Artificial Intelligence

Bayesian networks
Chapter 14, AIMA

Inference

- Inference in the statistical setting means computing probabilities for different outcomes to be true given the information

$$\mathbf{P}(Outcome \mid Information)$$

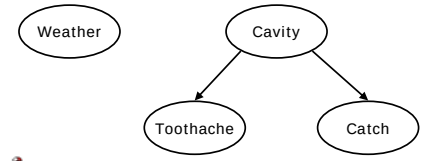
- We need an efficient method for doing this, which is more powerful than the naïve Bayes model.

Bayesian networks

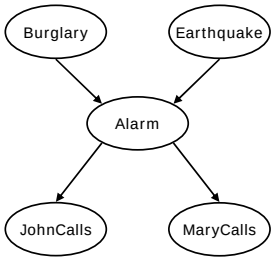
A **Bayesian network** is a directed graph in which each node is annotated with quantitative probability information:

1. A set of random variables, $\{X_1, X_2, X_3, \ldots\}$, makes up the nodes of the network.
2. A set of directed links connect pairs of nodes, parent $\rightarrow$ child
3. Each node X_i has a conditional probability distribution $\mathbf{P}(X_i \mid Parents(X_i))$.
4. The graph is a directed acyclic graph (DAG).

The dentist network



The alarm network



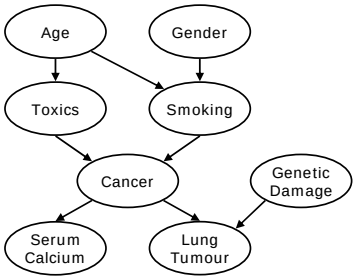
Burglar alarm responds to both earthquakes and burglars.

Two neighbors: John and Mary, who have promised to call you when the alarm goes off.

John always calls when there's an alarm, and sometimes when there's not an alarm.

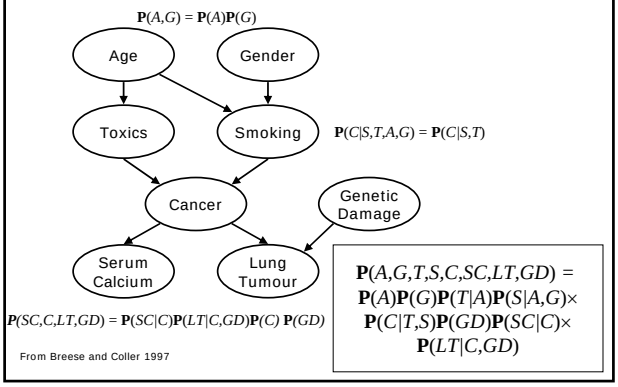
Mary sometimes misses the alarms (she likes loud music).

The cancer network



From Breese and Coller 1997

The cancer network



From Breese and Coller 1997

$$\mathbf{P}(A,G,T,S,C,SC,LT,GD) = \mathbf{P}(A)\mathbf{P}(G)\mathbf{P}(T|A)\mathbf{P}(S|A,G)\times \mathbf{P}(C|T,S)\mathbf{P}(GD)\mathbf{P}(SC|C)\times \mathbf{P}(LT|C,GD)$$

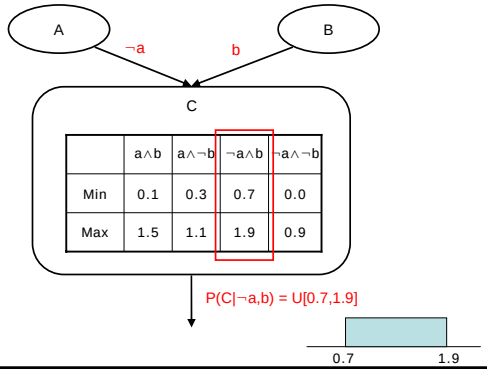
The product (chain) rule

$$P(X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_n = x_n) =$$

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

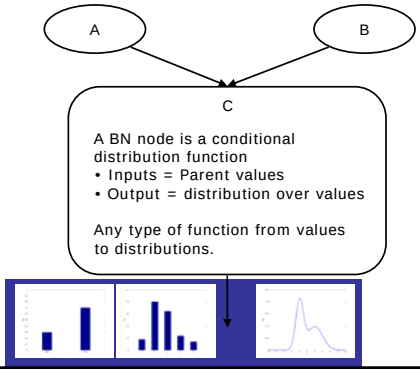
(This is for Bayesian networks, the general case comes later in this lecture)

Bayes network node is a function



	a∧b	a∧¬b	¬a∧b	¬a∧¬b
Min	0.1	0.3	0.7	0.0
Max	1.5	1.1	1.9	0.9

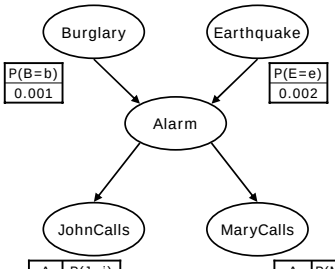
Bayes network node is a function



A BN node is a conditional distribution function
- Inputs = Parent values
- Output = distribution over values

Any type of function from values to distributions.

Example: The alarm network



Note: Each number in the tables represents a boolean distribution. Hence there is a distribution output for every input.

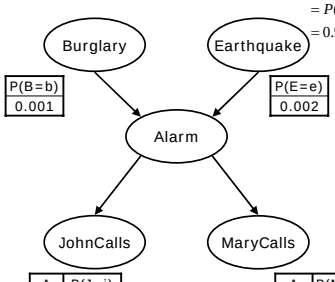
B	E	P(A=a)
b	e	0.95
b	¬e	0.94
¬b	e	0.29
¬b	¬e	0.001

A	P(J=j)
a	0.90
¬a	0.05

A	P(M=m)
a	0.70
¬a	0.01

Example: The alarm network

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$$
$$= P(\neg b)P(\neg e)P(a \mid \neg b, \neg e)P(m \mid a)P(j \mid a)$$
$$= 0.999 \cdot 0.998 \cdot 0.001 \cdot 0.70 \cdot 0.90 = 0.00063$$



Probability distribution for "no earthquake, no burglary, but alarm, and both Mary and John make the call"

B	E	P(A=a)
b	e	0.95
b	¬e	0.94
¬b	e	0.29
¬b	¬e	0.001

A	P(J=j)
a	0.90
¬a	0.05

A	P(M=m)
a	0.70
¬a	0.01

Meaning of Bayesian network

The general chain rule (always true):

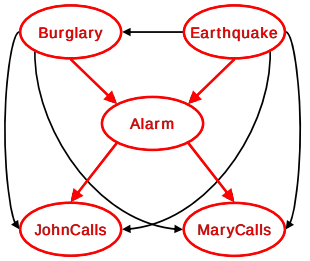
$$P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, x_3, \ldots, x_n) P(x_2, x_3, \ldots, x_n) =$$
$$P(x_1 \mid x_2, x_3, \ldots, x_n) P(x_2 \mid x_3, x_4, \ldots, x_n) P(x_3, x_4, \ldots, x_n) = \cdots$$
$$= \prod_{i=1}^{n} P(x_i \mid x_{i+1}, \ldots, x_n)$$

The Bayesian network chain rule:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

The BN is a correct representation of the domain iff each node is conditionally independent of its predecessors, given its parents.

The alarm network



The fully correct alarm network might look something like the figure.

The Bayesian network (red) assumes that some of the variables are independent (or that the dependecies can be neglected since they are very weak).

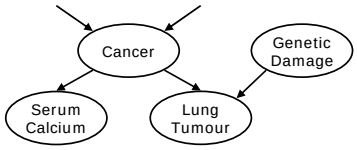
The correctness of the Bayesian network of course depends on the validity of these assumptions.

It is this sparse connection structure that makes the BN approach feasible (~linear growth in complexity rather than exponential)

How construct a BN?

- Add nodes in causal order ("causal" determined from expertize).
- Determine conditional independence using either (or all) of the following semantics:
  - Blocking/d-separation rule
  - Non-descendant rule
  - Markov blanket rule
  - Experience/your beliefs

Path blocking & d-separation



Intuitively, knowledge about Serum Calcium influences our belief about Cancer, if we don't know the value of Cancer, which in turn influences our belief about Lung Tumour, etc.

However, if we are given the value of Cancer (i.e. C= *true* or *false*), then knowledge of Serum Calcium will not tell us anything about Lung Tumour that we don't already know.

We say that Cancer **d-separates** (direction-dependent separation) Serum Calcium and Lung Tumour.

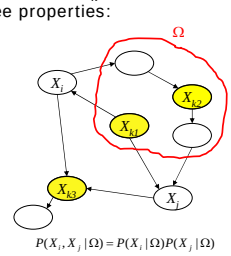
Path blocking & d-separation

X_i and X_j are d-separated if all paths betweeen them are blocked

Two nodes X_i and X_j are conditionally independent given a set $\Omega = \{X_1, X_2, X_3, \ldots\}$ of nodes if for every undirected path in the BN between X_i and X_j there is some node X_k on the path having one of the following three properties:

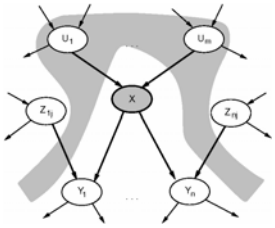
1. $X_k \in \Omega$, and both arcs on the path lead out of X_k.
2. $X_k \in \Omega$, and one arc on the path leads into X_k and one arc leads out.
3. Neither X_k nor any descendant of X_k is in Ω, and both arcs on the path lead into X_k.

X_k **blocks** the path between X_i and X_j



Non-descendants

A node is conditionally independent of its non-descendants (Z_{ij}), given its parents.

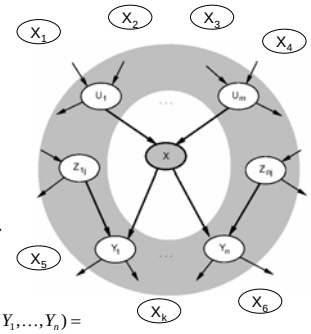


$$P(X, Z_{1j}, \ldots, Z_{nj} \mid U_1, \ldots, U_m) =$$
$$P(X \mid U_1, \ldots, U_m) P(Z_{1j}, \ldots, Z_{nj} \mid U_1, \ldots, U_m)$$

Markov blanket

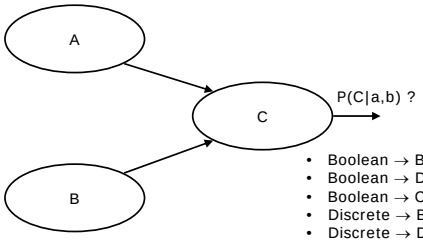
A node is conditionally independent of all other nodes in the network, given its parents, children, and children's parents

These constitute the nodes *Markov blanket*.



$$P(X, X_1,\ldots,X_k \mid U_1,\ldots,U_m,Z_{1j},\ldots,Z_{nj},Y_1,\ldots,Y_n) =$$
$$P(X \mid U_1,\ldots,U_m,Z_{1j},\ldots,Z_{nj},Y_1,\ldots,Y_n)P(X_1,\ldots,X_k \mid U_1,\ldots,U_m,Z_{1j},\ldots,Z_{nj},Y_1,\ldots,Y_n)$$

Efficient representation of PDs



- Boolean → Boolean
- Boolean → Discrete
- Boolean → Continuous
- Discrete → Boolean
- Discrete → Discrete
- Discrete → Continuous
- Continuous → Boolean
- Continuous → Discrete
- Continuous → Continuous

Efficient representation of PDs

Boolean → Boolean:
Noisy-OR, Noisy-AND

Boolean/Discrete → Discrete:
Noisy-MAX

Bool./Discr./Cont. → Continuous:
Parametric distribution (e.g. Gaussian)

Continuous → Boolean:
Logit/Probit

Noisy-OR example

Boolean → Boolean

P(E|C1,C2,C3)								
C_1	0	1	0	0	1	1	0	1
C_2	0	0	1	0	1	0	1	1
C_3	0	0	0	1	0	1	1	1
P(E=0)	1	0.1	0.1	0.1	0.01	0.01	0.01	0.001
P(E=1)	0	0.9	0.9	0.9	0.99	0.99	0.99	0.999

The effect (E) is off (false) when none of the causes are true. The probability for the effect increases with the number of true causes.

$$P(E=0) = 10^{-(\#True)}$$ (for this example)

Example from L.E. Sucar

Noisy-OR general case

Boolean → Boolean

$$P(E=0 \mid C_1,C_2,\ldots,C_n) = \prod_{i=1}^{n} q_i^{C_i}$$

$$C_i = \begin{cases} 1 & \text{if } true \\ 0 & \text{if } false \end{cases}$$

Example on previous slide used q_i = 0.1 for all i.

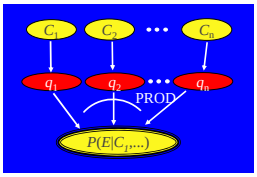


Image adapted from Laskey & Mahoney 1999

Noisy-MAX

Boolean → Discrete

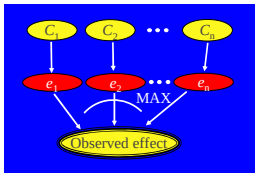


Image adapted from Laskey & Mahoney 1999

Effect takes on the max value from different causes

Restrictions:

- Each cause must have an off state, which does not contribute to effect
- Effect is off when all causes are off
- Effect must have consecutive escalating values: e.g., absent, mild, moderate, severe.

$$P(E=e_k \mid C_1,C_2,\ldots,C_n) = \prod_{i=1}^{n} q_{i,k}^{C_i}$$

Parametric probability densities

Boolean/Discr./Continuous → Continuous

Use parametric probability densities, e.g., the normal distribution

$$P(X) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) = N(\mu,\sigma)$$

Gaussian networks (a = input to the node)

$$P(X) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\alpha-\beta a)^2}{2\sigma^2}\right]$$

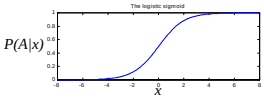
Probit & Logit

Discrete → Boolean

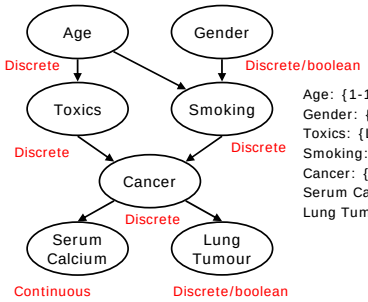
If the input is continuous but output is boolean, use probit or logit

$$\text{Logit}: P(A=a \mid x) = \frac{1}{1+\exp\left[-2(\mu-x)/\sigma\right]}$$

$$\text{Probit}: P(A=a \mid x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-(x-\mu)^2/\sigma^2)dx$$



The cancer network



Age: {1-10, 11-20,...}
Gender: {M, F}
Toxics: {Low, Medium, High}
Smoking: {No, Light, Heavy}
Cancer: {No, Benign, Malignant}
Serum Calcium: Level
Lung Tumour: {Yes, No}

Inference in BN

Inference means computing $\mathbf{P}(X|\mathbf{e})$, where X is a query (variable) and $\mathbf{e}$ is a set of evidence variables (for which we know the values).

Examples:

P(Burglary | john_calls, mary_calls)

P(Cancer | age, gender, smoking, serum_calcium)

P(Cavity | toothache, catch)

Exact inference in BN

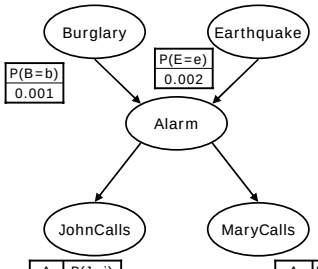
$$\mathbf{P}(X \mid \mathbf{e}) = \frac{\mathbf{P}(X,\mathbf{e})}{\mathbf{P}(\mathbf{e})} = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha\sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})$$

"Doable" for boolean variables: Look up entries in conditional probability tables (CPTs).

Example: The alarm network

What is the probability for a burglary if both John and Mary call?

$$\mathbf{P}(B \mid j,m) = \alpha \sum_{E=\{e,\neg e\}} \sum_{A=\{a,\neg a\}} \mathbf{P}(B,E,A,j,m)$$



Evidence variables = {J,M}
Query variable = B

A	P(J=j)
a	0.90
¬a	0.05

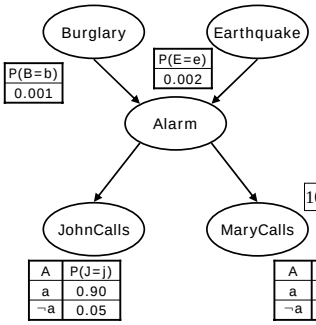
A	P(M=m)
a	0.70
¬a	0.01

B	E	P(A=a)
b	e	0.95
b	¬e	0.94
¬b	e	0.29
¬b	¬e	0.001

Example: The alarm network

What is the probability for a burglary if both John and Mary call?

$$\mathbf{P}(B \mid j,m) = \alpha \sum_{E=\{e,\neg e\}} \sum_{A=\{a,\neg a\}} \mathbf{P}(B,E,A,j,m)$$



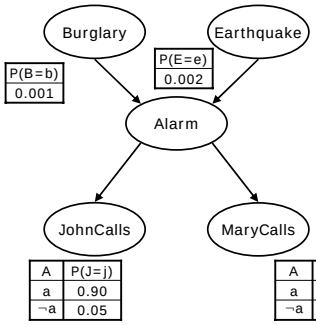
$$\mathbf{P}(B=b,E,A,j,m) =$$
$$\mathbf{P}(j,m \mid b,E,A)\mathbf{P}(b,E,A) =$$
$$P(j \mid A)P(m \mid A)\mathbf{P}(b,E,A) =$$
$$P(j \mid A)P(m \mid A)\mathbf{P}(a \mid b,E)\mathbf{P}(b,E) =$$
$$P(j \mid A)P(m \mid A)P(a \mid b,E)\underbrace{P(b)}_{=0.001}P(E)$$

$$10^{-3} \times P(j \mid A)P(m \mid A)P(A \mid b,E)P(E)$$

Example: The alarm network

What is the probability for a burglary if both John and Mary call?

$$\mathbf{P}(B \mid j,m) = \alpha \sum_{E=\{e,\neg e\}} \sum_{A=\{a,\neg a\}} \mathbf{P}(B,E,A,j,m)$$

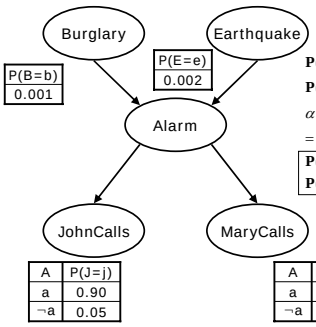


$$\mathbf{P}(b,j,m) =$$
$$10^{-3} \sum_{\substack{A=\{a,\neg a\} \\ E=\{e,\neg e\}}} P(j \mid A)P(m \mid A)P(A \mid b,E)P(E) =$$
$$10^{-3}[P(j \mid a)P(m \mid a)P(a \mid b,e)P(e) +$$
$$P(j \mid a)P(m \mid a)P(a \mid b,\neg e)P(\neg e) +$$
$$P(j \mid \neg a)P(m \mid \neg a)P(\neg a \mid b,e)P(e) +$$
$$P(j \mid \neg a)P(m \mid \neg a)P(\neg a \mid b,\neg e)P(\neg e)] =$$
$$0.5923 \times 10^{-3}$$
$$\mathbf{P}(\neg b,j,m) = 1.491 \times 10^{-3}$$

Example: The alarm network

What is the probability for a burglary if both John and Mary call?

$$\mathbf{P}(B \mid j,m) = \alpha \sum_{E=\{e,\neg e\}} \sum_{A=\{a,\neg a\}} \mathbf{P}(B,E,A,j,m)$$



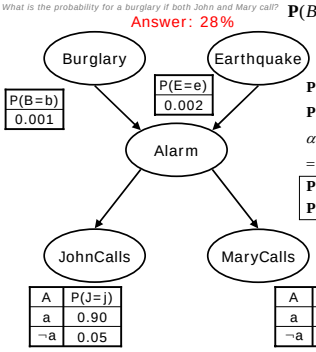
$$\mathbf{P}(b,j,m) = 0.5923 \times 10^{-3}$$
$$\mathbf{P}(\neg b,j,m) = 1.491 \times 10^{-3}$$
$$\alpha = \mathbf{P}(j,m)^{-1} = [\mathbf{P}(b,j,m) + \mathbf{P}(\neg b,j,m)]^{-1} =$$
$$= [2.083 \times 10^{-3}]^{-1}$$
$$\mathbf{P}(b \mid j,m) = \alpha \mathbf{P}(b,j,m) = 0.284$$
$$\mathbf{P}(\neg b \mid j,m) = \alpha \mathbf{P}(\neg b,j,m) = 0.716$$

Example: The alarm network

What is the probability for a burglary if both John and Mary call?

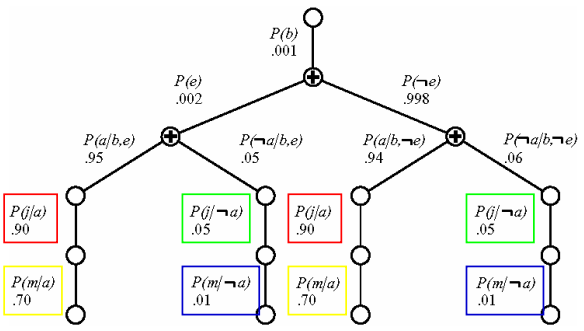
Answer: 28%

$$\mathbf{P}(B \mid j,m) = \alpha \sum_{E=\{e,\neg e\}} \sum_{A=\{a,\neg a\}} \mathbf{P}(B,E,A,j,m)$$



$$\mathbf{P}(b,j,m) = 0.5923 \times 10^{-3}$$
$$\mathbf{P}(\neg b,j,m) = 1.491 \times 10^{-3}$$
$$\alpha = \mathbf{P}(j,m)^{-1} = [\mathbf{P}(b,j,m) + \mathbf{P}(\neg b,j,m)]^{-1} =$$
$$= [2.083 \times 10^{-3}]^{-1}$$
$$\mathbf{P}(b \mid j,m) = \alpha \mathbf{P}(b,j,m) = 0.284$$
$$\mathbf{P}(\neg b \mid j,m) = \alpha \mathbf{P}(\neg b,j,m) = 0.716$$

Use depth-first search



A lot of unneccesary repeated computation…

Complexity of exact inference

- By eliminating repeated calculation & uninteresting paths we can speed up the inference a lot.
- Linear time complexity for singly connected networks (polytrees).
- Exponential for multiply connected networks.
  - Clustering can improve this

Approximate inference in BN

- Exact inference is intractable in large multiply connected BNs ⇒
  use approximate inference:
  Monte Carlo methods (random sampling).
  - Direct sampling
  - Rejection sampling
  - Likelihood weighting
  - Markov chain Monte Carlo

Markov chain Monte Carlo

1. Fix the evidence variables (E_1, E_2, ...) at their given values.
2. Initialize the network with values for all other variables, including the query variable.
3. Repeat the following many, many, many times:
   a. Pick a non-evidence variable at random (query X_i or hidden Y_j)
   b. Select a new value for this variable, conditioned on the current values in the variable's Markov blanket.

Monitor the values of the query variables.