Artificial Intelligence

15. Probabilistic Reasoning, Part II: Bayesian Networks

Putting the Machinery to Practical Use

Jörg Hoffmann, Daniel Fiser, Daniel Höller, Sophia Saller

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Summer Term 2022

Agenda

Reminder: Our Agenda for This Topic

→ Our treatment of the topic "Probabilistic Reasoning" consists of Chapters 14 and 15.

- **Chapter 14:** All the basic machinery at use in Bayesian networks.

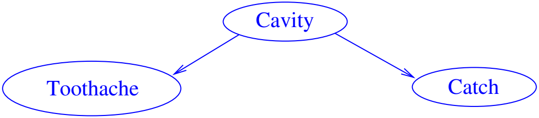
  → Sets up the framework and basic operations.

- **This Chapter:** Bayesian networks: What they are, how to build them, how to use them.

  → The most wide-spread and successful practical framework for probabilistic reasoning.

Reminder: Our Machinery

1. Graph captures variable dependencies: (Variables $X_1, \dots, X_n$)



$\rightarrow$ Given evidence $\mathbf{e}$, want to know $\mathbf{P}(X \mid e)$. Remaining vars: $\mathbf{Y}$.

2. Normalization+Marginalization:

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y} \in \mathbf{Y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

$\rightarrow$ A sum over atomic events!

3. Chain rule: $X_1, \dots, X_n$ consistently with dependency graph.

$$\mathbf{P}(X_1, \dots, X_n) = \mathbf{P}(X_n \mid X_{n-1}, \dots, X_1) * \mathbf{P}(X_{n-1} \mid X_{n-2}, \dots, X_1) * \dots * \mathbf{P}(X_1)$$

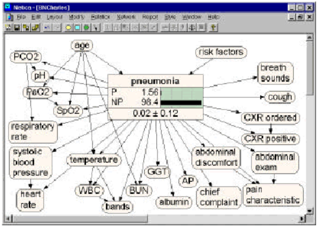
4. Exploit conditional independence: Instead of $\mathbf{P}(X_i \mid X_{i-1}, \dots, X_1)$, we can use $\mathbf{P}(X_i \mid Parents(X_i))$.

$\rightarrow$ Bayesian networks!

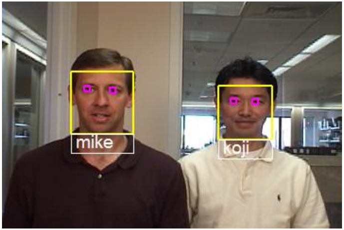
Some Applications

→ A ubiquitous problem: Observe "symptoms", need to infer "causes".

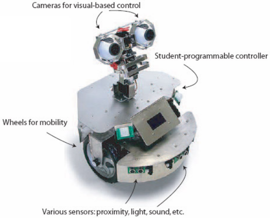
Medical Diagnosis



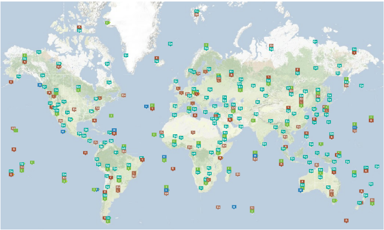
Face Recognition



Self-Localization



Nuclear Test Ban



Our Agenda for This Chapter

- **What is a Bayesian Network?** What is the syntax?

  $\rightarrow$ Tells you what Bayesian networks look like.

- **What is the Meaning of a Bayesian Network?** What is the semantics?

  $\rightarrow$ Makes the intuitive meaning precise.

- **Constructing Bayesian Networks:** How do we design these networks? What effect do our choices have on their size?

  $\rightarrow$ Before you can start doing inference, you need to model your domain.

- **Inference in Bayesian Networks:** How do we use these networks? What is the associated complexity?

  $\rightarrow$ Inference is our primary purpose. We (very) briefly analyze its complexity and how it can be improved.

What is a Bayesian Network? (Short: BN)

"A Bayesian network is a methodology for representing the full joint probability distribution. In some cases, that representation is compact."

"A Bayesian network is an acyclic directed graph whose nodes are random variables X_i and whose edges $X_j \rightarrow X_i$ denote a direct influence of X_j on X_i. Each node X_i is associated with a conditional probability table (CPT), specifying $\mathbf{P}(X_i \mid Parents(X_i))$."

"A Bayesian network is a graphical way to depict conditional independence relations within a set of random variables."

$\rightarrow$ A Bayesian network (BN) represents the structure of a given domain. Probabilistic inference exploits that structure for improved efficiency.

$\rightarrow$ BN inference: Determine the distribution of a query variable X given observed evidence $\mathbf{e}$: $\mathbf{P}(X \mid \mathbf{e})$.

John, Mary, and My Brand-New Alarm

Example

I got very valuable stuff at home. So I bought an alarm. Unfortunately, the alarm just rings at home, doesn't call me on my mobile. I've got two neighbors, Mary and John, who'll call me if they hear the alarm. The problem is that, sometimes, the alarm is caused by an earthquake. Also, John might confuse the alarm with his telephone, and Mary might miss the alarm altogether because she typically listens to loud music.

Question: Given that both John and Mary call me, what is the probability of a burglary?

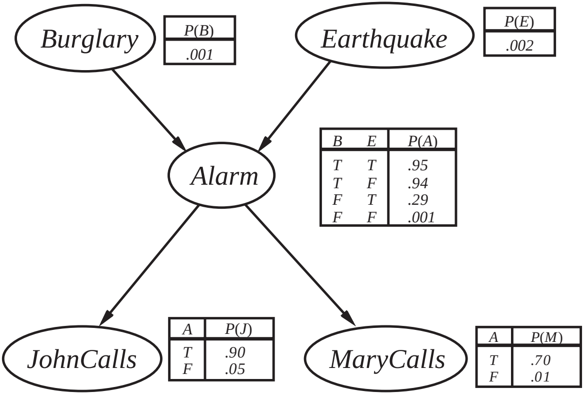
John, Mary, and My Alarm: Designing the BN

Cooking Recipe: (1) Design the random variables $X_1, \dots, X_n$; (2) Identify their dependencies; (3) Insert the conditional probability tables $\mathbf{P}(X_i \mid Parents(X_i))$.

Example: Let's cook!

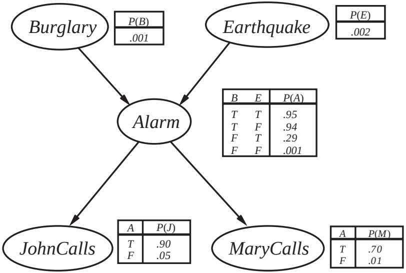
1. Random variables: $Burglary$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$. (All Boolean)
2. Dependencies: Burglaries and earthquakes are independent (this is actually debatable $\rightarrow$ design decision!); the alarm might be activated by either. John and Mary call if and only if they hear the alarm (they don't care about earthquakes).
3. Conditional probability tables: Assess the probabilities, see next slide.

John, Mary, and My Alarm: The BN



Note: In each $\mathbf{P}(X_i \mid Parents(X_i))$, we show only $\mathbf{P}(X_i = true \mid Parents(X_i))$. We don't show $\mathbf{P}(X_i = false \mid Parents(X_i))$ which is $= 1 - \mathbf{P}(X_i = true \mid Parents(X_i))$.

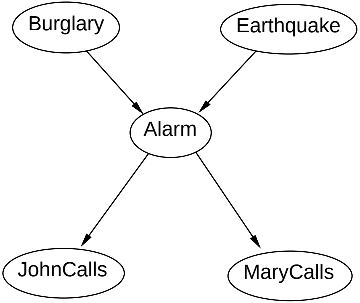
The Syntax of Bayesian Networks



Definition **(Bayesian Network).** *Given random variables* $X_1, \dots, X_n$ *with finite domains* $D_1, \dots, D_n$*, a* *Bayesian network* *is an acyclic directed graph* $BN = (\{X_1, \dots, X_n\}, E)$*. We denote* $Parents(X_i) := \{X_j \mid (X_j, X_i) \in E\}$*. Each* X_i *is associated with a function* $CPT(X_i) : D_i \times (\times_{X_j \in Parents(X_i)} D_j) \mapsto [0, 1]$*.*

[$\rightarrow$ Why "acyclic"? Slide 19 (*) $\mathbf{P}(X_1, \dots, X_n) = \Pi_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$. By (*), acyclic BN suffice to represent any full joint probability distribution. But for cyclic BN, (*) does NOT hold, indeed cyclic BNs may be self-contradictory.]

The Semantics of BNs: Example

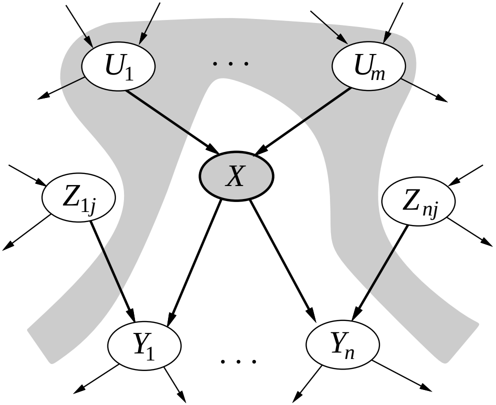


- $Alarm$ depends on $Burglary$ and $Earthquake$.
- $MaryCalls$ only depends on $Alarm$.
  $P(MaryCalls \mid Alarm, Burglary) = P(MaryCalls \mid Alarm)$

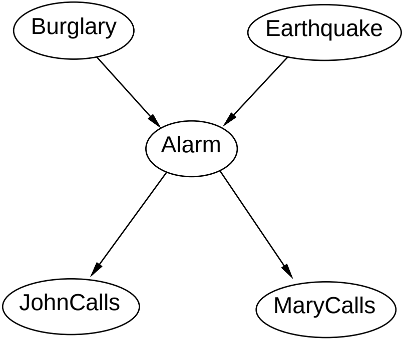
→ Bayesian networks represent sets of independence assumptions.

The Semantics of BNs: General Case

→ Each node X in a BN is conditionally independent of its non-descendants given its parents $Parents(X)$.

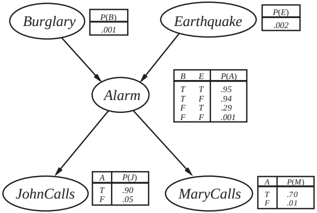


The Semantics of BNs: Example, ctd.



→ Given the value of *Alarm*, *MaryCalls* is independent of?
$\{Burglary, Earthquake, JohnCalls\}$.

The Semantics of BNs: Formal



Definition. *Given a Bayesian network* $BN = (\{X_1, \dots, X_n\}, E)$, *we identify* BN *with the following two assumptions:*

(A) *For* $1 \leq i \leq n$, X_i *is conditionally independent of* $NonDescendants(X_i)$ *given* $Parents(X_i)$, *where* $NonDescendants(X_i) := \{X_j \mid (X_i, X_j) \notin E^*\} \setminus Parents(X_i)$ *with* E^* *denoting the transitive closure of* E.

(B) *For* $1 \leq i \leq n$, *all values* x_i *of* X_i, *and all value combinations* $parents(X_i)$ *of* $Parents(X_i)$, *we have* $P(x_i \mid parents(X_i)) = CPT(x_i, parents(X_i))$.

Recovering the Full Joint Probability Distribution

"A Bayesian network is a methodology for representing the full joint probability distribution."

→ How to recover the full joint probability distribution $\mathbf{P}(X_1, \dots, X_n)$ from $BN = (\{X_1, \dots, X_n\}, E)$?

Chain rule: For any ordering $X_1, \dots, X_n$, we have:

$$\mathbf{P}(X_1, \dots, X_n) = \mathbf{P}(X_n \mid X_{n-1}, \dots, X_1)\mathbf{P}(X_{n-1} \mid X_{n-2}, \dots, X_1) \dots \mathbf{P}(X_1)$$

Choose $X_1, \dots, X_n$ consistent with BN: $X_j \in Parents(X_i) \implies j < i$.

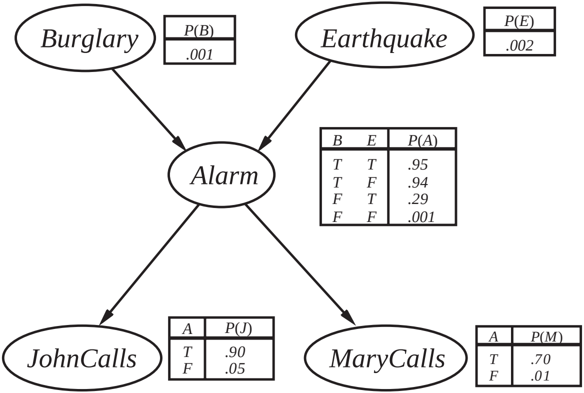
Exploit conditional independence: With BN assumption (A), instead of $\mathbf{P}(X_i \mid X_{i-1} \dots, X_1)$ we can use $\mathbf{P}(X_i \mid Parents(X_i))$:

$$\mathbf{P}(X_1, \dots, X_n) = \Pi_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$$

The distributions $\mathbf{P}(X_i \mid Parents(X_i))$ are given by BN assumption (B).

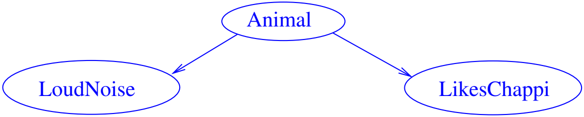
→ Same for atomic events $P(x_1, \dots, x_n)$.

Recovering a Probability for John, Mary, and the Alarm



$$
\begin{aligned}
{\color{red}P(j, m, a, \neg b, \neg e)} &= P(j \mid a)P(m \mid a)P(a \mid \neg b, \neg e)P(\neg b)P(\neg e) \\
&= 0.9 * 0.7 * 0.001 * 0.999 * 0.998 \\
&= 0.00062
\end{aligned}
$$

Questionnaire



Question!

Say BN is the Bayesian network above. Which statements are correct?

(A): $Animal$ is independent of $LikesChappi$.

(B): $LoudNoise$ is independent of $LikesChappi$.

(C): $Animal$ is conditionally independent of $LikesChappi$ given $LoudNoise$.

(D): $LikesChappi$ is conditionally independent of $LoudNoise$ given $Animal$.

$\rightarrow$ (A) No: $likeschappi$ indicates dog. (B) No: Not knowing what animal it is, $likeschappi$ is an indication for dog which indicates $loudnoise$. (C) No: For example, even if we know $loudnoise$, knowing in addition that $likeschappi$ gives us a stronger indication of $Animal = dog$. (D) Yes: $X_i = LikesChappi$ is conditionally independent of $NonDescendants(X_i) = \{LoudNoise\}$ given $Parents(X_i) = \{Animal\}$.

Constructing Bayesian Networks

BN construction algorithm:

1. Initialize $BN := (\{X_1, \dots, X_n\}, E)$ where $E = \emptyset$.
2. Fix any order of the variables, $X_1, \dots, X_n$.
3. **for** $i := 1, \dots, n$ **do**
   a. Choose a minimal set $Parents(X_i) \subseteq \{X_1, \dots, X_{i-1}\}$ so that $\mathbf{P}(X_i \mid X_{i-1} \dots, X_1) = \mathbf{P}(X_i \mid Parents(X_i))$.
   b. For each $X_j \in Parents(X_i)$, insert (X_j, X_i) into E.
   c. Associate X_i with $CPT(X_i)$ corresponding to $\mathbf{P}(X_i \mid Parents(X_i))$.

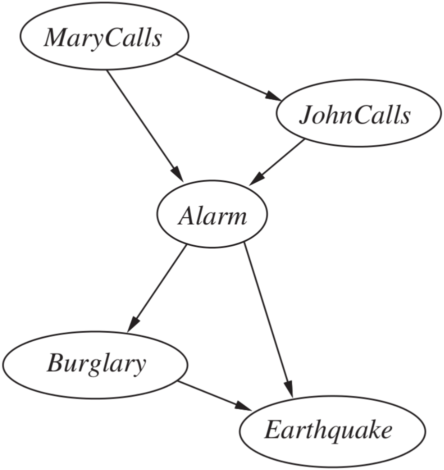
Attention! *Which* variables we need to include into $Parents(X_i)$ depends on what "$\{X_1, \dots, X_{i-1}\}$" is . . . !

$\rightarrow$ The size of the resulting BN depends on the chosen order $X_1, \dots, X_n$.

$\rightarrow$ The size of a Bayesian network is *not* a fixed property of the domain. It depends on the skill of the designer.

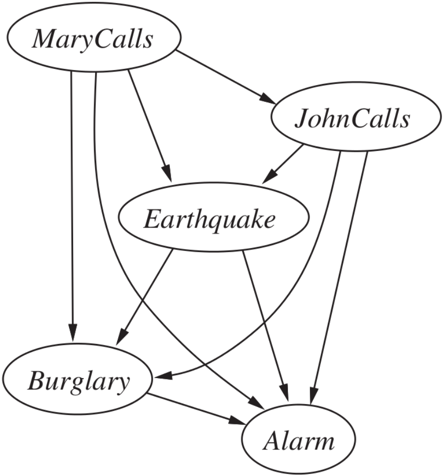
John and Mary Depend on the Variable Order!

Example: $MaryCalls, JohnCalls, Alarm, Burglary, Earthquake$.

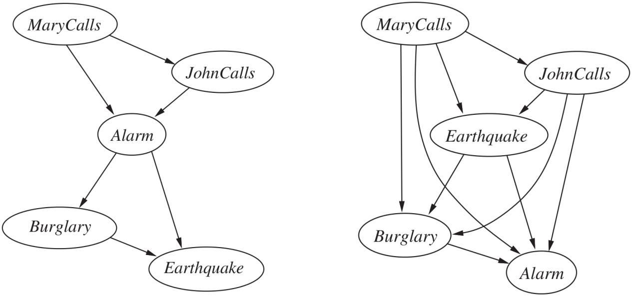


John and Mary Depend on the Variable Order! Ctd.

Example: $MaryCalls, JohnCalls, Earthquake, Burglary, Alarm.$



John and Mary, What Went Wrong?



→ These BNs link from symptoms to causes! $(\mathbf{P}(Cavity \mid Toothache))$

- We fail to identify many conditional independence relations (e.g., get dependencies between conditionally independent symptoms).
- Also recall: Conditional probabilities $\mathbf{P}(Symptom \mid Cause)$ are more robust and often easier to assess than $\mathbf{P}(Cause \mid Symptom)$.

→ We should order causes before symptoms.

The Size of Bayesian Networks

Definition. *Given random variables $X_1, \dots, X_n$ with finite domains $D_1, \dots, D_n$, the size of $BN = (\{X_1, \dots, X_n\}, E)$ is defined as* $size(BN) := \sum_{i=1}^{n} |D_i| * \Pi_{X_j \in Parents(X_i)} |D_j|$. (= The total number of entries in the CPTs.)

→ Smaller BN $\Longrightarrow$ assess less probabilities, more efficient inference.

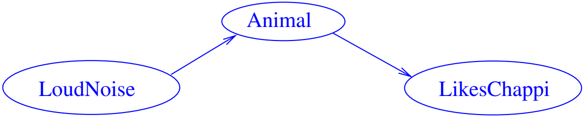
- Explicit full joint probability distribution has size $\Pi_{i=1}^{n} |D_i|$.
- If $|Parents(X_i)| \leq k$ for every X_i, and $D_{\max}$ is the largest variable domain, then $size(BN) \leq n * |D_{\max}|^{k+1}$.
  → For $|D_{\max}| = 2$, $n = 20$, $k = 4$ we have $2^{20} = 1048576$ probabilities, but a Bayesian network of size $\leq 20 * 2^5 = 640 \dots$!
- In the *worst case*, $size(BN) = \sum_{i=1}^{n} \Pi_{j=1}^{i} |D_i|$, namely if *every variable depends on all its predecessors in the chosen order*.

→ BNs are compact if each variable is directly influenced only by few of its predecessor variables.

Questionnaire

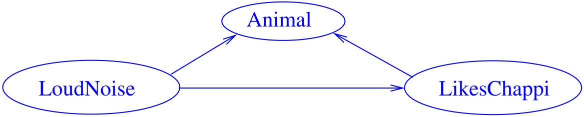
Question!

What is the Bayesian network we get by constructing according to the ordering $X_1 = LoudNoise, X2 = Animal, X3 = LikesChappi$?



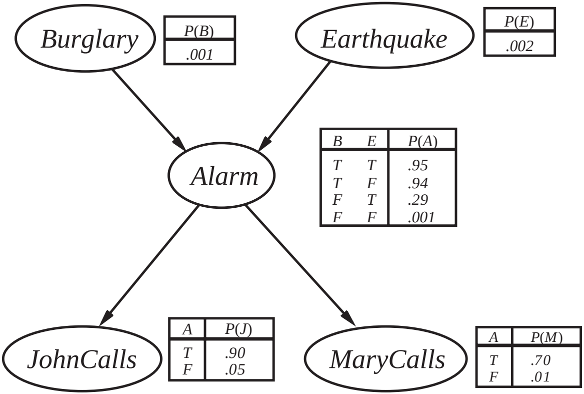
Question!

What is the Bayesian network we get by constructing according to the ordering $X_1 = LoudNoise, X2 = LikesChappi, X3 = Animal$?



Inference for Mary and John

$\rightarrow$ Observe evidence variables and draw conclusions on query variables.



What is $\mathbf{P}(Burglary \mid johncalls)$?

What is $\mathbf{P}(Burglary \mid johncalls, marycalls)$?

Probabilistic Inference Tasks in Bayesian Networks

Definition (Probabilistic Inference Task). *Given random variables $X_1, \dots, X_n$, a probabilistic inference task consists of a set $\mathbf{X} \subseteq \{X_1, \dots, X_n\}$ of query variables, a set $\mathbf{E} \subseteq \{X_1, \dots, X_n\}$ of evidence variables, and an event $\mathbf{e}$ that assigns values to $\mathbf{E}$. We wish to compute the posterior probability distribution $\mathbf{P}(\mathbf{X} \mid \mathbf{e})$.*

Notes:

- $\mathbf{Y} := \{X_1, \dots, X_n\} \setminus (\mathbf{X} \cup \mathbf{E})$ are the hidden variables.
- We assume that a BN for $X_1, \dots, X_n$ is given.
- In the remainder, for simplicity, $\mathbf{X} = \{X\}$ is a singleton.

Example: In $\mathbf{P}(Burglary \mid johncalls, marycalls)$, $X = Burglary$, $\mathbf{e} = johncalls, marycalls$, and $\mathbf{Y} = \{Alarm, EarthQuake\}$.

Inference by Enumeration: The Principle (A Reminder!)

Given evidence $\mathbf{e}$, want to know $\mathbf{P}(X \mid \mathbf{e})$. Hidden variables: $\mathbf{Y}$.

1. **Bayesian network BN captures variable dependencies.**

2. **Normalization+Marginalization.**
$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e})$; if $\mathbf{Y} \neq \emptyset$ then $\mathbf{P}(X \mid \mathbf{e}) = \alpha \sum_{\mathbf{y} \in \mathbf{Y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$
$\rightarrow$ Recover the summed-up probabilities $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ from BN!

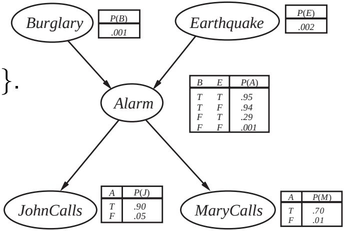
3. **Chain rule.** Order $X_1, \dots, X_n$ consistent with BN.
$\mathbf{P}(X_1, \dots, X_n) = \mathbf{P}(X_n \mid X_{n-1}, \dots, X_1)\mathbf{P}(X_{n-1} \mid X_{n-2}, \dots, X_1) \dots \mathbf{P}(X_1)$

4. **Exploit conditional independence.** Instead of $\mathbf{P}(X_i \mid X_{i-1}, \dots, X_1)$, use $\mathbf{P}(X_i \mid Parents(X_i))$.

$\rightarrow$ Given a Bayesian network BN, probabilistic inference tasks can be solved as sums of products of conditional probabilities from BN.

$\rightarrow$ Sum over all value combinations of hidden variables.

Inference by Enumeration: John and Mary



- Want: $\mathbf{P}(Burglary \mid johncalls, marycalls)$.
  Hidden variables: $\mathbf{Y} = \{Earthquake, Alarm\}$.
- Normalization+Marginalization:
  $\mathbf{P}(B \mid j, m) = \alpha\mathbf{P}(B, j, m)$
  $= \alpha \sum_{v_E} \sum_{v_A} \mathbf{P}(B, j, m, v_E, v_A)$
- Order $X_1 = B, X_2 = E, X_3 = A, X_4 = J, X_5 = M$.
- Chain rule and conditional independence: $\mathbf{P}(B \mid j, m) =$
  $\alpha \sum_{v_E} \sum_{v_A} \mathbf{P}(B)P(v_E)\mathbf{P}(v_A \mid B, v_E)P(j \mid v_A)P(m \mid v_A)$
- Move variables outwards (until we hit the first parent): $\mathbf{P}(B \mid j, m) =$
  $\alpha\mathbf{P}(B) \sum_{v_E} P(v_E) \sum_{v_A} \mathbf{P}(v_A \mid B, v_E)P(j \mid v_A)P(m \mid v_A)$

  $\rightarrow$ The probabilities of the outside-variables multiply the entire "rest of the sum" (compare slides 35 and 36).
- Continuation on next slide ...

Inference by Enumeration: John and Mary, ctd.

Chain rule and conditional independence, ctd.: $\mathbf{P}(B \mid j, m) =$

$$\alpha \mathbf{P}(B) \sum_{v_E} P(v_E) \sum_{v_A} \mathbf{P}(v_A \mid B, v_E) P(j \mid v_A) P(m \mid v_A)$$

$$\alpha \langle P(b)[P(e) \underbrace{(\overbrace{P(a \mid b, e)P(j \mid a)P(m \mid a)}^{a} + \overbrace{P(\neg a \mid b, e)P(j \mid \neg a)P(m \mid \neg a)}^{\neg a})}_{e} +$$

$$P(\neg e) \underbrace{(\overbrace{P(a \mid b, \neg e)P(j \mid a)P(m \mid a)}^{a} + \overbrace{P(\neg a \mid b, \neg e)P(j \mid \neg a)P(m \mid \neg a)}^{\neg a})}_{\neg e}],$$

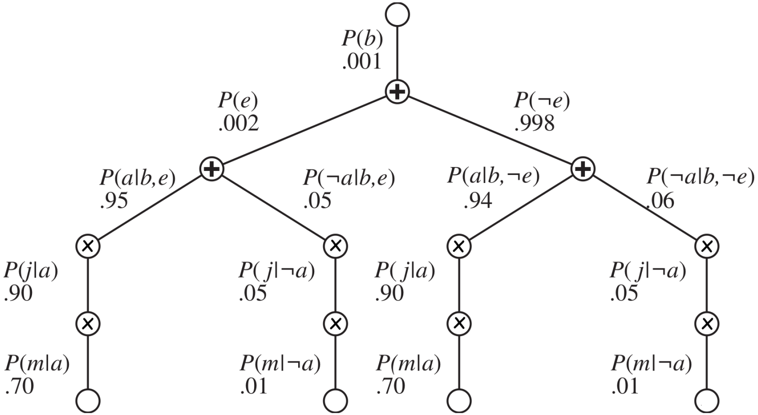
$$P(\neg b)[P(e) \underbrace{(\overbrace{P(a \mid \neg b, e)P(j \mid a)P(m \mid a)}^{a} + \overbrace{P(\neg a \mid \neg b, e)P(j \mid \neg a)P(m \mid \neg a)}^{\neg a})}_{e} +$$

$$P(\neg e) \underbrace{(\overbrace{P(a \mid \neg b, \neg e)P(j \mid a)P(m \mid a)}^{a} + \overbrace{P(\neg a \mid \neg b, \neg e)P(j \mid \neg a)P(m \mid \neg a)}^{\neg a})}_{\neg e}]\rangle$$

$$= \alpha \langle 0.00059224, 0.0014919 \rangle \approx \langle 0.284, 0.716 \rangle$$

$\rightarrow$ This computation can be viewed as a "search tree", see next slide.

The Evaluation of $P(b \mid j, m)$, as a "Search Tree"



→ Inference by enumeration = a tree with "sum nodes" branching over values of hidden variables, and with non-branching "multiplication nodes".

Inference by Enumeration: Pseudo-Code

→ With $bn.\textsc{Vars}$ being a variable ordering consistent with bn:

```
function ENUMERATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayes net with variables {X} ∪ E ∪ Y   /* Y = hidden variables */

   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
      Q(x_i) ← ENUMERATE-ALL(bn.VARS, e_{x_i})
         where e_{x_i} is e extended with X = x_i
   return NORMALIZE(Q(X))
```

```
function ENUMERATE-ALL(vars, e) returns a real number
   if EMPTY?(vars) then return 1.0
   Y ← FIRST(vars)
   if Y has value y in e
      then return P(y | parents(Y)) × ENUMERATE-ALL(REST(vars), e)
      else return ∑_y P(y | parents(Y)) × ENUMERATE-ALL(REST(vars), e_y)
         where e_y is e extended with Y = y
```

Inference by Enumeration: Properties

Inference by Enumeration:

- Evaluates the tree in a depth-first manner.
- Space Complexity: Linear in the number of variables.
- Time Complexity: Exponential in the number of hidden variables, e.g., $O(2^{|\mathbf{Y}|})$ in case these variables are Boolean.
  $\rightarrow$ Can we do better than this?

Bad News: Not in general.

- Probabilistic inference is **#P**-hard.
- **#P** is harder than **NP** (i.e., **NP** $\subseteq$ **#P**).

But: Variable Elimination.

- Improves on inference by enumeration through (A) avoiding repeated computation, and (B) avoiding irrelevant computation.
- In some special cases, variable elimination runs in polynomial time.

Variable Elimination: Sketch of Ideas

(A) Avoiding repeated computation: Evaluate expressions from right to left, storing all intermediate results. For query $P(B \mid j, m)$:

1. CPTs of BN yield factors (probability tables): $\mathbf{P}(B \mid j, m) =$
$$\alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_{v_E} \underbrace{P(v_E)}_{\mathbf{f}_2(E)} \sum_{v_A} \underbrace{\mathbf{P}(v_A \mid B, v_E)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid v_A)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid v_A)}_{\mathbf{f}_5(A)}$$
2. Then the computation is performed in terms of factor product and summing out variables from factors: $\mathbf{P}(B \mid j, m) =$
$$\alpha\, \mathbf{f}_1(B) \times \sum_{v_E} \mathbf{f}_2(E) \times \sum_{v_A} \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

(B) Avoiding irrelevant computation: Repeatedly remove hidden variables that are leaf nodes. For query $P(JohnCalls \mid burglary)$:

$$\mathbf{P}(J \mid b) = \alpha P(b) \sum_{v_E} P(v_E) \sum_{v_A} P(v_A \mid b, v_E) \mathbf{P}(J \mid v_A) \sum_{v_M} P(v_M \mid v_A)$$

$\rightarrow$ The rightmost sum equals 1 and can be dropped.

Variable Elimination Runtime

An important easy case:

- A graph is called singly connected, or a polytree, if there is at most one undirected path between any two nodes in the graph.
- On polytree Bayesian networks, variable elimination runs in polynomial time.

$\rightarrow$ Is our BN for Mary & John a polytree? Yes.

Summary

- Bayesian networks (BN) are a wide-spread tool to model uncertainty, and to reason about it. A BN represents conditional independence relations between random variables. It consists of a graph encoding the variable dependencies, and of conditional probability tables (CPTs).
- Given a variable order, the BN is small if every variable depends on only a few of its predecessors.
- Probabilistic inference requires to compute the probability distribution of a set of query variables, given a set of evidence variables whose values we know. The remaining variables are hidden.
- Inference by enumeration takes a BN as input, then applies Normalization+Marginalization, the Chain rule, and exploits conditional independence. This can be viewed as a tree search that branches over all values of the hidden variables.
- Variable elimination avoids unnecessary computation. It runs in polynomial time for poly-tree BNs. In general, exact probabilistic inference is **#P**-hard. Approximate probabilistic inference methods exist.

Topics We Didn't Cover Here

- Inference by sampling: A whole zoo of methods for doing this exists.
- Clustering: Pre-combining subsets of variables to reduce the runtime of inference.
- Compilation to SAT: More precisely, to "weighted model counting" in CNF formulas. Model counting extends DPLL with the ability to determine the number of satisfying interpretations. Weighted model counting allows to define a mass for each such interpretation (= the probability of an atomic event).
- Dynamic BN: BN with one slice of variables at each "time step", encoding probabilistic behavior over time.
- Relational BN: BN with predicates and object variables.
- First-order BN: Relational BN with quantification, i.e., probabilistic logic. E.g., the BLOG language developed by Stuart Russel and co-workers.

Reading

- *Chapter 14: Probabilistic Reasoning* [Russell and Norvig (2010)].

  Content: Section 14.1 roughly corresponds to my "What is a Bayesian Network?".

  Section 14.2 roughly corresponds to my "What is the Meaning of a Bayesian Network?" and "Constructing Bayesian Networks".The main change I made here is to *define* the semantics of the BN in terms of the conditional independence relations, which I find clearer than RN's definition that uses the reconstructed full joint probability distribution instead.

  Section 14.4 roughly corresponds to my "Inference in Bayesian Networks". RN give full details on variable elimination, which makes for nice ongoing reading.

  Section 14.3 discusses how CPTs are specified in practice. Section 14.5 covers approximate sampling-based inference. Section 14.6 briefly discusses relational and first-order BNs. Section 14.7 briefly discusses other approaches to reasoning about uncertainty. All of this is nice as additional background reading.

References I

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (Third Edition)*. Prentice-Hall, Englewood Cliffs, NJ, 2010.