Fundamentals of AI and KR - Module 3

3. Building Bayesian networks

Paolo Torroni

Fall 2024

Notice

Credits

The present slides are largely an adaptation of existing material, including:

- slides from Russel & Norvig
- slides by Daphne Koller on Probabilistic Graphical Models
- slides by Fabrizio Riguzzi on Data Mining and Analytics

I am especially grateful to these authors.

Downloading and ~~sharing~~

A copy of these slides can be downloaded from virtuale and stored for personal use only. Please do not redistribute.

Table of Contents

Constructing Bayesian networks

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
   - add X_i to the network
   - select parents from $X_1, \ldots, X_{i-1}$ such that $\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$

Every time we add an X, we have to decide if we have to connetct it to X1 or not, it depence by the condition of indipendent

This choice of parents guarantees the global semantics:

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

Xi is the variable that we are adding to the graph

We compare the join probability of some variables

$$= \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i)) \quad \text{(by construction)}$$

Example

Let's construct a Bayesian network that helps us win a prize

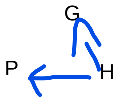
Monty Hall puzzle

We're guests on a TV game show. We stand in front of three closed doors. A prize hides behind one of them. We choose the door on the left. At this point, the host, who knows where prize is, opens the middle door, to reveal it is empty. We are offered to modify our choice. *Should we?*

Guest, Prie, Host are our variables

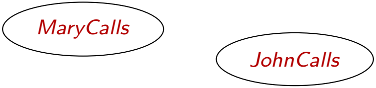
We have to connect G to P,
Does that the coice i make as a guest an influence the position of the prize? The answer is no, so there is no connection, they are indipendent
P(P|G) = P(P)

The host has connection with both G and P
P(H) != P(H|G) because the move of the host depend of where is the price and the choice of the guest



Example

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

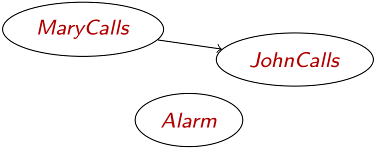


- $P(J|M) = P(J)$?

  No so they are not indipendent

Example

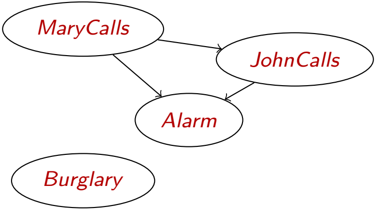
Suppose we choose the ordering M, J, A, B, E



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

Example

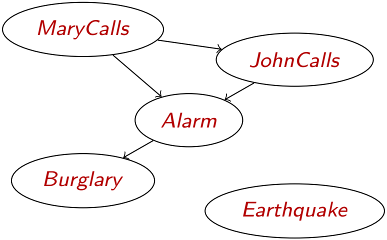
Suppose we choose the ordering M, J, A, B, E



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

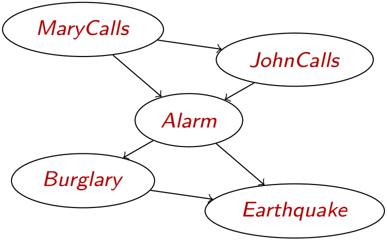
Suppose we choose the ordering M, J, A, B, E



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

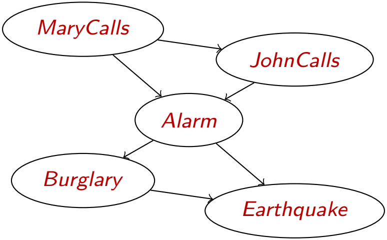
Suppose we choose the ordering M, J, A, B, E



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

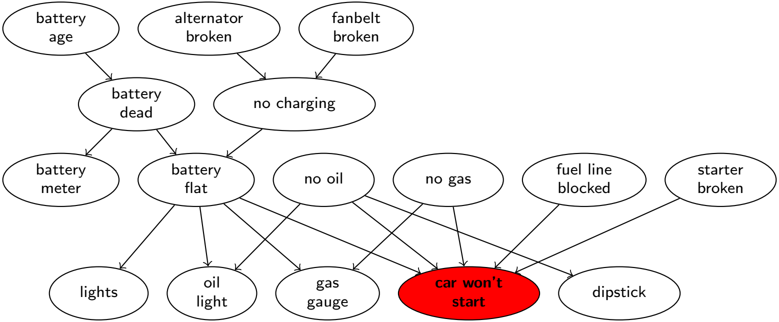


Deciding conditional independence is hard in noncausal directions
(Causal models and conditional independence seem hardwired for humans!)
Assessing conditional probabilities is hard in noncausal directions
Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

Example: Car diagnosis

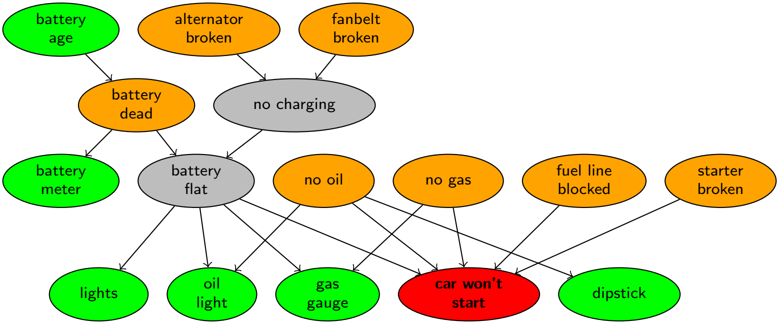
- Initial evidence: car won't start

The nodes in this graph can be divided in different classes



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



Structure learning

If manual design not possible, learn the network from the available data.

Two approaches:

- Constraint-based
  1. **independence test** to identify a set of edge constraints for the graph
  2. search to best satisfy the constraints
- Score-based:
  1. **define a criterion** (score) to evaluate how well the Bayesian network fits the data,
  2. search to maximize score
- Obtaining good quality results may be tricky
- Use domain knowledge if available

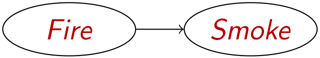
Causal networks

Causal networks

- In principle, any ordering of nodes permits a consistent construction of the network
- However, best to respect order of causality for many reasons, including compactness
- **Causal networks** are a restricted class of Bayesian networks that forbids all but causally compatible orderings

Causal networks

- Consider the following example:



(a)

Causal networks

- Consider the following example:

How we can say which of these 2 network is a casual network? We have to respond to the question, which respond to which?

- Equally good distributions can be defined for (a) and (b)
  - But *are these networks equivalent?*

Not really

Beyond probabilistic dependence: assignment

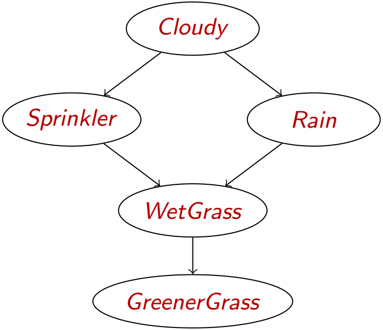
- Causal networks devised to represent causal asymmetries
- Arrow directionality decided based on considerations beyond probabilistic dependence
- The question is: **which responds to which?**
  - Draw an arrow from Fire to Smoke if nature "assigns" a value to Smoke on the basis of what nature learns about Fire
  - Do not draw an arrow from Smoke to Fire if you judge that nature "assigns" a Fire a truth value based on variables other than Smoke
  - For each variable X_i that can take values $x_i = f_i(OtherVariables)$, draw $X_j \to X_i$ if and only if X_j is one of the arguments of f_i

$x_i = f_i(\cdot)$ is called a **structural equation**

they represent phenomena thata are indipendent by the probability of the value, but the link, the dependece is given

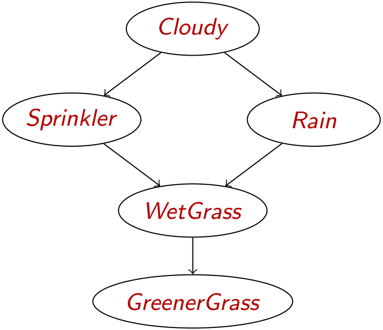
Lawn example

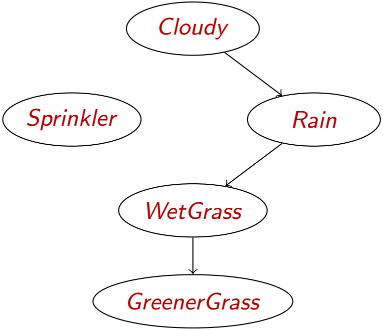
- Structural equations describe mechanism in nature invariant to measurements and local changes in the environment



Lawn example

- Structural equations describe mechanism in nature invariant to measurements and local changes in the environment

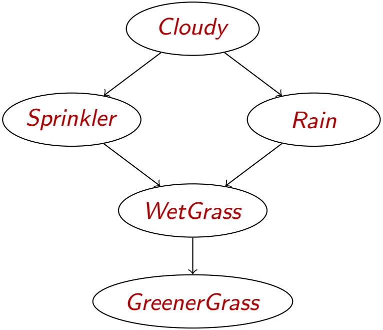


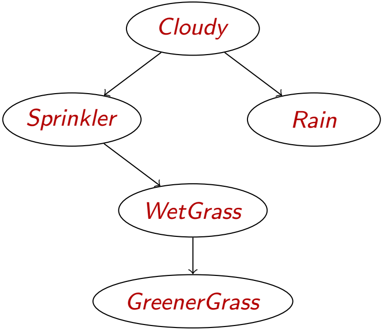


(disabled sprinkler)

Lawn example

- Structural equations describe mechanism in nature invariant to measurements and local changes in the environment



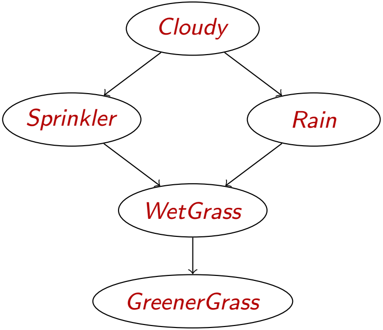


(lawn covered by a tent)

The *do*-operator

In the case of Caual network, instead of only observing we can make also the "if-else" analyisis we can intervent on our network and see the consequences, this is the diffrent between Bayes and casual net.

- Stability is important for representing **interventions** and predicting their observable consequences



- Semantics of Bayes nets:

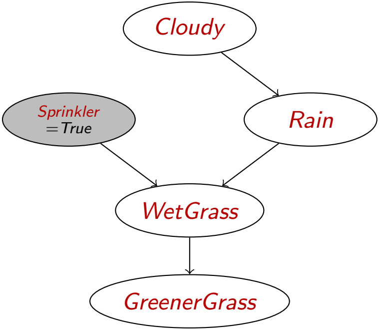
$$P(c, r, s, w, g)$$

- System of structural equations with "U-variables" (**unmodeled variables** or **error terms**):

$$C = f_C(U_C)$$
$$R = f_R(C, U_R)$$
$$S = f_S(C, U_S)$$
$$W = f_W(S, R, U_W)$$
$$G = f_G(W, U_G)$$

The *do*-operator

- Stability is important for representing **interventions** and predicting their observable consequences



do(*Sprinkler* = *True*):
we turn the sprinkler on

- Semantics of Bayes nets:

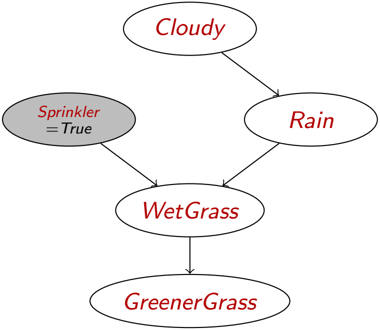
$$P(c, r, w, g \mid do(S = true))$$

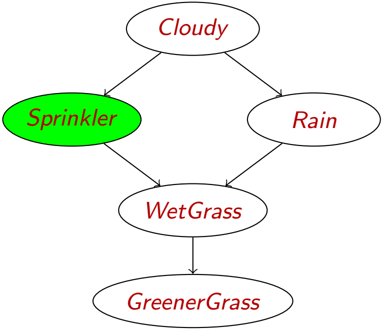
- System of structural equations with "*U*-variables" (**unmodeled variables** or **error terms**):

$$C = f_C(U_C)$$
$$R = f_R(C, U_R)$$
$$S = True$$
$$W = f_W(S, R, U_W)$$
$$G = f_G(W, U_G)$$

The *do*-operator

- What difference?





$P(WetGrass|do(Sprinkler = True))$:
we **turn** the sprinkler on

$P(WetGrass|Sprinkler = True)$:
we **observe** that sprinkler is on

Representing conditional distributions

Compact conditional distributions

CPT grows exponentially with number of parents
CPT becomes infinite with continuous-valued parent or child
Solution: canonical distributions that are defined compactly

Deterministic nodes are the simplest case:
$X = f(Parents(X))$ for some function f

e.g., Boolean functions
$NorthAmerican \Leftrightarrow Canadian \vee US \vee Mexican$

e.g., numerical relationships among continuous variables

$$\frac{\partial Level}{\partial t} = \text{inflow} + \text{precipitation - outflow - evaporation}$$

Compact conditional distributions

Noisy-OR distributions model multiple noninteracting causes

1. Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2. Independent failure probability q_i for each cause alone
   $\Rightarrow P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

Noisy-Or It assumes that each potential caus can indipendently produce the effect and each cause has a probability of being noisy, meaning it might not always lead to the effect when expected
Suppose you have three sympthoms cold flu malaria and a leak nodes that is something of which we dont have a specific ‘value’/reason , and the cause is fever.
If you consider that these reasons for fever, they may or may not cause fever
P(fever| cold) = p1
P(not fever | cold) = 1 - p1 this is called failure probability. If you assume that all of that fail. probability are indipendent, total fail prob would be the product of 1- p1 * 1-p2 * ecc.

Compact conditional distributions

Noisy-OR distributions model multiple noninteracting causes

1. Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2. Independent failure probability q_i for each cause alone
   $\Rightarrow \ P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
F	F	F	0.0	
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		

Compact conditional distributions

Noisy-OR distributions model multiple noninteracting causes

1. Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2. Independent failure probability q_i for each cause alone

$$\Rightarrow P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
F	F	F	0.0	
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		

Number of parameters **linear** in number of parents

Compact conditional distributions

Noisy-OR distributions model multiple noninteracting causes

1. Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2. Independent failure probability q_i for each cause alone
   $\Rightarrow P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
F	F	F	0.0	
F	F	T		**0.1**
F	T	F		**0.2**
F	T	T		$0.02 = 0.2 \times 0.1$
T	F	F		**0.6**
T	F	T		$0.06 = 0.6 \times 0.1$
T	T	F		$0.12 = 0.6 \times 0.2$
T	T	T		$0.012 = 0.6 \times 0.2 \times 0.1$

Fail. probab. of having flu and malaria would be 0.1 * 0.2

Compact conditional distributions

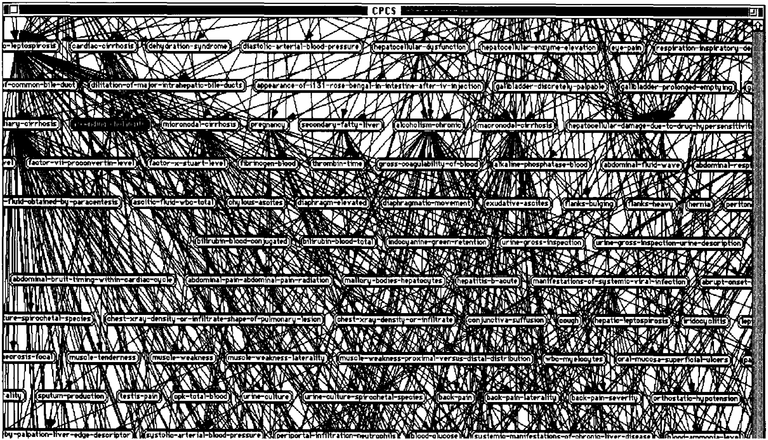
Noisy-OR distributions model multiple noninteracting causes

1. Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2. Independent failure probability q_i for each cause alone
   $\Rightarrow P(X|U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
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	$0.012 = 0.6 \times 0.2 \times 0.1$

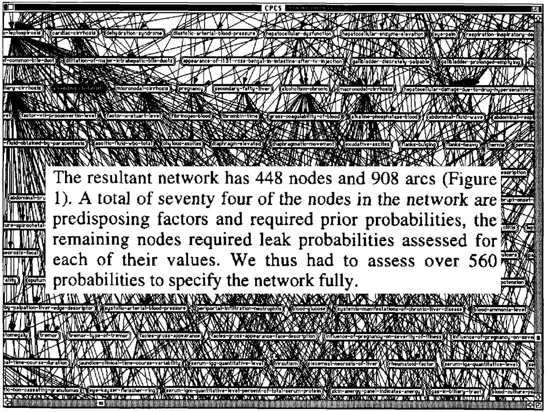
A Bayesian network for internal medicine

Knowledge Engineering for Large Belief Networks, Pradhan et al., UAI 1994



A Bayesian network for internal medicine

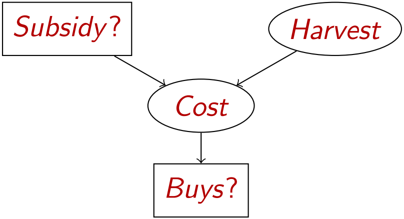
Knowledge Engineering for Large Belief Networks, Pradhan et al., UAI 1994



The resultant network has 448 nodes and 908 arcs (Figure 1). A total of seventy four of the nodes in the network are predisposing factors and required prior probabilities, the remaining nodes required leak probabilities assessed for each of their values. We thus had to assess over 560 probabilities to specify the network fully.

Figure 1. A small portion of the CPCS BN displayed in the Netview visualization program. The node *ascending-cholangitis* in the third row shown in inverse has been selected by the user.

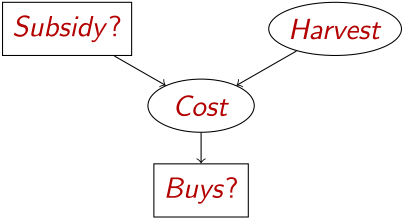
Hybrid (discrete+continuous) networks



In this case the parent of a discrete variable (buys) is a continuos variable (cost).

Hybrid (discrete+continuous) networks

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



- Option 1: discretization—possibly large errors, large CPTs
- Option 2: finitely parameterized canonical families
  - Continuous variable, discrete+continuous parents (e.g., *Cost*)
  - Discrete variable, continuous parents (e.g., *Buys?*)

Continuous child variables

Need one conditional density function for child variable given continuous parents, for each possible assignment to discrete parents

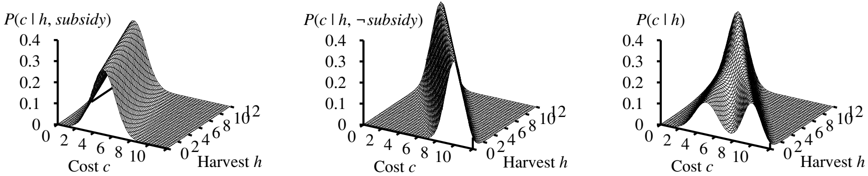
Most common is the linear Gaussian model, e.g.,:

$$P(Cost = c | Harvest = h, Subsidy? = true)$$
$$= \mathcal{N}(a_t h + b_t, \sigma_t)(c) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_t h + b_t)}{\sigma_t}\right)^2}$$

Mean *Cost* varies linearly with *Harvest*, variance is fixed

Linear variation is unreasonable over the full range but works OK if the **likely** range of *Harvest* is narrow

Continuous child variables



The last figure is the sum of the other two

All-continuous network with LG distributions

$\Rightarrow$ full joint distribution is a multivariate Gaussian

Discrete parents added to continuous variables

$\Rightarrow$ LG network is a conditional Gaussian network

i.e., a multivariate Gaussian over all continuous variables for each combination of discrete variable values

e.g., $\mathbf{P}(Cost|Harvest)$ obtained by summing over subsidy cases

Discrete variable w/ continuous parents

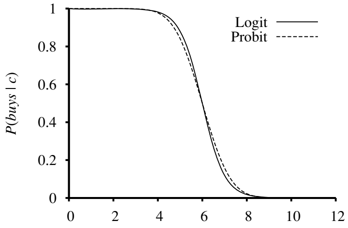
Probability of *Buys?* given *Cost* should be a "soft" threshold

Probit or sigmoid (logit) distribution

e.g., sigmoid: $P(Buys? = true \mid Cost = c) = \frac{1}{1+exp(-2\frac{-c+\mu}{\sigma})}$

Probit uses integral of Gaussian

Sigmoid has similar shape to probit
but much longer tails

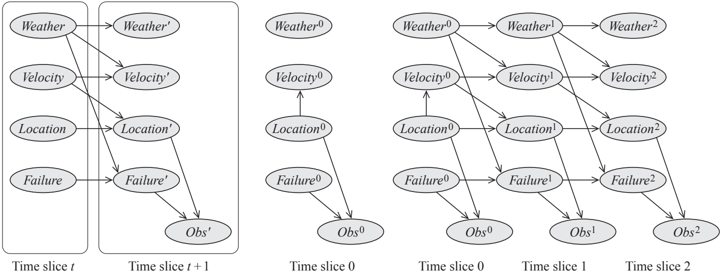


Template-based representations

Especially useful for reasoning about world that evolves over time

System state at time t, $\mathcal{X}^{(t)}$ $X_i \in \mathcal{X}$ is a template variable, instantiated at each time. $X_i^{(t)}$ is a variable. Each "possible world" is a trajectory

Goal: to represent a joint distribution over trajectories



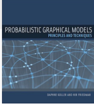
(a) $\mathcal{B}_{\rightarrow}$ (b) $\mathcal{B}_0$ (c) DBN unrolled over 3 steps

Density estimation

Parameters (conditional distributions) can be elicited by domain experts, imposed by the environment or model, or learned from experience

- Many methods for learning distributions from data (**density estimation**). Idea is:
  - Data are **evidence**,
  - **Hypotheses** are probabilistic theories about the domain
- **Bayesian learning** calculates the probability of each hypothesis, given the data
  - Predictions made using *all* hypotheses, weighted by their probabilities
  - Learning is probabilistic inference
  - Optimal prediction, but computationally expensive process
- Approximations: **MAP hypothesis** (keeps only one theory) and **maximum-likelihood hypothesis** (assumes uniform prior)
- **EM** algorithm for learning with hidden variables

See Russel & Norvig, 4th Ed, Chapter 21 *Learning Probabilistic Models*

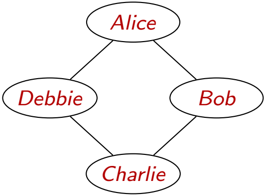


Undirected Graphical Models

Another type is the markov network where you dont have directionality In the markov network factors represent compatibility functions (or "affinity") between connected variables. These functions measure how compatible different variable states are with one another, rather than defining exact conditional probabilities.

Bayesian networks are a type of Probabilistic Graphical Models (PGM)
Another class of PGMs are Markov networks

- undirected
- factors, not CPT (do not represent probability distributions)
- naturally capture conditional independence relations



$\phi_1(A,B)$			$\phi_2(B,C)$			$\phi_3(C,D)$			$\phi_4(D,A)$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

compatibility factors denoting affinity

Misconception network

Summary so far

- Bayes nets provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = compact representation of joint distribution
- Generally easy for (non)experts to construct
- Canonical distributions (e.g., noisy-OR) = compact representations
- Continuous variables $\Rightarrow$ parameterized distributions (e.g., linear Gaussian)
- Bayes nets capture probabilistic influences, whereas causal networks capture causal relations and allow predicting effects of interventions
- Bayes nets instance of larger class of Probabilistic Graphical Models
- Various methods for learning probabilistic model structure and parameters

Suggested exercises from Russel & Norvig, 3^{rd} Ed.

- 14.8 (car diagnosis), (a)-(d)
- 14.11 (nuclear plant)
- 14.12 (telescope problem)
- 14.14 (wrongful conviction)

Questions?