TDT4171 Artificial Intelligence Methods

Lecture 3 – Bayesian Networks

Norwegian University of Science and Technology

Helge Langseth
Gamle Fysikk 255
helge.langseth@ntnu.no

NTNU

Outline

Summary from last time

- **Probability** is a rigorous formalism for uncertain knowledge
- **Joint probability distribution** specifies probability of every **atomic event**
- Queries can be answered by **summing** over atomic events
- For nontrivial domains, we must find a way to **reduce** the size of the joint distribution, as it grows like $O(d^n)$
- **Independence** and **conditional independence** provide the tools for simplification.
- Calculations can be rather heavy; today we will **start using a SW tool**, which does this for us

Chapter 13: Bayesian networks

Curriculum:

- Syntax of Bayesian networks
- Semantics
- **Modelling**
- Inference (superficially)

Announcements:

- New week, new assignment!

Basic statistics – Bayes' rule

The product rule $P(x \wedge y) = P(x|y)P(y) = P(y|x)P(x)$, can be used to prove Bayes' rule:

$$\boxed{P(x|y) = \frac{P(y|x)P(x)}{P(y)}}$$

Calculation challenge:

- Two factories, **Factory A** and **Factory B**, make light-bulbs. **Factory A** produces 60% of the bulbs, **Factory B** the rest.
- The probability of a light-bulb from **Factory A** being defect is 0.01, from **Factory B** the probability is 0.02.
- What is the probability of a lightbulb being from **Factory A** **given that it is defect**?

Discuss with neighbour for a couple of minutes.

BAYES RULE

FACTORY A: 60% OF BULBS. DEFECT-PROB 0.01

FACTORY B: 40% OF BULBS. DEFECT-PROB 0.02

$$P(\text{Factory}=a \mid \text{defect}) = \frac{P(\text{defect} \mid \text{Factory}=a) \cdot P(\text{Factory}=a)}{P(\text{defect})}$$

$$P(\text{defect}) = P(\text{defect} \wedge \text{Factory}=a) + P(\text{defect} \wedge \text{Factory}=b)$$

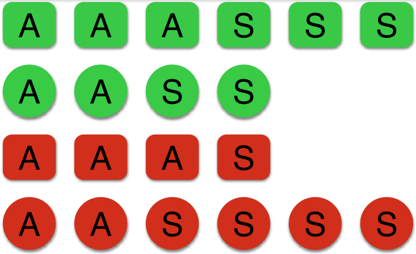
$$= 0.6 \cdot 0.01 + 0.4 \cdot 0.02 = 0.014$$

$$P(\text{Factory}=a \mid \text{defect}) = \frac{0.01 \cdot 0.6}{0.014} = \underline{0.429}$$

Important concept: Independence

- **Independence is an extremely powerful, albeit quite rare property when making models.**
- If X is independent of Y (often written as $X \perp\!\!\!\perp Y$) then $\mathbf{P}(X|Y) = \mathbf{P}(X)$ and — equivalently — $\mathbf{P}(Y|X) = \mathbf{P}(Y)$.
- Independence leads to **more compact representation** and **simplified inference**:
  - $\mathbf{P}(X, Y) = \mathbf{P}(X \mid Y)\mathbf{P}(Y) = \mathbf{P}(X)\mathbf{P}(Y)$.
- Independence statements sanctioned by either of the two:
  - Domain knowledge, e.g., $\texttt{Cavity} \perp\!\!\!\perp \texttt{WinRBK}$
  - Quantifications, e.g., $\mathbf{P}(\texttt{Cavity} \mid \texttt{WinRBK}) = \mathbf{P}(\texttt{Cavity})$

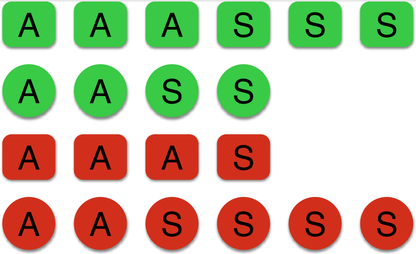
Independence – Example



1. Are Shape and Color independent?
2. Are Shape and Letter independent?
3. Are Letter and Color independent?

Discuss with your neighbour for a couple of minutes.

Independence – Example

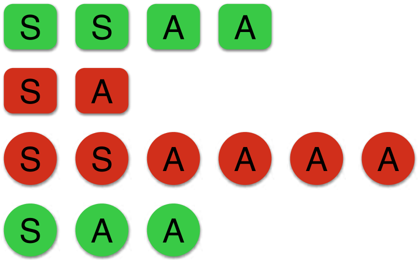


1. Are $\texttt{Shape}$ and $\texttt{Color}$ independent?
   No! E.g., $0.6 = P(\texttt{circle} \mid \texttt{red}) \neq P(\texttt{circle}) = 0.5$.
2. Are $\texttt{Shape}$ and $\texttt{Letter}$ independent?
   No! E.g., $0.4 = P(\texttt{a} \mid \texttt{circle}) \neq P(\texttt{a}) = 0.5$.
3. Are $\texttt{Letter}$ and $\texttt{Color}$ independent?
   YES! $\mathbf{P}(\texttt{Letter} \mid \texttt{Color}) = \mathbf{P}(\texttt{Letter}) = [0.5\ 0.5]$.

Important concept: Conditional independence

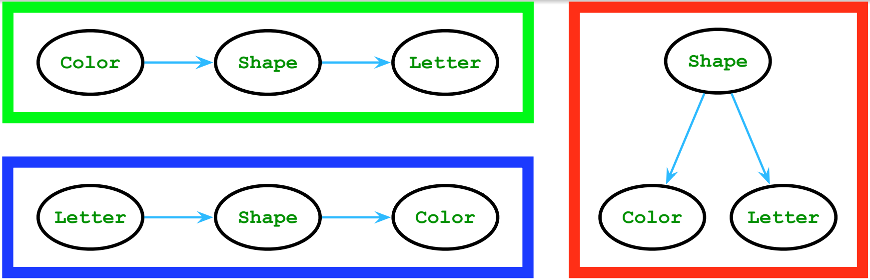
- **Independence extremely powerful, but also extremely rare in applications** $\Rightarrow$ **Conditional independence is our most basic and robust form of knowledge about uncertain environments.**
- If X is independent of Y given Z (often written as $X \perp\!\!\!\perp Y \mid Z$) then $\mathbf{P}(X|Y,Z) = \mathbf{P}(X \mid Z)$ and $\mathbf{P}(Y|X,Z) = \mathbf{P}(Y \mid Z)$.
- Conditional independence leads to **more compact representation** and **simplified inference**.
  - $\mathbf{P}(X,Y,Z) = \mathbf{P}(X \mid Y,Z) \cdot \mathbf{P}(Y,Z) = \mathbf{P}(X \mid Z) \cdot \mathbf{P}(Y,Z)$.
- Conditional independence statements sanctioned by either of the two:
  - Domain knowledge, e.g., $\mathtt{Catch} \perp\!\!\!\perp \mathtt{Toothache} \mid \mathtt{Cavity}$
  - Quantifications, e.g., $\mathbf{P}(\mathtt{Toothache} \mid \mathtt{Catch}, \mathtt{Cavity}) = \mathbf{P}(\mathtt{Toothache} \mid \mathtt{Cavity})$
- Conditional independence sounds a bit involved, but really isn't! **Think of it as "irrelevance of new information".**

Conditional independence – Example



- **All variables are dependent:**
  - `Color` is not independent of `Shape`.
  - `Letter` is not independent of `Shape`.
  - `Letter` is not independent of `Color`.
- **But `Letter` is cond. independent of `Color` given `Shape`.**
  $\mathbf{P}(\texttt{Letter} \mid \texttt{Shape}, \texttt{Color}) = \mathbf{P}(\texttt{Letter} \mid \texttt{Shape})$

Conditional independence and Causality



Three "causal stories" can generate the data:

1. **Nature** first determines `Color`, thereafter chooses `Shape`. Based on the value of `Shape` she determines `Letter`.
2. `Letter` first, then `Letter` → `Shape` and `Shape` → `Color`.
3. **Nature** first determines `Shape`, thereafter chooses `Color` and `Letter` independently, but based on the value of `Shape`.

Only with domain knowledge (e.g., what does `Letter` represent?) can we choose between the three alternative stories.

Conditional independence and Causality (cont'd)

1. Looking for independence and conditional independence in a large data set with many attributes is **difficult**.
2. ... but we need to find them in order to **reduce the number** of necessary probabilities down to a reasonable size.
3. Via **background knowledge** about the domain, we can see the raw data as more than just meaningless vectors of attribute values.
4. This will lead to good **hypotheses** about possible independences and causal independences.
5. These can be easily **checked** against the raw data.
6. Structuring the "causal stories" turns out to be **helpful**.

Bayesian networks

A Bayesian Network is a simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions.

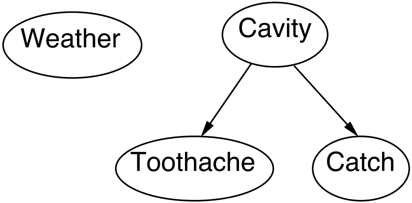
Syntax:

- a set of nodes, one per variable
- a directed, acyclic graph (link $\approx$ "directly influences")
- a conditional distribution for each node given its parents: $\mathbf{P}(X_i|\text{Parents}(X_i))$

Today, all conditional distribution represented as a conditional probability **table** (CPT) giving the distribution over X_i for each combination of parent values (all nodes are discrete).

Example – The Dentist's Domain

Topology of network encodes conditional independence assertions:



- `Weather` is **independent** of the other variables as it does not influence anything, and is not influenced by anything.
- `Toothache` and `Catch` are **conditionally independent** given `Cavity` because any variable is **conditionally independent of all its non-descendants given its parents**.

Example – Hypothetical Covid Domain (in 2020)

Vaccination hinders covid. In 2020, vaccines were limited, so people with pre-existing conditions were prioritized. Smokers were under-represented among those infected. Smoking is more common among male than female. Infected people typically cough and have a fever. **Interesting question:** Is a person infected?

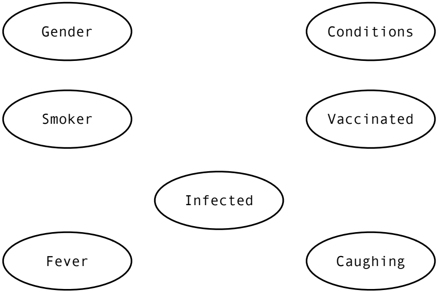
To make a model we must decide ...

- Which are the **random variables**?
- What is the resulting BN **structure**?

Discuss with your neighbour for a couple of minutes.

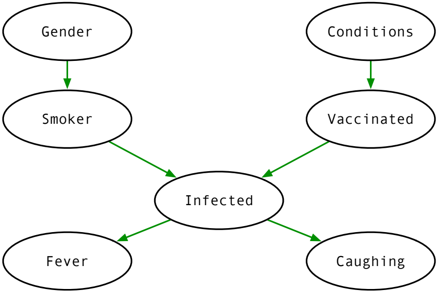
Example – Hypothetical Covid Domain (in 2020)

Vaccination hinders covid. In 2020, vaccines were limited, so people with pre-existing conditions were prioritized. Smokers were under-represented among those infected. Smoking is more common among male than female. Infected people typically cough and have a fever. **Interesting question:** Is a person infected?



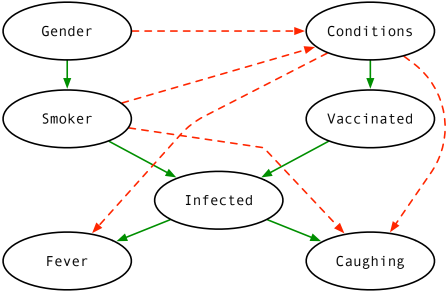
Example – Hypothetical Covid Domain (in 2020)

Vaccination hinders covid. In 2020, vaccines were limited, so people with pre-existing conditions were prioritized. Smokers were under-represented among those infected. Smoking is more common among male than female. Infected people typically cough and have a fever. **Interesting question:** Is a person infected?



Example – Hypothetical Covid Domain (in 2020)

Vaccination hinders covid. In 2020, vaccines were limited, so people with pre-existing conditions were prioritized. Smokers were under-represented among those infected. Smoking is more common among male than female. Infected people typically cough and have a fever. **Interesting question:** Is a person infected?



Example – The Burglary Domain

I'm at work, neighbour John has called (to say my alarm is ringing?), but neighbour Mary hasn't called. Sometimes it's set off by minor earthquakes. **Interesting question:** Is there a burglar?

To make a model we must decide ...

- Which are the **random variables**?
- How is the **casual structure** of the story?
- What is the resulting BN **structure**?

Discuss with your neighbour for a couple of minutes.

Example – The Burglary Domain

I'm at work, neighbour John has called (to say my alarm is ringing?), but neighbour Mary hasn't called. Sometimes it's set off by minor earthquakes. **Interesting question:** Is there a burglar?

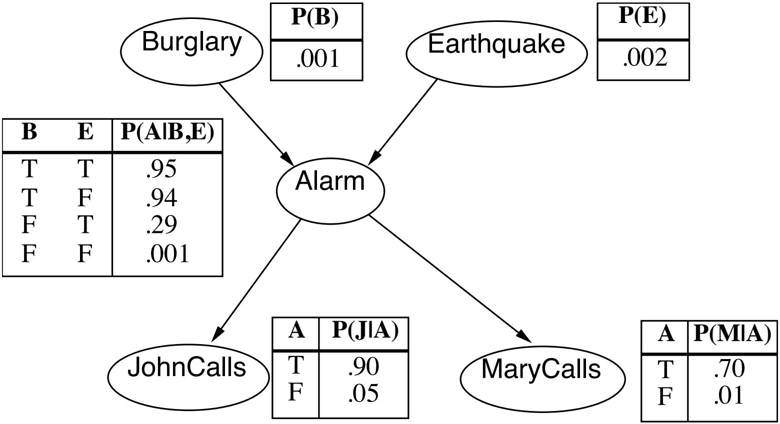
Variables:

- `Burglar`, `Earthquake`
- `Alarm`
- `JohnCalls`, `MaryCalls`

Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

Example cont'd.



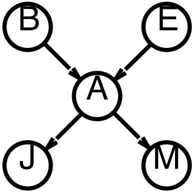
Example contd.

Demo of the **GeNIe** system: `burglary.xdsl`

System availability:

https://download.bayesfusion.com/files.html?category=Academia

Compactness



A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values

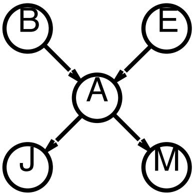
Each row requires one number p for $X_i = \mathtt{true}$ (the number for $X_i = \mathtt{false}$ is just $1 - p$)

If each variable has no more than k parents, the complete network requires $O(n \cdot 2^k)$ numbers

I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution! For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Global semantics

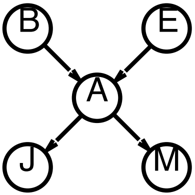
Global semantics defines the full joint distribution as the product of the local conditional distributions:



$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(x_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

Global semantics



Global semantics defines the full joint distribution as the product of the local conditional distributions:

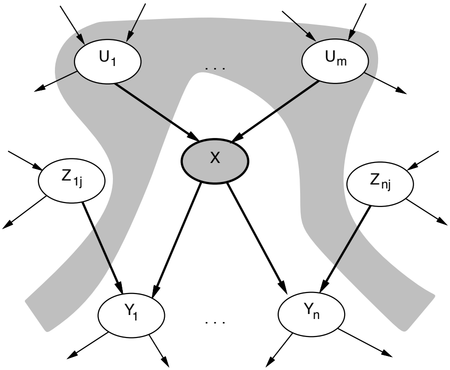
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathsf{parents}(x_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.0006 \end{aligned}$$

Local semantics

Local semantics: each node is conditionally independent of its nondescendants given its parents



Theorem: Local semantics $\Leftrightarrow$ global semantics

Local semantics (Example)

4 models in GeNIe:

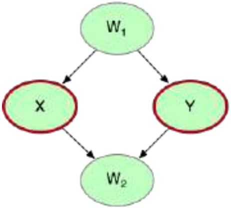
- Defect Lightbulbs: `Factory.xdsl`
- Cause & Effect: `CauseEffect.xdsl`
- Dividing structure (fork): `Divider.xdsl`
- V-structure (collider): `Collider.xdsl`

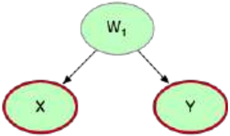
Checking for conditional independence

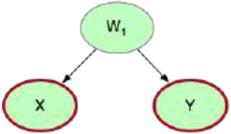
To check if a set of variables $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ do the following:

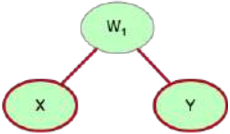
1. Create the **ancestral graph**: $\{\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}$ and their ancestral nodes
2. **Moralize** the graph: Every pair of nodes with a common child must be connected.
3. Drop all directions.
4. **Check:** Does $\boldsymbol{Z}$ block all paths between $\boldsymbol{X}$ and $\boldsymbol{Y}$?
   - If all paths are blocked, then $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$.
   - If there is at least one unblocked path, then $\boldsymbol{X} \not\!\perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$.

Checking for conditional independence – Examples

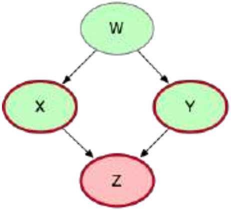


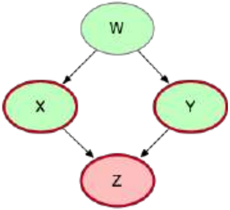


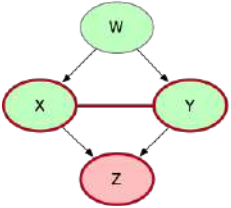


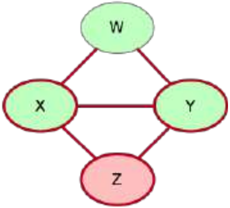


Checking for conditional independence – Examples

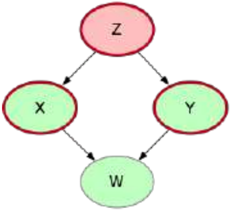


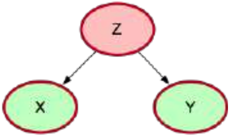


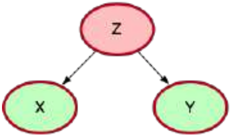


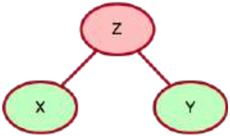


Checking for conditional independence – Examples

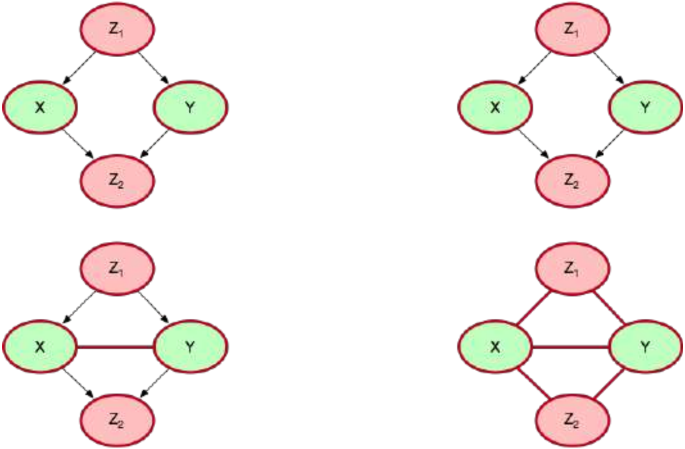






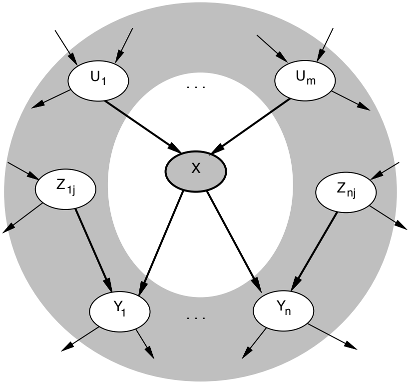


Checking for conditional independence – Examples



Markov blanket

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



Yet another example

Conditional independence in `GeNIe` – Forest_Fire.xdsl

- Are `Storm` and `Thunder` . . .
  - **independent**?
  - **conditionally independent** given `Lightning`?
- Which variables are . . .
  - **independent** of `BusTourGroup`?
  - **conditionally independent** of `BusTourGroup` given `ForestFire`?

Discuss with your neighbour for a couple of minutes.

Constructing Bayesian networks

Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
    1. add X_i to the network
    2. select parents from $X_1, \ldots, X_{i-1}$ such that $\mathbf{P}(X_i|\text{Parents}(X_i)) = \mathbf{P}(X_i|X_1, \ldots, X_{i-1})$

This choice of parents guarantees the global semantics:

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \mathbf{P}(X_i|X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i|\text{Parents}(X_i)) \quad \text{(by construction)}\end{aligned}$$

Example

Suppose we choose the ordering B, E, A, J, M

`Burglary` is the only variable, so cannot have parents. Move on. . .

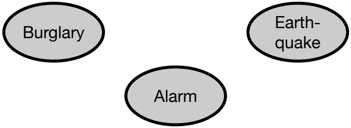
Example

Suppose we choose the ordering B, E, A, J, M

$P(E|B) = P(E)$?

Example

Suppose we choose the ordering B, E, A, J, M

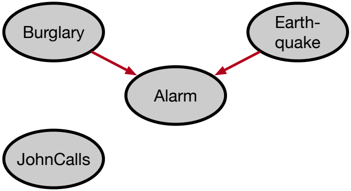


$P(E|B) = P(E)$? Yes

$P(A|B,E) = P(A|B)$? $P(A|B,E) = P(A|E)$?

Example

Suppose we choose the ordering B, E, A, J, M



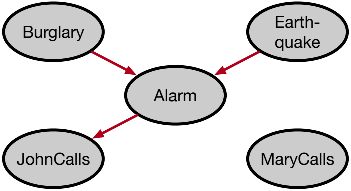
$P(E|B) = P(E)$? Yes

$P(A|B,E) = P(A|B)$? $P(A|B,E) = P(A|E)$? No

$P(J|A,B,E) = P(J|A)$?

Example

Suppose we choose the ordering B, E, A, J, M



$P(E|B) = P(E)$? Yes

$P(A|B,E) = P(A|B)$? $P(A|B,E) = P(A|E)$? No

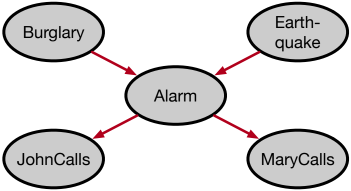
$P(J|A,B,E) = P(J|A)$? Yes

$P(M|A,B,J,E) = P(M)$?

$P(M|A,B,J,E) = P(M|A)$?

Example

Suppose we choose the ordering B, E, A, J, M

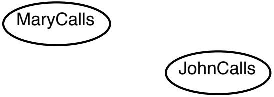


$P(E|B) = P(E)$? Yes
$P(A|B,E) = P(A|B)$? $P(A|B,E) = P(A|E)$? No
$P(J|A,B,E) = P(J|A)$? Yes
$P(M|A,B,J,E) = P(M)$? No
$P(M|A,B,J,E) = P(M|A)$? Yes

Finding the structure was fairly easy in this case, and the result is a "sparse" network with $1+1+4+2+2=10$ required parameters.

Example

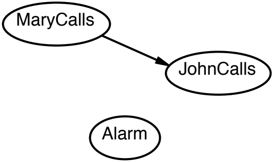
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$?

Example

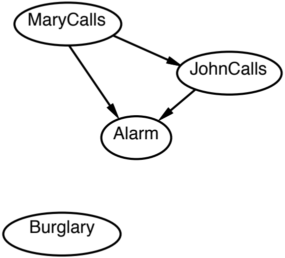
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$?

Example

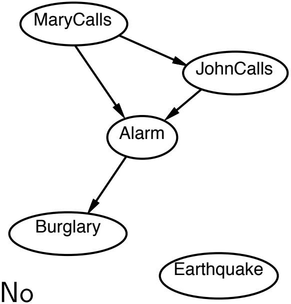
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$?
$P(B|A,J,M) = P(B)$?

Example

Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No

$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No

$P(B|A,J,M) = P(B|A)$? Yes

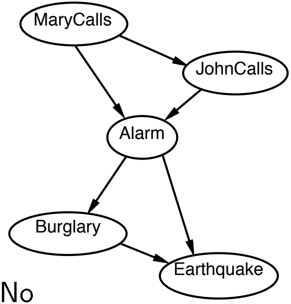
$P(B|A,J,M) = P(B)$? No

$P(E|B,A,J,M) = P(E|A)$?

$P(E|B,A,J,M) = P(E|A,B)$?

Example

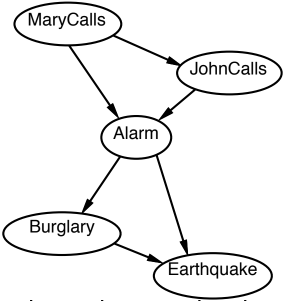
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$? Yes
$P(B|A,J,M) = P(B)$? No
$P(E|B,A,J,M) = P(E|A)$? No
$P(E|B,A,J,M) = P(E|A,B)$? Yes

Example

Suppose we choose the ordering M, J, A, B, E



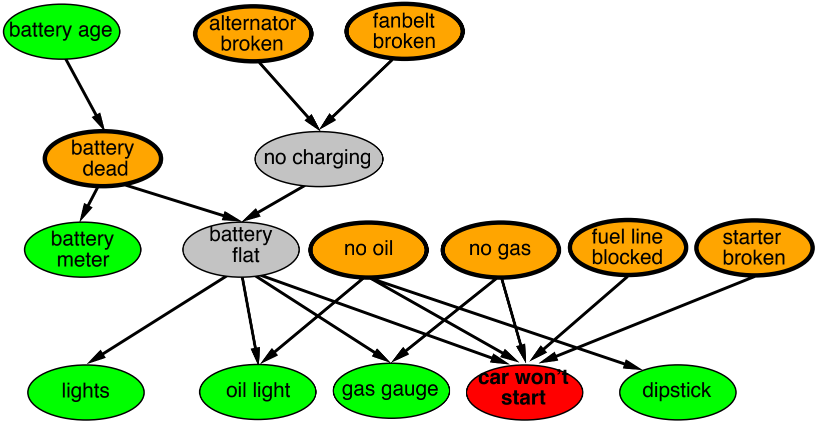
Deciding conditional independence is hard in non-causal directions
– Causal models and conditional independence seem hardwired for humans!

Assessing conditional probabilities is hard in non-causal directions

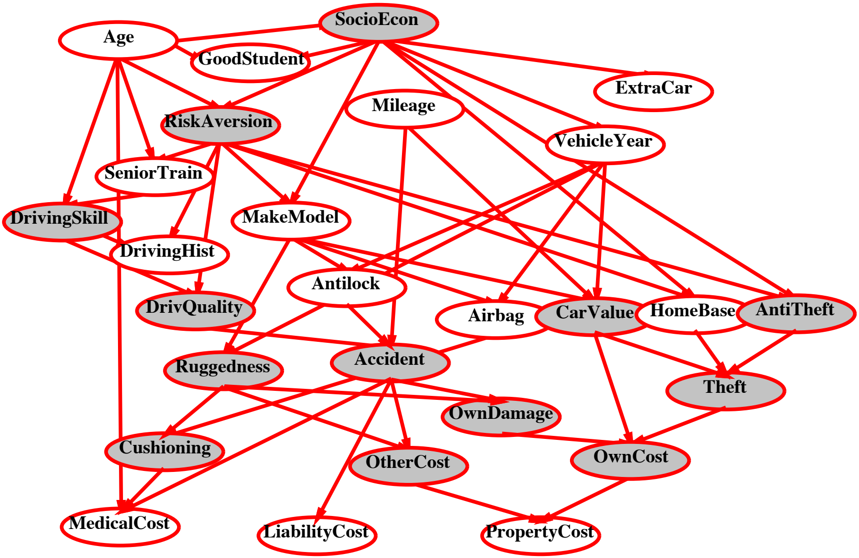
Network is **less compact**: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed.

Example: Car diagnosis

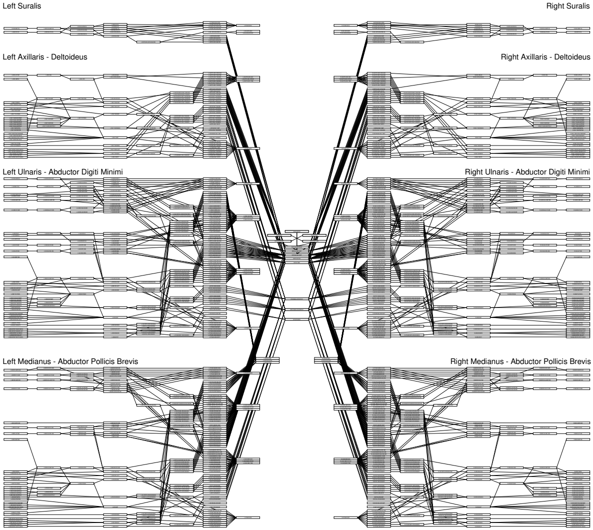
Initial evidence: car won't start
Testable variables (green), **"broken, so fix it" variables** (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car insurance



Example: MUNIN



Phases of the model building process

Step 0 – Decide what to model: Select the boundary for what to include in the model.

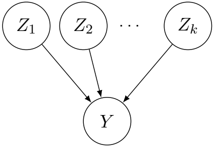
Step 1 – Defining variables: Select the important variables in the domain.

Step 2 – The qualitative part: Define the graphical structure that connects the variables.

Step 3 – The quantitative part: Fix parameters to specify each $P(x_i \mid \mathrm{pa}\,(x_i))$. **This is the difficult part.**

Step 4 – Verification: Verification of the model.

The quantitative part: Defining $P(y|\text{pa}(y))$

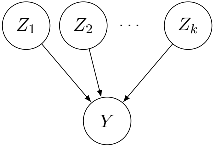


Consider a binary node with k binary parents. The CPT $P(y|z_1, \ldots, z_k)$ contains 2^k parameters.

Naïve approach: 2^k conditional probabilities:

All parameters are required if no other assumptions can be made.

The quantitative part: Defining $P(y|\mathrm{pa}(y))$



Consider a binary node with k binary parents. The CPT $P(y|z_1, \ldots, z_k)$ contains 2^k parameters.

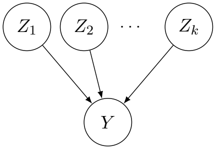
Naïve approach: 2^k conditional probabilities

Deterministic relations: Parameter free:

Y considered a deterministic function of its parents, e.g.,

$$\{Y = \texttt{true}\} \iff \{Z_1 = \texttt{true}\} \vee \{Z_2 = \texttt{true}\} \vee \ldots \vee \{Z_k = \texttt{true}\}.$$

The quantitative part: Defining $P(y|\mathrm{pa}(y))$

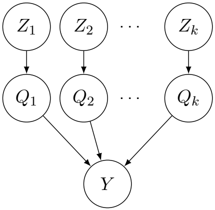


Consider a binary node with k binary parents. The CPT $P(y|z_1, \ldots, z_k)$ contains 2^k parameters.

Naïve approach: 2^k conditional probabilities

Deterministic relations: Parameter free

Noisy OR relation: $k+1$ conditional probabilities:



Independent inhibitors $Q_1, \ldots, Q_k$; Assume $\{Q_1 = \mathtt{true}\} \vee \ldots \vee \{Q_k = \mathtt{true}\} \Rightarrow \{Y = \mathtt{true}\}$.

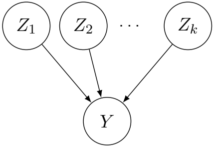
For each Q_i we have

$$\begin{aligned} P(Q_i = \mathtt{true}|Z_i = \mathtt{true}) &= q_i, \\ P(Q_i = \mathtt{true}|Z_i = \mathtt{false}) &= 0. \end{aligned}$$

"Leak probability":

$$P(Y = \mathtt{true}|Q_1 = \ldots = Q_k = \mathtt{false}) = q_0.$$

The quantitative part: Defining $P(y|\text{pa}(y))$



Consider a binary node with k binary parents. The CPT $P(y|z_1,\ldots,z_k)$ contains 2^k parameters.

Naïve approach: 2^k conditional probabilities

Deterministic relations: Parameter free

Noisy OR relation: $k+1$ conditional probabilities

Special structures: From 2 to 2^k conditional probabilities:

Y defined, e.g., by *rules* such as
"$P(Y=\texttt{true}|Z_1=\texttt{true},\ldots,Z_k=\texttt{true})=p_1$, but
$P(Y=\texttt{true}|z_1,\ldots,z_k)=p_2$ for all other configurations $\boldsymbol{z}$".

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$, e.g.,

$P(\texttt{NoGas}|\texttt{Gauge}=\texttt{empty},\texttt{Lights}=\texttt{on},\texttt{Starts}=\texttt{false})$

Conjunctive queries:

$$\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$$

Optimal decisions: decision networks include utility information; probabilistic inference required for $P(\texttt{outcome}|\texttt{action},\texttt{evidence})$

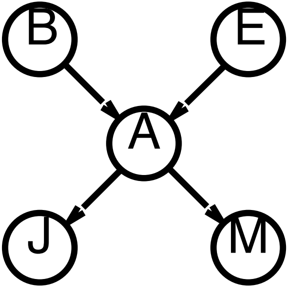
Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

Explanation: why do I need a new starter motor?

Inference tasks – Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation.



Simple query on the burglary network:

$$\begin{aligned}\mathbf{P}(B|j,m) &= \mathbf{P}(B,j,m)/P(j,m)\\ &= \alpha\mathbf{P}(B,j,m)\\ &= \alpha\ \Sigma_e\ \Sigma_a\ \mathbf{P}(B,e,a,j,m)\end{aligned}$$

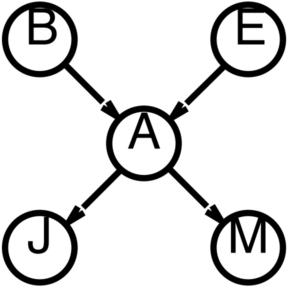
Rewrite full joint entries using product of CPT entries:

$$\mathbf{P}(B|j,m) = \alpha\Sigma_e\Sigma_a\mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a)$$

Recursive depth-first enumeration: $O(n)$ space, $O(n \cdot d^n)$ time

Inference tasks – Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation.



Simple query on the burglary network:

$$\begin{aligned}\mathbf{P}(B|j,m) &= \mathbf{P}(B,j,m)/P(j,m) \\ &= \alpha\mathbf{P}(B,j,m) \\ &= \alpha\ \Sigma_e\ \Sigma_a\ \mathbf{P}(B,e,a,j,m)\end{aligned}$$

Rewrite full joint entries using product of CPT entries:

$$\begin{aligned}\mathbf{P}(B|j,m) &= \alpha\Sigma_e\Sigma_a\mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a) \\ &= \alpha\mathbf{P}(B)\ \Sigma_e P(e)\Sigma_a\ \mathbf{P}(a|B,e)P(j|a)P(m|a)\end{aligned}$$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Enumeration algorithm

function ENUMERATION-ASK(X, **e**, *bn*) **returns** distr. over X
inputs: X, the query variable
e, observed values for variables **E**
bn, a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

$\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
for each value x_i of X **do**
  extend **e** with value x_i for X
  $\mathbf{Q}(x_i) \leftarrow$ ENUM-ALL(VARS[*bn*], **e**)
return NORMALIZE($\mathbf{Q}(X)$)

---

function ENUM-ALL(*vars*, **e**) **returns** a real number
if EMPTY?(*vars*) **then return** 1.0
$Y \leftarrow$ FIRST(*vars*)
if Y has value y in **e** **then**
  return $P(y|\text{Pa}(Y)) \times$ ENUM-ALL(REST(*vars*), **e**)
else return $\sum_y P(y|\text{Pa}(Y)) \times$ ENUM-ALL(REST(*vars*), $\mathbf{e}_y$)
  where $\mathbf{e}_y$ is **e** extended with $Y = y$

Evaluation of $P(b, j, m)$ with Enum-All

Level 1: **Enum-All**(*vars* $= [B, E, A, J, M]$, $\boldsymbol{e} = \{b, j, m\}$):
$\rightarrow P(b)\cdot$ **Enum-All**($[E, A, J, M]$, $\{b, j, m\}$)

Level 2: **Enum-All**(*vars* $= [E, A, J, M]$, $\boldsymbol{e} = \{b, j, m\}$):
$\rightarrow P(e)\cdot$ **Enum-All**($[A, J, M]$, $\{b, j, m, e\}$)
$+ P(\neg e)\cdot$ **Enum-All**($[A, J, M]$, $\{b, j, m, \neg e\}$)

Level 3: **Enum-All**(*vars* $= [A, J, M]$, $\boldsymbol{e} = \{b, j, m, e\}$):
$\rightarrow P(a \mid b, e)\cdot$ **Enum-All**($[J, M]$, $\{b, j, m, e, a\}$)
$+ P(\neg a \mid b, e)\cdot$ **Enum-All**($[J, M]$, $\{b, j, m, e, \neg a\}$)

Level 4: **Enum-All**(*vars*$=[J, M]$, $\boldsymbol{e} = \{b, j, m, e, a\}$):
$\rightarrow P(j \mid a)\cdot$ **Enum-All**($[M]$, $\{b, j, m, e, a\}$)

Level 5: **Enum-All**(*vars*$=[M]$, $\boldsymbol{e} = \{b, j, m, e, a\}$):
$\rightarrow P(m \mid a)\cdot$ **Enum-All**($[\cdot]$, $\{b, j, m, e, a\}$)

Level 6: **Enum-All**(*vars*$=[\cdot]$, $\boldsymbol{e} = \{b, j, m, e, a\}$):
$\rightarrow 1.0$

Level 4: **Enum-All**(*vars*$=[J, M]$, $\boldsymbol{e} = \{b, j, m, e, \neg a\}$):
$\rightarrow P(j \mid \neg a)\cdot$ **Enum-All**($[M]$, $\{b, j, m, e, \neg a\}$)

Level 5: **Enum-All**(*vars*$=[M]$, $\boldsymbol{e} = \{b, j, m, e, \neg a\}$):
$\rightarrow P(m \mid \neg a)\cdot$ **Enum-All**($[\cdot]$, $\{b, j, m, e, \neg a\}$)

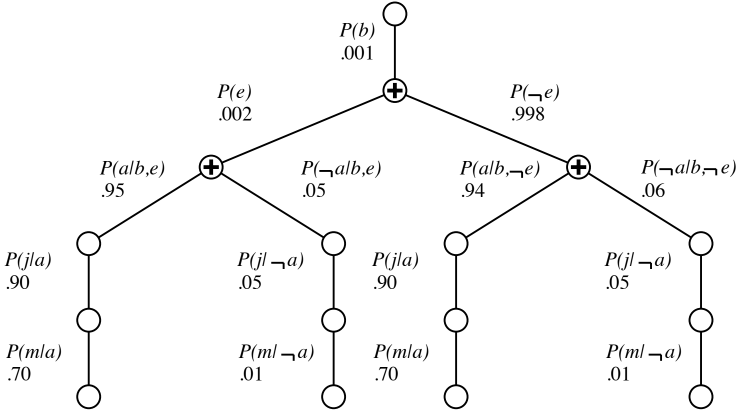
Level 6: **Enum-All**(*vars*$=[\cdot]$, $\boldsymbol{e} = \{b, j, m, e, \neg a\}$):
$\rightarrow 1.0$

Level 3: **Enum-All**(*vars*$=[A, J, M]$, $\boldsymbol{e} = \{b, j, m, \neg e\}$):
$\rightarrow P(a \mid b, \neg e)\cdot$ **Enum-All**($[J, M]$, $\{b, j, m, \neg e, a\}$)
$+ P(\neg a \mid b, \neg e)\cdot$ **Enum-All**($[J, M]$, $\{b, j, m, \neg e, \neg a\}$)

...

Evaluation tree

Enum-All with *vars*$=[B, E, A, J, M]$ and $\boldsymbol{e} = \{b, j, m\}$



Enumeration is still **inefficient**, as we have repeated computation of e.g., $P(j|a) \cdot P(m|a)$ for each value of e.

$\Rightarrow$ **Nice to know that even better methods are available...**

Summary

- **Bayes nets** provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = **compact representation** of joint distribution
- Generally **easy to construct** – also for non-experts
- **Canonical distributions** (e.g., noisy-OR) = compact representation of CPTs
- **Efficient inference** calculations are available (but the good ones are outside the scope of this course)