Intelligent Autonomous Agents
and Cognitive Robotics
Topic 5: Bayesian Networks

Ralf Möller, Rainer Marrone
Hamburg University of Technology

Uncertainty in prior knowledge

- Diagnosis:
  - Toothache => Cavity $\vee$ *GumProblem* $\vee$ *Abscess* $\vee$ …

    RootInfection $\vee$ … $\vee$ *Cavity* => *Toothache*

- The connection between toothaches and cavity is just not a logical consequence. For medical diagnosis logic does not seem to be appropriate.

2

Probability

Probabilistic assertions summarize effects of

- laziness:
  It is too much work to list the complete set of antecedents or consequents needed to ensure an exceptionless rule and too hard to use such rules
- theoretical ignorance:
  no complete theory, e.g., medical science has no complete theory for the domain.
- practical ignorance:
  lack of relevant facts, initial conditions, tests, etc.

3

Making decisions under uncertainty

Suppose I believe the following:

$P(A_{25}$ gets me there on time | …) = 0.04
$P(A_{90}$ gets me there on time | …) = 0.70
$P(A_{120}$ gets me there on time | …) = 0.95
$P(A_{1440=24h}$ gets me there on time | …) = 0.9999

- Which action to choose?

  Depends on my preferences for missing flight vs. time spent waiting, etc.
  - Utility theory is used to represent and use preferences
  - Decision theory = **probability theory + utility theory**

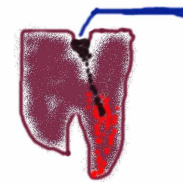
Later in this lecture

4

Example world

Example: ***Dentist problem*** **with four variables:**

Toothache **(I have a toothache)**
Cavity **(I have a cavity)**
Catch **(steel probe catches in my tooth)**
Weather **(*sunny,rainy,cloudy,snow*)**



5

Prior probability

- Prior or unconditional probabilities of propositions
  e.g., P(*Cavity* = true) = 0.1 and P(*Weather* = sunny) = 0.72
  correspond to belief prior to arrival of any (new) evidence

- Probability distribution
  gives values for all possible assignments
  (*sunny,rainy,cloudy,snow*):
  P(*Weather*) = <0.72,0.1,0.08,0.1>
  (normalized, i.e., sums to 1 because one must be the case)

6

Full joint probability distribution

- Joint probability distribution for a set of random variables gives the probability of every atomic event on those random variables
  P(*Weather,Cavity*) = a 4 × 2 matrix of values:

Weather =	sunny	rainy	cloudy	snow	
Cavity = true	0.144	0.02	0.016	0.02	= 0.2
Cavity = false	0.576	0.08	0.064	0.08	= 0.8
					= 1.0

- Full joint probability distribution: all random variables involved
  - **P**(Toothache, Catch, Cavity, Weather)
- Every question about a domain can be answered by the full joint distribution

7

Conditional probability

- Conditional or posterior probabilities (after having received some information)
  e.g., P(*cavity* | *toothache*) = 0.8
- Definition of conditional probability (in terms of uncond. prob.):
  P(a | b) = P(a ∧ b) / P(b) if P(b) > 0
- Product rule gives an alternative formulation (∧ is commutative):
  P(a ∧ b) = P(a | b) P(b) = P(b | a) P(a)
- Chain rule is derived by successive application of product rule:

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_n) &= \mathbf{P}(X_1,\ldots,X_{n-1})\ \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1}) \\ &= \mathbf{P}(X_1,\ldots,X_{n-2})\ \mathbf{P}(X_{n-1} \mid X_1,\ldots,X_{n-2})\ \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1}) \\ &= .\prod_{i=1}^{n} \\ &= \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})\end{aligned}$$

8

Bayes rule

Product rule: $P(x,y) = P(x|y)P(y) = P(y|x)P(x)$

$\Longrightarrow$

$$P(cause|effect) = \frac{P(effect|cause) * P(cause)}{P(effect)} = \frac{likelihood * prior}{evidence}$$

$$P(X|Y) = \alpha\, P(Y|X)P(X)$$

9

Inference by enumeration

- Start with the joint probability distribution:

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	.108	.012	.072	.008
¬ *cavity*	.016	.064	.144	.576

- For any proposition φ, sum the atomic events where it is true: $P(\varphi) = \Sigma_{\omega \epsilon \varphi} P(\omega)$

10

Inference by enumeration

- Start with the joint probability distribution:

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	.108	.012	.072	.008
¬ *cavity*	.016	.064	.144	.576

P(cavity ∨ *toothache*) =
0.108 + 0.012 + 0.072 + 0.008+ 0.016 + 0.064 = 0.28

11

Inference by enumeration

- Start with the joint probability distribution:

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	.108	.012	.072	.008
¬ *cavity*	.016	.064	.144	.576

- Can also compute conditional probabilities:

$$P(\neg cavity \mid toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)}$$

Product rule

$$= \frac{0.016+0.064}{0.108 + 0.012 + 0.016 + 0.064}$$

$$= 0.4$$

12

Normalization

	toothache		¬ toothache	
	catch	¬ catch	catch	¬ catch
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- Denominator **P**(**z**) (or P(toothache) in the example before) can be viewed as a normalization constant α

P(*Cavity* | *toothache*) = **P**(Cavity,toothache)/P(toothache)
= α **P**(*Cavity,toothache*)
= α [**P**(*Cavity,toothache,catch*) + **P**(*Cavity,toothache,*¬ *catch*)]
= α [<0.108,0.016> + <0.012,0.064>]
= α <0.12,0.08> = <0.6,0.4>

α*(0,12+0,08)=1
α=1/0,2=5
5*0,12=0,6
5*0,08=0,4

General idea: compute distribution on query variable by fixing evidence variables (toothache) and summing over hidden variables (Catch)

13

General inference procedure

Typically, we are interested in
the posterior joint distribution of the query variables **Y**
given specific values **e** for the evidence variables **E**
X are all variables of the modeled world

Let the hidden variables be **H** = **X** - **Y** – **E** then the required summation of joint entries is done by summing out the hidden variables:

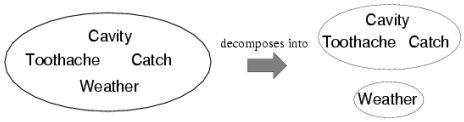
P(**Y** | **E** = **e**) = α**P**(**Y**,**E** = **e**) = $\alpha\Sigma_h$**P**(**Y**,**E**= **e**, **H** = **h**)

- The terms in the summation are joint entries because **Y**, **E** and **H** together exhaust the set of random variables (**X**)
- Obvious problems:
  1. Space complexity $O(d^n)$ to store the joint distribution where d is the largest arity and n denotes the number of random variables
  2. Worst-case time complexity $O(d^n)$
  3. How to find the numbers for $O(d^n)$ entries?

14

Independence

- *A* and *B* are independent iff
  P(*A*|*B*) = **P**(*A*) or **P**(*B*|*A*) = **P**(*B*) or **P**(A, B) = **P**(*A*) **P**(*B*)



P(*Toothache, Catch, Cavity, Weather*)
= **P**(*Toothache, Catch, Cavity*) **P**(*Weather*)

- 32 entries table can be constructed from 8 and 4 entries;
- Absolute independence powerful but rare
- How can we check whether we have independent variables in the full joint?

15

Example #1

Bread	Bagels	Butter	p(r,a,u)
0	0	0	0.24
0	0	1	0.06
0	1	0	0.12
0	1	1	0.08
1	0	0	0.12
1	0	1	0.18
1	1	0	0.04
1	1	1	0.16

Bread	p(r)
0	
1	

P(a,u)=P(a)P(u)?

P(r,a)=P(r)P(a)?

16

Example #1

Bread	Bagels	Butter	p(r,a,u)
0	0	0	0.24
0	0	1	0.06
0	1	0	0.12
0	1	1	0.08
1	0	0	0.12
1	0	1	0.18
1	1	0	0.04
1	1	1	0.16

Butter	p(u)
0	0.52
1	0.48

Bagels	p(a)
0	0.6
1	0.4

Bread	p(r)
0	0.5
1	0.5

Bagels	Butter	p(a,u)
0	0	
0	1	
1	0	
1	1	

?

P(a,u)=P(a)P(u)?

P(r,a)=P(r)P(a)?

17

Example #1

Bread	Bagels	Butter	p(r,a,u)
0	0	0	0.24
0	0	1	0.06
0	1	0	0.12
0	1	1	0.08
1	0	0	0.12
1	0	1	0.18
1	1	0	0.04
1	1	1	0.16

Butter	p(u)
0	0.52
1	0.48

Bagels	p(a)
0	0.6
1	0.4

Bread	p(r)
0	0.5
1	0.5

Bagels	Butter	p(a,u)
0	0	0.36
0	1	0.24
1	0	0.16
1	1	0.24

≠ 0.52*0.6=0.312

Bread	Bagels	p(r,a)
0	0	0.3
0	1	0.2
1	0	0.3
1	1	0.2

P(a,u)=P(a)P(u)? *NO*

P(r,a)=P(r)P(a)? *YES*

18

Conditional independence

- **P**(*Toothache, Cavity, Catch*) has $2^3 - 1 = 7$ independent entries
- If I have a cavity, the probability that the probe catches in doesn't depend on whether I have a toothache:
  (1) **P**(*catch* | *toothache, cavity*) = **P**(*catch* | *cavity*)
  (2) **P**(*catch* | *toothache*,¬*cavity*) = **P**(*catch* | ¬*cavity*)
- *Catch* is conditionally independent of *Toothache* given *Cavity*:
  P(*Catch* | *Toothache,Cavity*) = **P**(*Catch* | *Cavity*)
- Equivalent statements:
  P(*Toothache* | *Catch, Cavity*) = **P**(*Toothache* | *Cavity*)

  P(*Toothache, Catch* | *Cavity*) = **P**(*Toothache* | *Cavity*) **P**(*Catch* | *Cavity*)

19

Conditional independence contd.

- Write out full joint distribution using chain rule:
  P(*Toothache, Catch, Cavity*)
  = **P**(*Toothache* | *Catch, Cavity*) **P**(*Catch, Cavity*)
  = **P**(*Toothache* | *Catch, Cavity*) **P**(*Catch* | *Cavity*) **P**(*Cavity*)
  conditional independence
  = **P**(*Toothache* | *Cavity*) **P**(*Catch* | *Cavity*) **P**(Cavity)

  i.e., 2 + 2 + 1 = 5 independent numbers
- In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in *n* to linear in *n*.
- Conditional independence is our most basic and robust form of knowledge about uncertain environments.

20

Car Example

- Three variables:
  - Gas, Battery, Starts
- P(Battery|Gas) = P(Battery)
  Gas and Battery are independent
- P(Battery|Gas,Starts) $\overset{?}{\neq}$ P(Battery|Starts)

  Gas and Battery are not independent given Starts
- Independence does not imply conditional independence.
- Conditional independence does not imply independence

21

Question

- How can we make use of
  - independence
  - and conditional independence

  Need a model that can express this

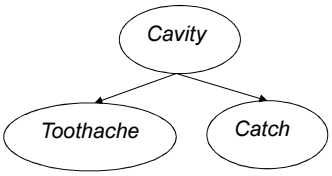
22

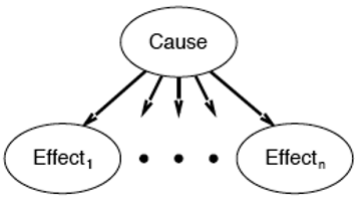
Bayesian networks

- A simple, graphical notation for **conditional independence assertions** and hence for compact specification of the full joint distributions
- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - a conditional distribution for each node given its parents:
    P (X_i | Parents (X_i))
- In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

23

Simplest Bayesian Network





$$P(Cause|Effect_1, Effect_2, \ldots) = \alpha P(Cause) \prod_i P(Effect_i|Cause)$$

- also called Naïve Bayesian networks
- conditional independence of all effect variables

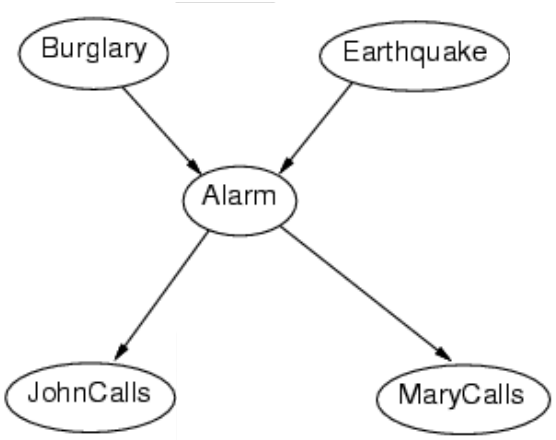
24

More complex example

- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary calls but not as often as John. Sometimes it's set off by minor earthquakes but also on burglary. Is there a burglar?
- Variables: *Burglary*, *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

25

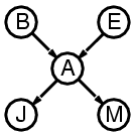
Example contd.



26

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 = *true* (the number for X_i = *false* is just *1-p*)



- If each of n Boolean variables 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)

k parents with **n** values each and **m** values for the child node of the parents?

Number of indepenent values = $n^k \cdot (m-1)$

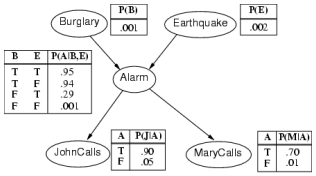
27

Semantics

The full joint distribution can be rewritten using the *chain rule*:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$$

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \, parent(X_i))$$



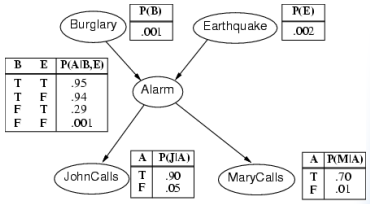
Assumption: Independence and Conditional independence assertions are correctly modeled

28

Semantics

The full joint distribution is defined as the product of the local conditional distributions:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parent(X_i))$$



e.g., ***P***(*j* ∧ *m* ∧ *a* ∧ ¬*b* ∧ ¬*e*)

= ***P*** (*j* | *a*) ***P*** (*m* | *a*) ***P*** (*a* | ¬*b*, ¬*e*) ***P*** (¬*b*) ***P*** (¬*e*)

= 0.90x0.7x0.001x0.999x0.998

≈ 0.00063

29

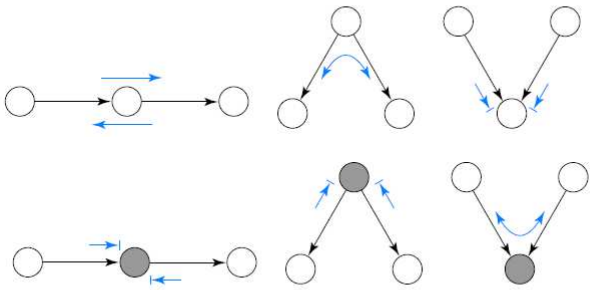
Encoding conditional independence via d-separation

- We can determine if conditional independence holds by a graph separation criterion called ***d-separation*** *(direction dependent separation)*
- X and Y are ***d-separated*** if there is no active path between them.
- The formals definition of ***active*** is somewhat involved. The Bayes Ball Algorithm gives a nice graphical definition.

30

The six rules of Bayes Ball

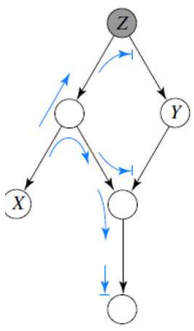
An undirected path is active if a Bayes ball travelling along it never encounters the "stop" symbol: $\rightarrow\!|$

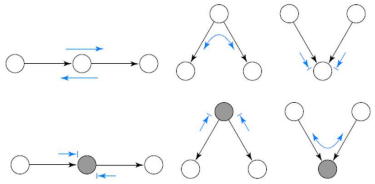


If there are no active paths from X to Y when $\{Z_1, \ldots, Z_k\}$ are shaded, then $X \perp\!\!\!\perp Y \mid \{Z_1, \ldots, Z_k\}$.

31

A double-header: two games of Bayes Ball



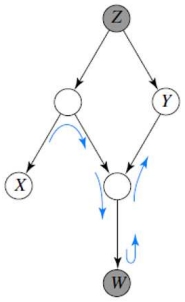


no active paths

$X \perp\!\!\!\perp Y \mid Z$

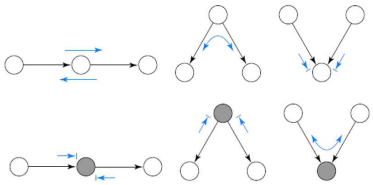
32

A double-header: two games of Bayes Ball



one active path

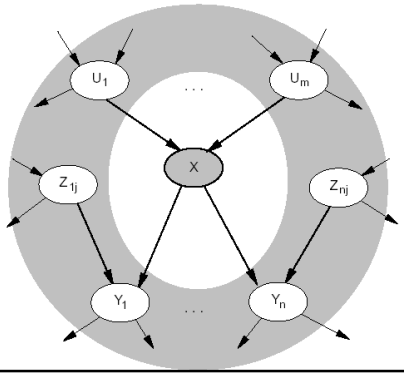
$X \not\perp Y \mid \{W, Z\}$



33

Markov Blanket

- Markov blanket: Parents + children + children's parents
- Node is conditionally independent of all other nodes in network, given its Markov Blanket -> simplifies computation -> gather information on the nodes of the Markov Blanket?



34

Constructing Bayesian networks

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

This choice of parents guarantees:

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$$
(chain rule)
$$= \prod_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$$
(by construction)

35

Example

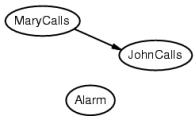
- Suppose we choose the ordering *M, J, A, B, E*

$\mathbf{P}(J \mid M) = \mathbf{P}(J)$? **No**

36

Example

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



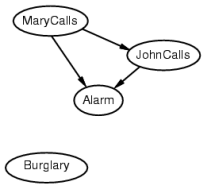
P(*A* | *J, M*) = ***P***(*A*)? ***No***

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

37

Example

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



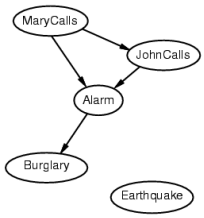
P(*B* | *A, J, M*) = ***P***(*B*)? ***No***

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

38

Example

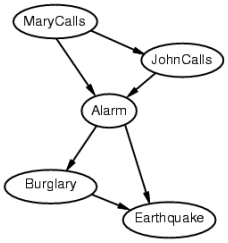
- Suppose we choose the ordering M, J, A, B, E



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

39

Example contd.



- Deciding conditional independence is hard in noncausal directions
- (Causal models and conditional independence seem hardwired for humans!)
- Network is less compact: 1 + 2 + 4 + 2 + 4 = 13 numbers needed instead of 10.

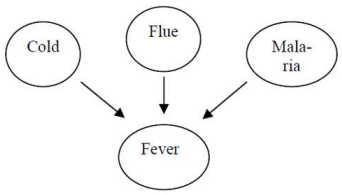
40

Efficient implementation of CPTs

- The number of independent entries grow exponentially with the number of parents.
- Two ways to overcome this
  - Restrict the number of parents if possible
  - Instead of free distributions, often canonical (parameterized) distributions are suggested. One popular example of such a pattern is the noisy OR for discrete cases.

41

Example



The noisy OR is a generalization of the logical OR. Three assumptions:

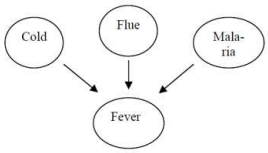
1. All possible causes U_i for a event X are listed (you can add a *leak* node)
2. Negated causes $\neg U_i$ do not have any influence on X
3. Independent failure probability q_i for each cause alone.

$$q_{\text{cold}} = P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6\ ,$$
$$q_{\text{flu}} = P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2\ ,$$
$$q_{\text{malaria}} = P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1\ .$$

42

Example

$q_{\text{cold}} = P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6$,
$q_{\text{flu}} = P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2$,
$q_{\text{malaria}} = P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1$.

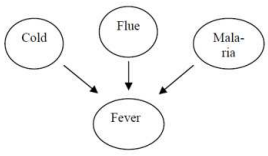


$$P(\neg x \mid o_1, o_2, \ldots, o_r, \neg o_{r+1}, \ldots, \neg o_n) = \prod_{r=1}^{r} q_i$$

Cold	*Flu*	*Malaria*	*P(Fever)*	*P(¬Fever)*
F	F	F		
F	F	T		**0.1**
F	T	F		**0.2**
F	T	T		
T	F	F		**0.6**
T	F	T		
T	T	F		
T	T	T		

Example

$q_{\text{cold}} = P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6$,
$q_{\text{flu}} = P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2$,
$q_{\text{malaria}} = P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1$.

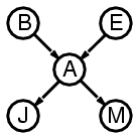


$$P(x \mid o_1, o_2, \ldots, o_r, \neg o_{r+1}, \ldots, \neg o_n) = 1 - \prod_{r=1}^{r} q_i$$

Cold	*Flu*	*Malaria*	*P(Fever)*	*P(¬Fever)*
F	F	F	0.0	1.0
F	F	T	0.9	**0.1**
F	T	F	0.8	**0.2**
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Last Time

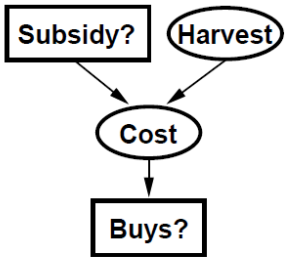
- Structure and semantic of BN
  - Modelling of independence and conditional independence
  - Causal and non-causal networks
  - d-separation, Markov blanket
  - Efficient CPTs, e.g., noisy OR, trees, Min, Max, …



45

Hybrid (discrete+contionous) networks

Discrete (*Subsidy?* and *Buys?*); continuous (*Harvest* and *Cost*)



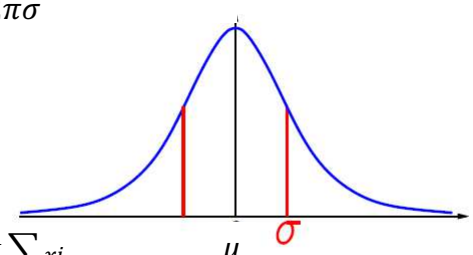
Option 1: discretization—possibly large errors, large CPTs
Option 2: finitely parameterized canonical families

1) Continuous variable, discrete+continuous parents (e.g., *Cost*)
2) Discrete variable, continuous parents (e.g., *Buys?*)

46

Continous variables: Gaussian density

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}$$



$$Mean \quad \mu = \frac{1}{n}\sum_{n} xi$$

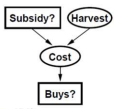
$$Variance\ \sigma^2 = \frac{1}{n-1}\sum_{n}(xi - \mu)^2$$

$$Standard\ deviation\ \sigma = \sqrt{\sigma^2}$$

47

Continuous child variables

- Need one *conditional density* function for child variable given
  - continuous parents
  - for each discrete value of parents



$$P(c \mid h, subsidy) = N(a_t h + b_t, \sigma_t^2)(c) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_t h+b_t)}{\sigma_t}\right)^2}$$

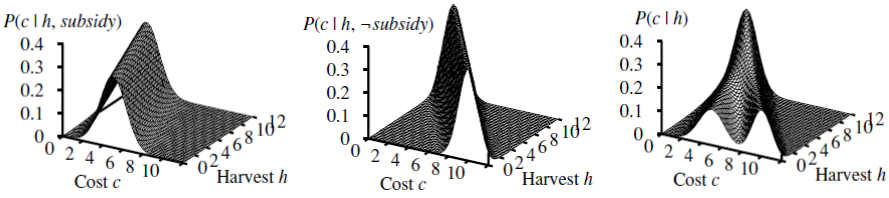
$$P(c \mid h, \neg subsidy) = N(a_f h + b_f, \sigma_f^2)(c) = \frac{1}{\sigma_f\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_f h+b_f)}{\sigma_f}\right)^2}$$

- Mean **Cost** varies *linearly* with **Harvest**, variance fixed
- Linear variation is unreasonable over the full range but works if the likely range of **Harvest** is narrow

48

Continuous child variables

- Determine a Gaussian for *subsidy* and ¬*subsidy*
- What happens if subsidy is not given P(c|h)?



All-continuous network with LG distributions
⇒ full joint distribution is a multivariate Gaussian

Discrete+continuous LG network is a conditional Gaussian network i.e., a multivariate Gaussian over all continuous variables for each combination of discrete variable values

49

Discrete variabel cont. parents

- Probability of Buys given Cost should be a soft threshold

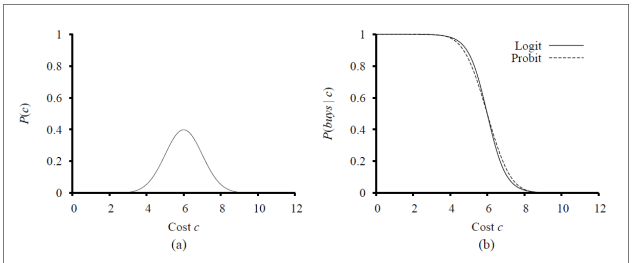


Figure 14.7 **FILES:** . (a) A normal (Gaussian) distribution for the cost threshold, centered on $\mu = 6.0$ with standard deviation $\sigma = 1.0$. (b) Logit and probit distributions for the probability of *buys* given *cost*, for the parameters $\mu = 6.0$ and $\sigma = 1.0$.

Use integral $\Phi(x) = \int^{x} N(0,1)(x)dx$

Leads to $P(buys \mid Cost = c) = \Phi((-c + \mu)/\sigma)$ Probit

Alternativ $P(buys \mid Cost = c) = \dfrac{1}{1 + exp(-2\frac{-c+\mu}{\sigma})}$. Logit

50

Inference tasks

- **Simple queries**: $P(X_1,\ldots, X_n|e_2, e_4, e_5)$
- **Optimal decisions**: decision networks include utility information; inference must handle utility nodes.
- **Value of information**: which evidence to seek next?
- **Sensitivity analysis**: which probability values are more critical?
- **Explanation**: why do I need a new engine?

51

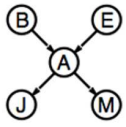
Inference by enumeration

$P(b|j,m) = \alpha\, P(b,j,m)$

$= \alpha \Sigma_a \Sigma_e P(b \wedge j \wedge m \wedge a \wedge e)$ [marginalization]

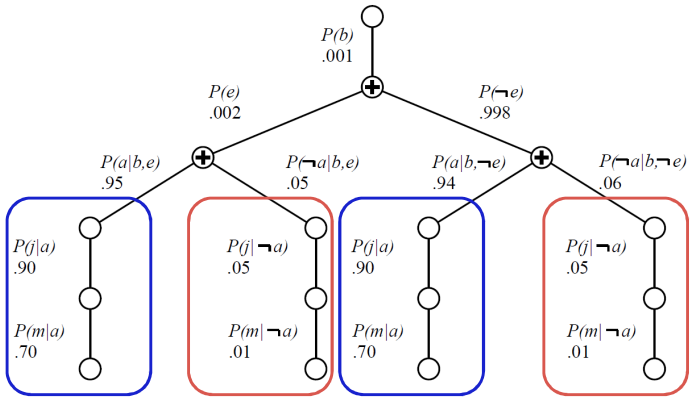
$= \alpha \Sigma_a \Sigma_e P(b)P(e)P(a|b,e)P(j|a)P(m|a)$ [BN]

$= \alpha P(b)\Sigma_e P(e)\Sigma_a P(a|b,e)P(j|a)P(m|a)$ [re-ordering]



52

Evaluation Tree



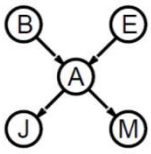
Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e

53

Irrelevant variables

Consider the query $P(JohnCalls|Burglary = true)$

$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(J|a) \sum_m P(m|a)$$



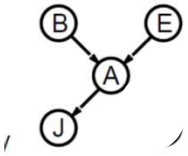
*What about **M**?*
*We sum over all possible values of **m***
For each row it means that the value is 1

62

Irrelevant variables

Consider the query $P(JohnCalls|Burglary=true)$

$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(J|a) \sum_m P(m|a)$$

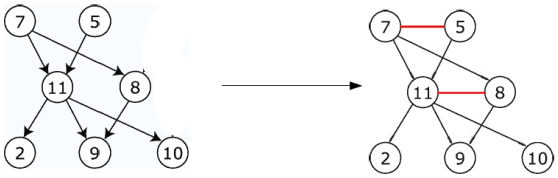


For each row it means that the value is 1

63

Moral Graph: Markov Blanket

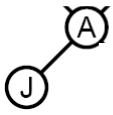
- The moral graph is an undirected graph that is obtained as follows:
  - connect all parents of all nodes
  - make all directed links undirected
- Note:
  - the moral graph connects each node to all nodes of its **Markov blanket**
    - it is already connected to parents and children
    - now it is also connected to the parents of its children



64

Irrelevant variables continued:

- m-separation:
  - A is m-separated from B by C iff it is separated by C in the moral graph
- Example:
  - J is m-separated from E by A



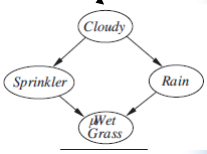
Theorem 2: *Y* is irrelevant if it is m-separated from *X* by *E*

- Example:
  For P(JohnCalls| Alarm=true),
  Burglary, Earthquake and MarryCalls are irrelevant.

65

Approximate Inference In Bayesian Network

- Singly connected networks (or polytrees):
  - any two nodes are connected by at most one (undirected) path
  - time and space cost of variable elimination linear in the size of the network (number of CPT entries; number of parents $O(d^k n)$).
- Multiply connected networks: NP-hard
- We need **approximate inference techniques**!!!!!!!
- Monte Carlo algorithm
  - Widely used to estimate quantities that are difficult to calculate exactly
  - Randomized sampling algorithm
  - Accuracy depends on the number of samples
  - Two families
    - Direct sampling
    - Markov chain sampling



66

Inference by stochastic simulation

Basic idea:

1) Draw N samples from a sampling distribution S
2) Compute an approximate posterior probability $\hat{P}$
3) Show this converges to the true probability P

Outline:

- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

67

Sampling from empty network

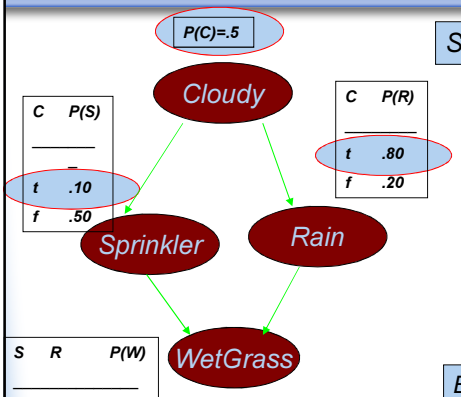
- Generating samples from a network that has no evidence associated with it (*empty* network)
- Basic idea
  - sample a value for each variable in topological order
  - using the specified conditional probabilities

function PRIOR-SAMPLE(bn) **returns** an event sampled from bn
  inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for $i = 1$ **to** n **do**
    $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
      given the values of $Parents(X_i)$ in $\mathbf{x}$
  return $\mathbf{x}$

68

Example in simple case



Sampling

[Cloudy, Sprinkler, Rain, WetGrass]

[true, , ,]

[true, false, ,]

[true, false, true,]

[true, false, true, true]

Estimating

N = 1000
N(Rain=true) = N([_ , _ , true, _]) = 511
P(Rain=true) = 0.511

69

Properties

Probability that PRIORSAMPLE generates a particular event

$$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \ldots x_n)$$

i.e., the true prior probability

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let $N_{PS}(x_1 \ldots x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$

Then we have

$$\begin{aligned} \lim_{N\to\infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

That is, estimates derived from PRIORSAMPLE are consistent

Shorthand: $\hat{P}(x_1, \ldots, x_n) \approx P(x_1 \ldots x_n)$

70

Rejection Sampling

- Used to compute conditional probabilities
- Procedure
  - Generating sample from prior distribution specified by the Bayesian Network
  - Rejecting all that do not match the evidence
  - Estimating probability

71

Rejection Sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
  local variables: N, a vector of counts over X, initially zero
  for j = 1 to N do
      x ← PRIOR-SAMPLE(bn)
      if x is consistent with e then
          N[x] ← N[x]+1 where x is the value of X in x
  return NORMALIZE(N[X])
```

72

Rejection Sampling
Example

- Let us assume we want to estimate P(Rain|Sprinkler = true) with 100 samples
- 100 samples
  - 73 samples => Sprinkler = false
  - 27 samples => Sprinkler = true
    - 8 samples => Rain = true
    - 19 samples => Rain = false
- P(Rain|Sprinkler = true) = NORMALIZE({8,19}) = {0.296,0.704}
- The true answer ist <0.3,0.7>
- Problem
  - It rejects too many samples

73

Analysis of rejection sampling

$$\begin{aligned}\hat{\mathbf{P}}(X|\mathbf{e}) &= \alpha \mathbf{N}_{PS}(X,\mathbf{e}) && \text{(algorithm defn.)}\\ &= \mathbf{N}_{PS}(X,\mathbf{e})/N_{PS}(\mathbf{e}) && \text{(normalized by } N_{PS}(\mathbf{e})\text{)}\\ &\approx \mathbf{P}(X,\mathbf{e})/P(\mathbf{e}) && \text{(property of PRIORSAMPLE)}\\ &= \mathbf{P}(X|\mathbf{e}) && \text{(defn. of conditional probability)}\end{aligned}$$

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if $P(\mathbf{e})$ is small

$P(\mathbf{e})$ drops off exponentially with number of evidence variables!

75

Likelihood Weighting

- Goal
  - Avoiding inefficiency of rejection sampling
- Idea
  - Generating only events consistent with evidence
  - Each event is weighted by likelihood that the event accords to the evidence

76

Likelihood weighting

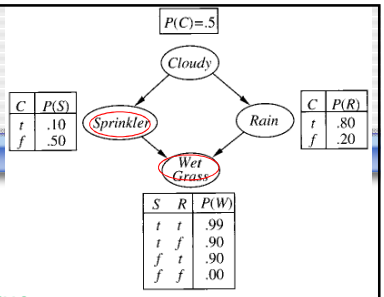
Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

function LIKELIHOOD-WEIGHTING($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{W}$, a vector of weighted counts over X, initially zero
  for $j = 1$ to N **do**
    $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE(bn ,e)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}[X]$)

function WEIGHTED-SAMPLE($bn, \mathbf{e}$) **returns** an event and a weight
  $\mathbf{x} \leftarrow$ an event with n elements; $w \leftarrow 1$
  for $i = 1$ **to** n **do**
    if X_i has a value x_i in $\mathbf{e}$
      then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}, w$

77

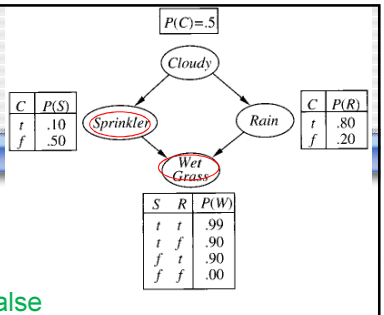
Likelihood Weighting *Example*



- **P**(*Rain*|*Sprinkler*=*true*, *WetGrass* = *true*)?
- Sampling, start with weight=1
  - Sample from **P**(*Cloudy*) = {0.5,0.5} => true
  - *Sprinkler* is an evidence variable with value *true*
    w ← *w* * *P*(*Sprinkler*=*true* | *Cloudy* = *true*) = 0.1
  - Sample from P(*Rain*|*Cloudy*=*true*)={0.8,0.2} => *true*
  - *WetGrass* is an evidence variable with value *true*
    w ←*w* * *P*(*WetGrass*=*true* |*Sprinkler*=*true*, *Rain* = *true*) = 0.099
  - [true, true, true, true] with weight 0.099

78

Likelihood Weighting *Example*

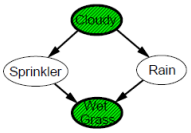


- **P**(*Rain*|*Sprinkler*=*true*, *WetGrass* = *true*)?
- Sampling, start with weight=1
  - Sample from **P**(*Cloudy*) = {0.5,0.5} => false
  - *Sprinkler* is an evidence variable with value *true*
    w ← *w* * *P*(*Sprinkler*=*true* | *Cloudy* = *false*) = 0.5
  - Sample from P(*Rain*|*Cloudy*= *false*)={0.2,0.8} => false
  - *WetGrass* is an evidence variable with value *true*
    w ←*w* * *P*(*WetGrass*=*true* |*Sprinkler*=*true*, *Rain* = *false*) = 0.45
  - [true, true, true, true] with weight 0.45
- Estimating
  - Accumulating weights to either Rain=true or Rain=false
  - Normalize

79

Likelihood analysis

Sampling probability for WEIGHTEDSAMPLE is
$S_{WS}(\mathbf{z}, \mathbf{e}) = \Pi_{i=1}^{l} P(z_i | parents(Z_i))$
Note: pays attention to evidence in **ancestors** only
$\Rightarrow$ somewhere "in between" prior and posterior distribution



Weight for a given sample $\mathbf{z}, \mathbf{e}$ is
$w(\mathbf{z}, \mathbf{e}) = \Pi_{i=1}^{m} P(e_i | parents(E_i))$

Weighted sampling probability is
$S_{WS}(\mathbf{z}, \mathbf{e}) w(\mathbf{z}, \mathbf{e})$
$= \Pi_{i=1}^{l} P(z_i | parents(Z_i)) \ \Pi_{i=1}^{m} P(e_i | parents(E_i))$
$= P(\mathbf{z}, \mathbf{e})$ (by standard global semantics of network)

Hence likelihood weighting returns consistent estimates
but performance still degrades with many evidence variables
because a few samples have nearly all the total weight

81

Markov Chain Monte Carlo

- Let's think of the network as being in a particular current state specifying a value for every variable
- MCMC generates each event by making a random change to the preceding event
- The next state is generated by randomly sampling a value for one of the non evidence variables X_i, **conditioned on the current values of the variables in the MarkovBlanket of X_i**
- Likelihood Weighting only takes into account the evidences of the parents.

82

Gibbs sampling

- Gibbs sampling is a MCMC method
  - State of the network => current assignment
  - Generate next state by sampling one non-evidence variable given Markov blanket
  - Sample each variable in turn (can choose it random)

function GIBBS-ASK(X, **e**, bn, N) **returns** an estimate of **P**(X|**e**)
  local variables: **N**, a vector of counts for each value of X, initially zero
    Z, the nonevidence variables in bn
    x, the current state of the network, initially copied from **e**

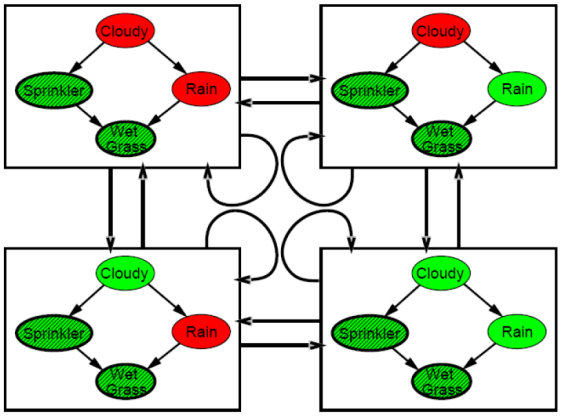
  initialize **x** with random values for the variables in **Z**
  for $j = 1$ to N **do**
    for each Z_i in **Z** **do**
      set the value of Z_i in **x** by sampling from **P**($Z_i|mb(Z_i)$)
      N[x] $\leftarrow$ **N**[x] + 1 where x is the value of X in **x**
  return NORMALIZE(**N**)

Figure 14.16 The Gibbs sampling algorithm for approximate inference in Bayesian networks; this version cycles through the variables, but choosing variables at random also works.

83

Example

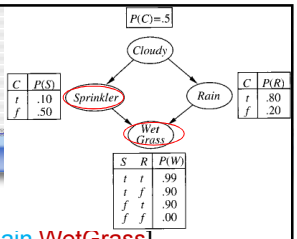
With $Sprinkler = true, WetGrass = true$, there are four states:



Wander about for a while, average what you see

84

Gibbs *Example*



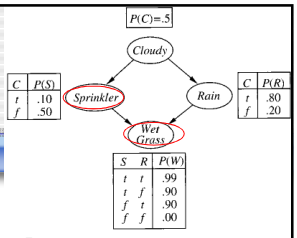
- Query P(Rain| Sprinkler = true, WetGrass = true)
- Initial state is [true, true, false, true] [Cloudy,Sprinkler,Rain,WetGrass]

- The following steps are executed repeatedly:
  - *Cloudy* is sampled, given the current values of its Markov Blanket variables
    So, we sample from *P(Cloudy|Sprinkler= true, Rain=false)*
    The result is Cloudy = false (???????)
  - Now current state is [false, true, false, true] and counts are updated
  - *Rain* is sampled, given the current values of its Markov Blanket variables
    Sample from *P(Rain|Cloudy=false,Sprinkler=true, WetGrass=true)*
    First create the distribution we want to sample from.
    →Rain = true.
  - Current state is [false, true, true, true]

- After all the iterations, let's say the process visited 20 states where rain is true and 60 states where rain is false then the answer of the query is NORMALIZE({20,60})={0.25,0.75}

85

Sample distribution



Want to sample *Cloudy*.
The current state is [Cloudy?, true, false, true]
What is the Markov blanket, the sampling distribution?

evidence *sampled*

P(Cloudy | Sprinkler= true, Rain=false) =

α *P(Cloudy) * P(Sprinkler= true | Cloudy) P(Rain=false | Cloudy)=*
α *(<0.5, 0.5> * <0.1 , 0.5> * <0.2, 0.8>)*
α *(<0.5, 0.5> * <0.1 * 0.2, 0.5*0.8>) =*
α *(<0.5, 0.5> * <0.02 , 0.4>) =*
α *<0.01, 0.2>* $\approx$ *<0.05, 0,95>*

[false, true, false, true] with probability 0,95
[true, true, false, true] with probability 0,05

86

Summary

- Bayesian networks provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = compact representation of joint distribution
- Generally easy for domain experts to construct (if not to big)
- Exact inference by variable elimination
  - polytime on polytrees, NP-hard on general graphs
  - space can be exponential as well
- Approximate inference based on sampling and counting help to overcome complexity of exact inference

87