inference in Bayes Nets

$$P(X_1|X_3 = x_3) = \frac{P(X_1, X_3)}{P(X_3)}$$



$$\begin{aligned} P(X_1, X_3) &= \sum_{X_2} \sum_{X_4} P(X_1, X_2, X_3, X_4) \\ &= \sum_{X_2} \sum_{X_4} P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_2) \\ &= P(X_1) \sum_{X_2} P(X_2|X_1)P(X_3|X_2) \underbrace{\sum_{X_4} P(X_4|X_2)}_{=1} \end{aligned}$$

Inference in Bayes Nets - Variable Elimination

Variable Elimination Algorithm

Input: BN, query variable X, observed values $\mathbf{e}$ for variables $\mathbf{E}$
Output: $P(X|\mathbf{E} = \mathbf{e})$

1. Choose ordering of variables: $X_1, ..., X_n$
2. For $i = 1$ to n: create initial factors $\mathbf{F} = \{f_i = P(X_i|\text{Parents}(X_i))\}$
3. For $i = 1$ to n, if $X_i \notin \{X, \mathbf{E}\}$:
   1. multiply all factors $\{f_{j_1}, ..., f_{j_m}\}$ that include X_i, and
   2. marginalize out X_i
      $\Rightarrow$ new factor $g = \sum_{X_i}(f_{j_1} \odot f_{j_2} \odot \cdots \odot f_{j_m})$
   3. $F \leftarrow F \cup \{g\} \setminus \{f_{j_1}, ..., f_{j_m}\}$
4. Normalize remaining factor, constituting $P(X, \mathbf{e})$, to get $P(X|\mathbf{e})$

Inference in Bayes Nets - Variable Elimination

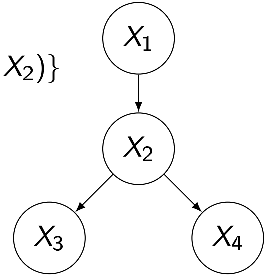
1. multiply all factors $\{f_{j_1}, ..., f_{j_m}\}$ that include X_i, and
2. marginalize out X_i
   $\Rightarrow$ new factor $g = \sum_{X_i}(f_{j_1} \odot f_{j_2} \odot \cdots \odot f_{j_m})$

$f(X_1, ..., X_k)$: k-dimensional tensor

```python
import numpy as np


f1 = np.array([1.0, 0.0]) # f_1(X_1)
f2 = np.array([[0.2, 0.8], [0.3, 0.8]])  #f_2(X_1, X2)

# element-wise product over 1st dimension
g = np.multiply(f1[:, None], f2)

# sum over dimension 1
g = np.sum(g, axis=0)

print(g) # [0.2 0.8]
```

Inference in Bayes Nets - Variable Elimination

Query: $P(X_1 | X_4 = x_4)$
Variable Ordering: X_4, X_3, X_2, X_1



$$\mathbf{F}_0 = \{P(X_1), P(X_2|X_1), P(X_3|X_2), P(X_4 = x_4|X_2)\}$$
$$= \{f_1(X_1), f_2(X_1, X_2), f_3(X_2, X_3), f_4(X_2)\}$$

1. Eliminate X_3:

$$\{f_1(X_1), f_2(X_1, X_2), \underbrace{f_3(X_2, X_3)}_{g_1(X_2) = (1,1)^\top}, f_4(X_2)\}$$

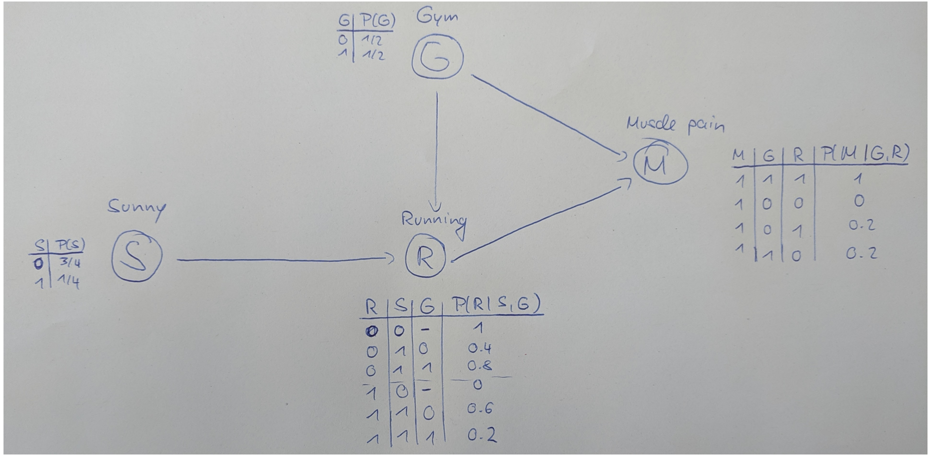
2. Eliminate X_2:

$$\{f_1(X_1), \underbrace{f_2(X_1, X_2), g_1(X_2), f_4(X_2)}_{g_2(X_1)}\}$$

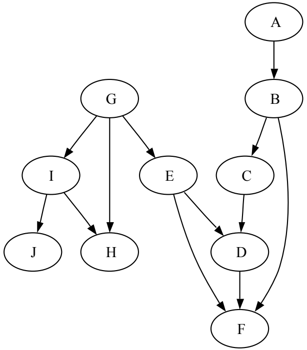
3. Multiply remaining X_1 factors and normalize:

$$P(X_1 | x_4) = \frac{f_1(X_1) \odot g_2(X_1)}{\sum_{X_1} f_1(X_1) \odot g_2(X_1)}$$

Example: Variable Elimination



Homework Problem 2: Variable Elimination



Task: Perform variable elimination with the following variable ordering A, B, C, D, E, F, G, H, I, J

Homework Problem 4: Variable Elimination

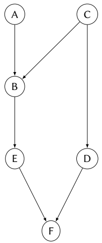


Figure 3: Bayesian network for problem 4.

$$P(A = t) = 0.3 \tag{1}$$
$$P(C = t) = 0.6 \tag{2}$$

Table 1: CPTs for problem 1.

A	C	$P(B = t)$
f	f	0.2
f	t	0.8
t	f	0.3
t	t	0.5

C	$P(D = t)$
f	0.9
t	0.75

B	$P(E = t)$
f	0.2
t	0.4

D	E	$P(F = t)$
f	f	0.95
f	t	1
t	f	0
t	t	0.25

Task: Compute inference query using variable elimination