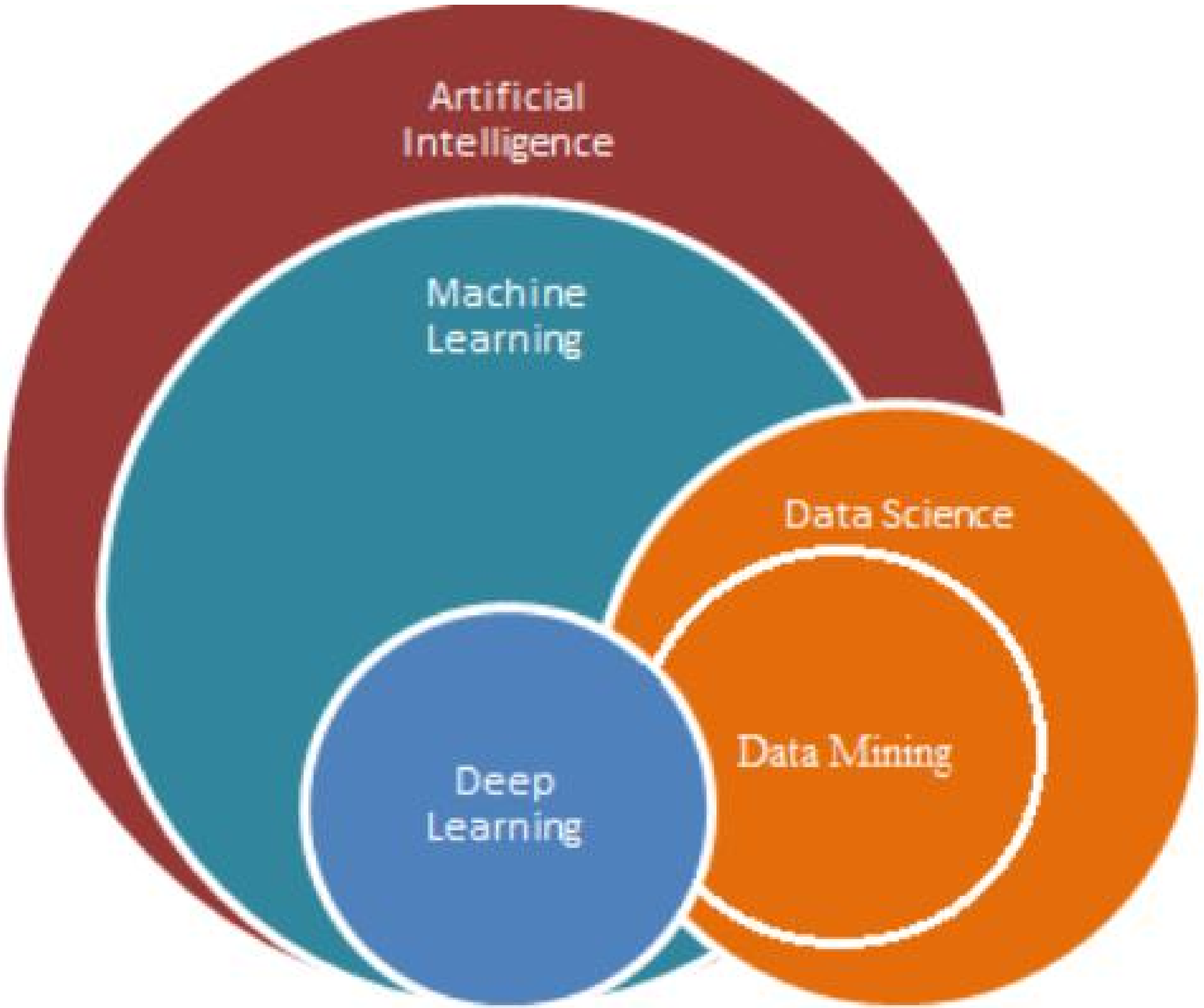
CES -161 - Modelos Probabilísticos em Grafos

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac Sala 110, IEC-ITA

Outlook

A “Reasonable” Graph Representation of Intersections of Related Areas to AI



E Modelos Probabilísticos em Grafos?

- Desafios enfrentados por IA
  - Resolução de Problemas
  - Conhecimento: Raciocínio e planejamento
  - **Incerteza: Conhecimento e Raciocínio**
  - **Aprendizado**
  - Comunicação, percepção e Ação

Agentes

- Um agente é tudo que pode ser considerado capaz de *perceber seu **ambiente*** por meio de ***sensores*** e de *agir sobre esse ambiente* por intermédio de ***atuadores***.
- **Exemplos:** agente animal, agente robótico, agente de software, termostatos….

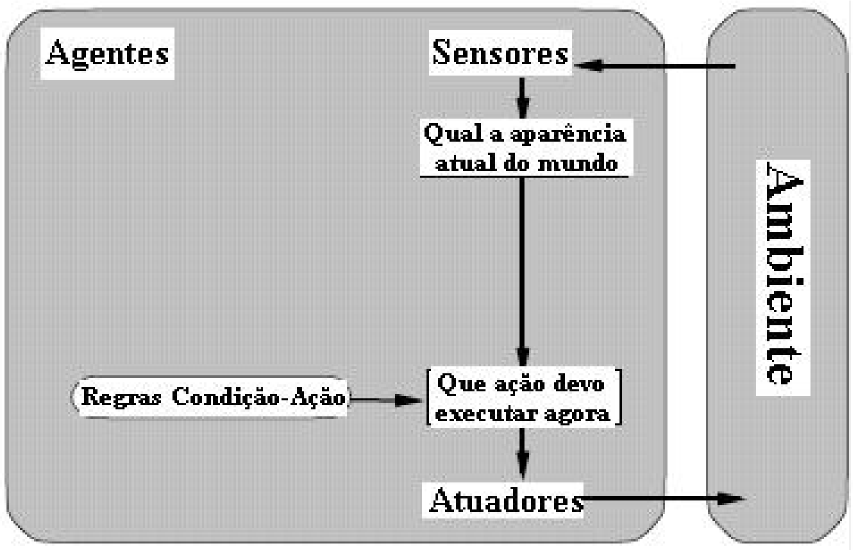


Diagrama esquemático de um agente reativo simples.

Ambiente: Onde os agentes vivem e atuam

- <u>Propriedades dos Ambientes</u>
- Observável x Parcialmente Observável
- Determinístico x Estocástico
- Episódico x Seqüencial
- Estático x Dinâmico
- Discreto x Contínuo
- Agente Único x Multiagente

Uncertainty (Partially observed or stochastic) environments?

1. **Ignorance.** The limits of our knowledge lead us to be uncertain about many things. Does our poker opponent have a flush or is she bluffing?
2. **Physical randomness or indeterminism.** Even if we know everything that we might care to investigate about a coin and how we impart spin to it when we toss it, there will remain an inescapable degree of uncertainty about whether it will land heads or tails when we toss it. A die-hard determinist might claim otherwise, that some unimagined amount of detailed investigation might someday reveal which way the coin will fall; but such a view is for the foreseeable future a mere act of scientistic faith. We are all practical indeterminists.
3. **Vagueness.** Many of the predicates we employ appear to be vague. It is often unclear whether to classify a dog as a spaniel or not, a human as brave or not, a thought as knowledge or opinion.

Revisão de Conceitos Básicos de Probabilidade

P(A | K) – probabilidade condicional ou posterior.
Crença em *A*, dado o corpo de informação *K.*

P(A) – probabilidade *a priori*: Crença em A, na falta de informação adicional proveniente de K.

Uma Variável aleatória tem um domínio (conjunto de valores)e associada a cada um a probabilidade de ocorrência daquele valor. Essa função é chamada de distribuição de Probabilidade.
Exemplo:
Variável Tempo = {Sol, Chuva, Nublado}
P*(Tempo) – é uma distribuição de probabilidade*
P*(Tempo) = <0,7;0,2;0,1>*

P*(Tempo=sol) = 0.7*
P(Tempo=chuva) = P(chuva) = 0.2
P*(Tempo=nublado) = P(nublado) = 0.1*

No caso contínuo, usa-se o termo função de densidade de probabilidade. Vamos focar no caso discreto.

O Axioma Básico da Prob. condicional

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Ou:

$$P(A,B) = P(A \mid B)P(B)$$

Corolário:

$$P(A) = \sum_i P(A, B_i)$$

$$P(A) = \sum_i P(A \mid B_i)P(B_i)$$

Regra da Cadeia

Regra da Cadeia:

$$P(E_1, E_2, \ldots E_n) = P(E_n \mid E_{n-1}, \ldots E_2, E_1) \ldots P(E_2 \mid E_1) P(E_1)$$

Prova:

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

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

$$= \ldots$$

$$= \Pi\, \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$$

Inversão Bayesiana (Regra de Bayes)

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)}$$

P(H|e): Probabilidade posterior

P(H): Probabilidade a priori

Por quê a fórmula é importante?

Muitas vezes P(e|H) é fácil de calcular, ao contrário de P(H|e)?

Exemplo.No cassino, um croupier fala 12! Ele jogou os dados ou estava comandando um jogo de roleta?

P(*12*|*dados*), **P**(*12*|*roleta*): fácil de modelar. **P**(*dados*), **P**(*roleta*): fácil, basta ver número de mesas de dado ou roleta no cassino. **P**(*dados*|*12*), **P**(*roleta*|*12*): não é tão fácil estimar . . .

Cause and Effect

- We usually observe an effect and try to identify its cause

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

- However, it is usually easier to determine P(Effect| Cause) than P(Cause | Effect)

Another example: Meningitis

- Let's assume 0.8 of people with Meningitis present stiff neck (S), probability of Meningitis is 1 in 10000 and Stiff neck prob. is 0.1

For assessing diagnostic probability from causal probability:

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

E.g., let M be meningitis, S be stiff neck:

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

Note: posterior probability of meningitis still very small!

Calculating the probability of the evidence

- Suppose we wish to computer the probability of the observed evidence, let's say P(B=b) and A has possible values a_1, ...a_m . We can apply Bayes' rule for each value of A:

$$P(A=a_1|B=b) = P(B=b|A=a_1)P(A=a_1)/P(B=b)$$
$$\ldots$$
$$P(A=a_m|B=b) = P(B=b|A=a_m)P(A=a_m)/P(B=b)$$

- Adding these up:

$$\Sigma_i P(A=a_i|B=b) = \Sigma_i P(B=b|A=a_i)P(A=a_i)/P(B=b)$$

- And noting that $\Sigma_i P(A=a_i|B=b) = 1$, then:

$$P(B=b) = \Sigma_i P(B=b|A=a_i)P(A=a_i)$$

Calculating the probability of the evidence - 2

- Since $P(B=b) = \Sigma_i P(B=b|A=a_i)P(A=a_i)$
- P(B=b) is a normalization factor regards i that we can denote α.
- In vectorial notation, we can write:

$$\mathbf{P}(A|B=b) = \alpha \mathbf{P}(B=b|A)\mathbf{P}(A)$$

Inference from Full joint distributions

Typically, we are interested in
the posterior joint distribution of the <u>query variables</u> $\mathbf{Y}$
given specific values $\mathbf{e}$ for the <u>evidence variables</u> $\mathbf{E}$

Let the <u>hidden variables</u> be $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$

Then the required summation of joint entries is done by summing out the hidden variables:

$$\mathbf{P}(\mathbf{Y}|\mathbf{E}=\mathbf{e}) = \alpha\mathbf{P}(\mathbf{Y},\mathbf{E}=\mathbf{e}) = \alpha\Sigma_{\mathbf{h}}\mathbf{P}(\mathbf{Y},\mathbf{E}=\mathbf{e},\mathbf{H}=\mathbf{h})$$

The terms in the summation are joint entries because $\mathbf{Y}$, $\mathbf{E}$, and $\mathbf{H}$ together exhaust the set of random variables

Obvious problems:
1) Worst-case time complexity $O(d^n)$ where d is the largest arity
2) Space complexity $O(d^n)$ to store the joint distribution
3) How to find the numbers for $O(d^n)$ entries???

d - number of possible elements of variable, n - number of variables

Inference from Full joint distributions - 2

- Inference from Full joint distributions could estimate any conditional probability even when involving hidden variables
- But, it would require a large amount of space to store it and even more data to build such full joint distribution

- Bayesian Network make it easier to build and store distributions

Introdução a Redes Bayesianas

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac Sala 110, IEC-ITA

Sumário

- Interpretação de Probabilidades
- Redes Bayesianas ou Redes de crença
- Inferência probabilística
- Aprendizado em método probabilísticos
- Métodos simplificados: Bayes ingênuo e Noisy-OR

Rede Bayesiana ou Rede de Crença (Belief Network)

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax:

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

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

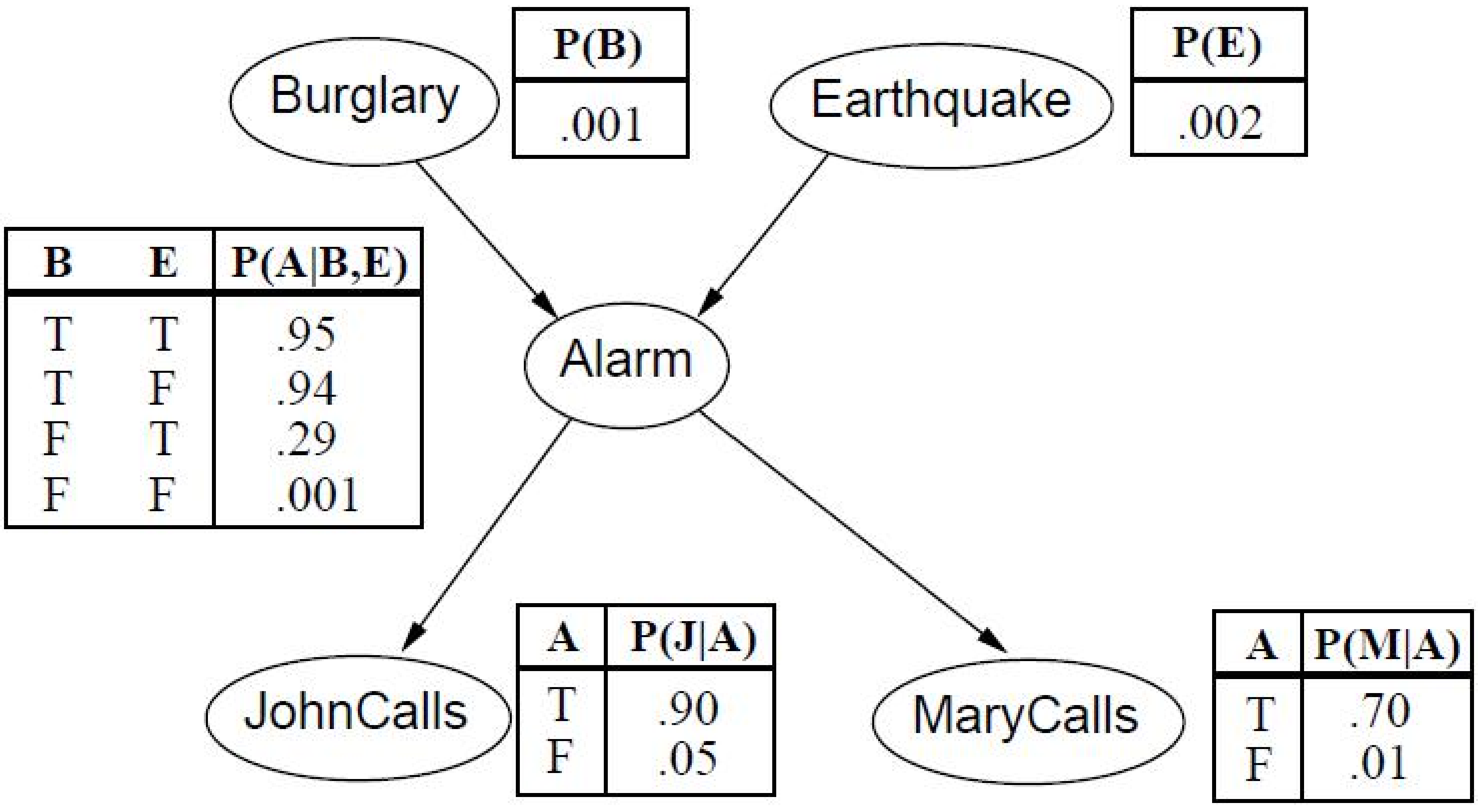
Example: Is it an Earthquake or burglar?

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$
Network topology reflects "causal" knowledge:

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

Example - 2



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Método para construção de uma rede

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

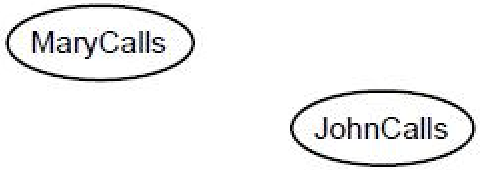
1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$

This choice of parents guarantees the global semantics:

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

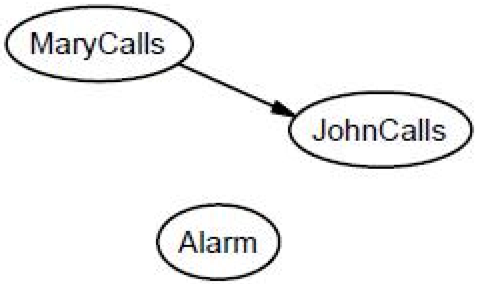
Exemplo

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



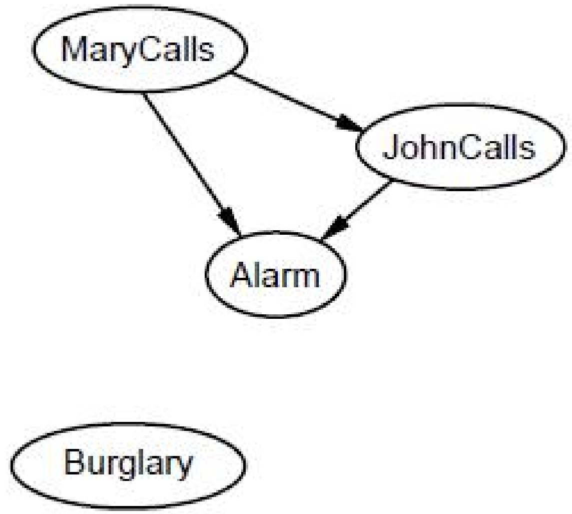
$P(J|M) = P(J)$?

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



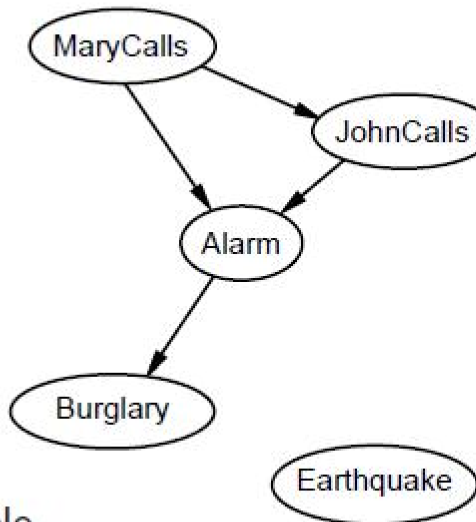
$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$?

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



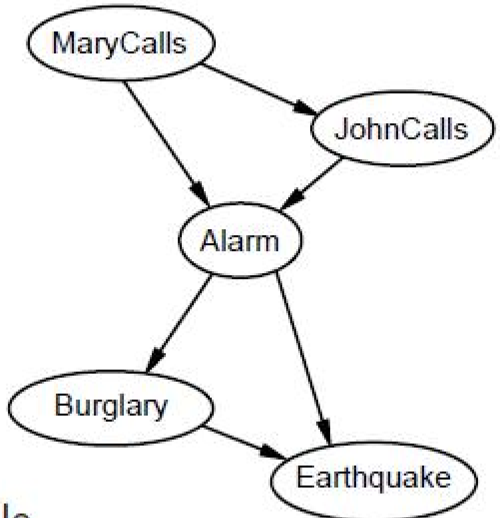
$P(J|M) = P(J)$? No
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No
$P(B|A,J,M) = P(B|A)$?
$P(B|A,J,M) = P(B)$?

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

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



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

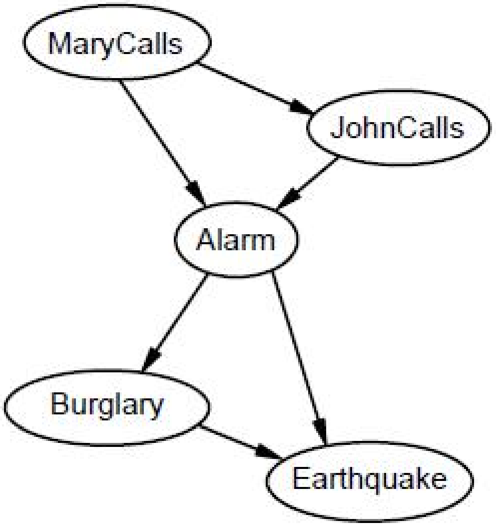
$P(A|J,M) = P(A|J)$? $P(A|J,M) = P(A)$? No

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



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

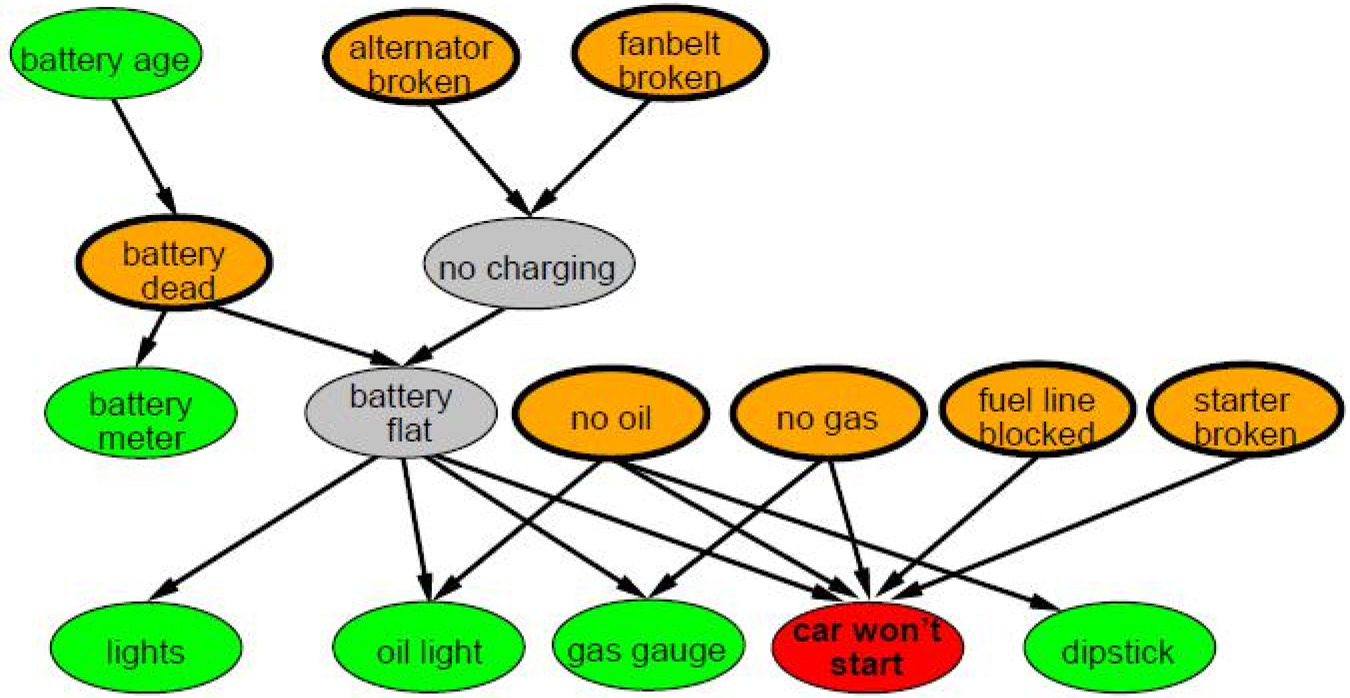
Network is less compact: $1+2+4+2+4=13$ numbers needed

Outro Exemplo: Conserto de Carro

Initial evidence: car won't start

Testable variables (green), "broken, so fix it" variables (orange)

Hidden variables (gray) ensure sparse structure, reduce parameters



I-map and D-map and Perfect Map

- I-map: All direct dependencies in the system being modeled are explicitly shown via arcs. (Independence Map or I-map for short).
- D-map: If every arc in a BN happens to correspond to a direct dependence in the system, then the BN is said to be a Dependence-map (or, D-map for short).
- A BN which is both an I-map and a D-map is said to be a perfect map.

Sumário

- Redes Bayesianas ou Redes de crença
- Inferência probabilística
- Aprendizado em método probabilísticos
- Métodos simplificados: Bayes ingênuo e Noisy-OR

Inferência em Redes Bayesianas

- Dada uma rede, devemos ser capaz de inferir a partir dela isto é :

- Busca responder questões simples, P(X| E=e)
  - Ex. : $P(NoGas|Gauge = empty, Lights = on, Starts = false)$
- Ou questões conjuntivas: P(X_i , X_j | E=e)
  - Usando o fato:

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

- A inferência pode ser feita a partir da distribuição conjunta total ou por **enumeração**

Inferência com Distribuição Conjunta Total: Exemplo

Por exemplo para saber

P(A|b) temos

$P(A|b) = P(A,b)/P(b) =$

A	B	C	P(A,B,C)
F	F	F	P(A=F,B=F,C=F)
F	F	T	P(A=F,B=F,C=T)
F	T	F	-- -
F	T	T	--
T	F	F	--- ----
T	F	T	----
T	T	F	-----
T	T	T	P(A=T,B=T,C=T)

$< P(a, b)/P(b) ; P(\neg a , b)/P(b) > =$

$= \alpha < P(a, b) ; P(\neg a , b) >$

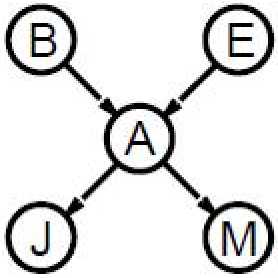
$= \alpha [< P(a,b,c)+P(a,b,\neg c); P(\neg a,b,c)+P(\neg a,b, \neg c) >]$

Observe que α pode ser visto como um fator de normalização para o vetor resultante da distribuição de probabilidade, pedida P(A|b). Assim pode-se evitar seu cálculo, Simplesmente normalizando $< P(a,b); P(\neg a , b) >$

Inferência em Redes Baysianas por Enumeração

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

Simple query on the burglary network:

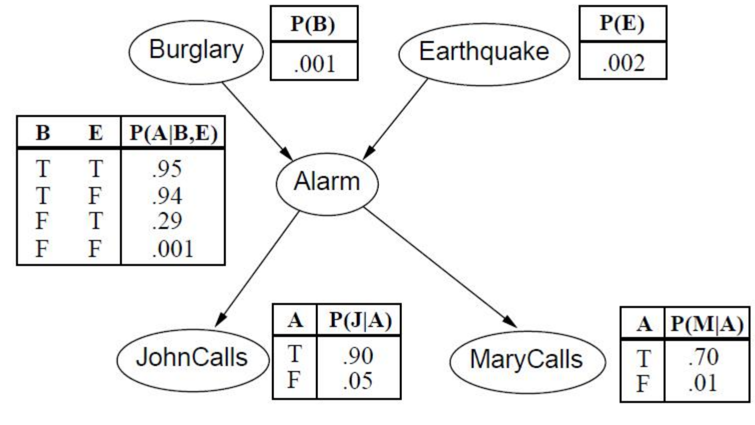


$\mathbf{P}(B|j,m)$
$= \mathbf{P}(B,j,m)/P(j,m)$
$= \alpha\mathbf{P}(B,j,m)$
$= \alpha\Sigma_e\Sigma_a\mathbf{P}(B,e,a,j,m)$

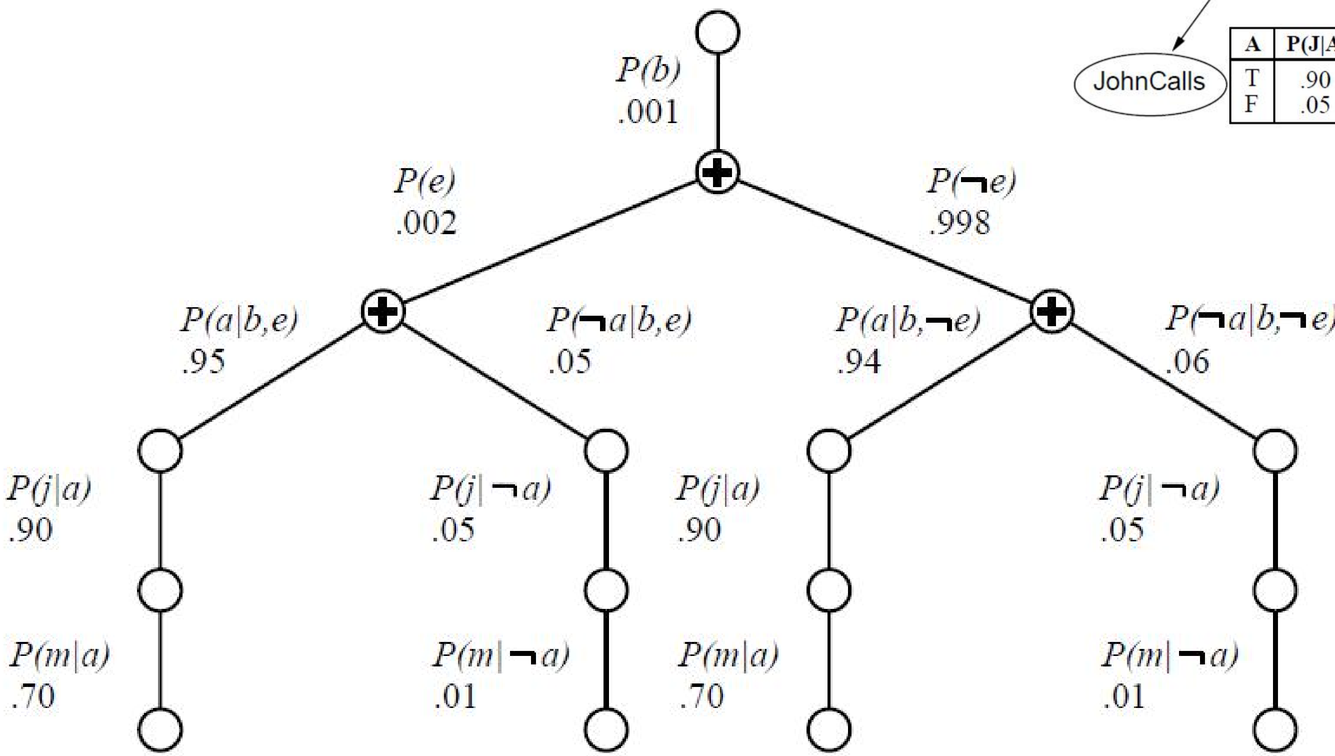
Rewrite full joint entries using product of CPT entries:
$\mathbf{P}(B|j,m)$
$= \alpha\Sigma_e\Sigma_a\mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a)$
$= \alpha\mathbf{P}(B)\Sigma_e P(e)\Sigma_a\mathbf{P}(a|B,e)P(j|a)P(m|a)$

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

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



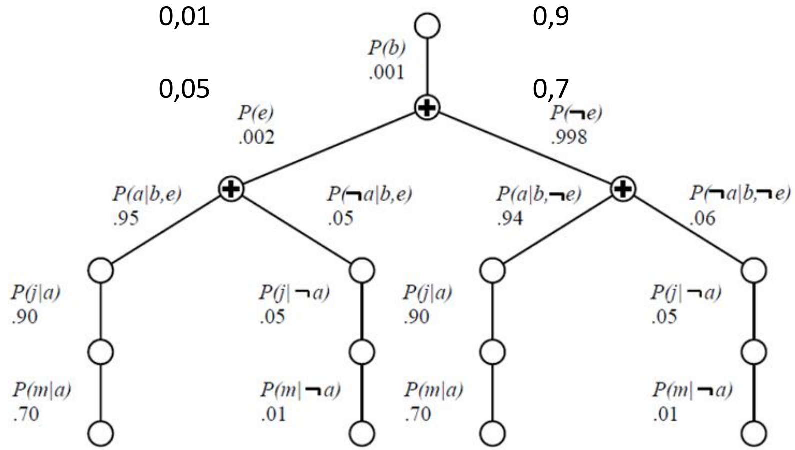
Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e



- Pode ser melhorada através do armazenamento dos valores já calculados (Programação Dinâmica)

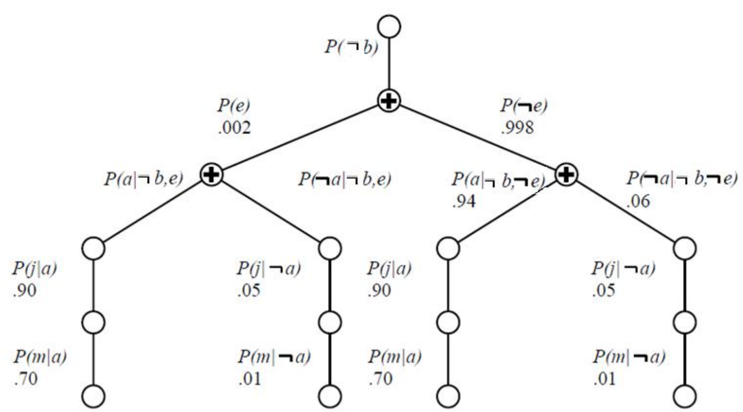
Calculando P(b|j,m) não normalizado

				"P(b) nao normalizado"		
				0,0005922		
				0,001		
	+			0,5922426		
		0,001197			0,591046	
*		0,002	*		0,998	
+		0,598525	+		0,59223	
	0,5985	0,000025		0,5922	0,00003	Produtorio
	0,95	0,05		0,94	0,06	
	0,9	0,01		0,9	0,01	
	0,7	0,05		0,7	0,05	



Calculando P(não b|j,m) não normalizado

			"P(nao b) nao normalizado"	
			0,001492	
			0,999	
	+		0,001493	
		0,000366		0,001127
*		0,002 *		0,998
+		0,183055 +		0,00113
	0,1827	0,000355	0,00063	0,0005
	0,29	0,71	0,001	0,999
	0,9	0,01	0,9	0,01
	0,7	0,05	0,7	0,05



Valores Normalizados P(b|j,m) e P(não b|j,m)

$$P(b \mid j,m) = \frac{0{,}0005922}{0{,}0005922 + 0{,}001492} = 0{,}2841$$

$$P(\neg b \mid j,m) = \frac{0{,}001492}{0{,}0005922 + 0{,}001492} = 0{,}7159$$

Algoritmo de Enumeração

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 $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(bn.VARS, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

---

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

Inferência por Enumeração

- Algoritmo de Enumeração permite determinar uma distribuição de probabilidade condicional
- P(variável de saída| evidências conhecidas)

- Também é possível responder perguntas conjuntivas usando o fato:

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

- Demonstração?….

Demonstração

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

como: $\mathrm{P}(A,B) = \mathrm{P}(A \mid B)\,\mathrm{P}(B)$

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

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

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

Tamanho das Tabelas de Probabilidade Condicional e Distribuição Conjunta Total

- Vamos supor que cada variável é influenciada por no máximo k outras variáveis (Naturalmente, $k<n$=total de variáveis).

- Supondo variáveis binárias, Enquanto, na distribuição conjunta Total haverá 2^n entradas em uma rede haverá no máximo $n*2^k$

- Para uma rede com $n=30$ com no máximo cinco pais ($k=5$) isto significa 960 ou menos (30*2^5) ao invés de mais um bilhão (2^{30})

Sumário

- Redes Bayesianas ou Redes de crença
- Inferência probabilística
- Aprendizado em método probabilísticos
- Métodos simplificados: Bayes ingênuo e Noisy-OR

Aprendizado em modelos probabilísticos

- Aprender em redes bayesianas é o processo de determinar a topologia da rede (isto é, seu grafo direcionado) e as tabelas de probabilidade condicional

- Problemas?
  - Como determinar a topologia?
  - Como estimar as probabilidades ?
  - Quão complexas são essas tarefas?
    - Isto é quantas topologias e quantas probabilidades precisariam ser determinadas….

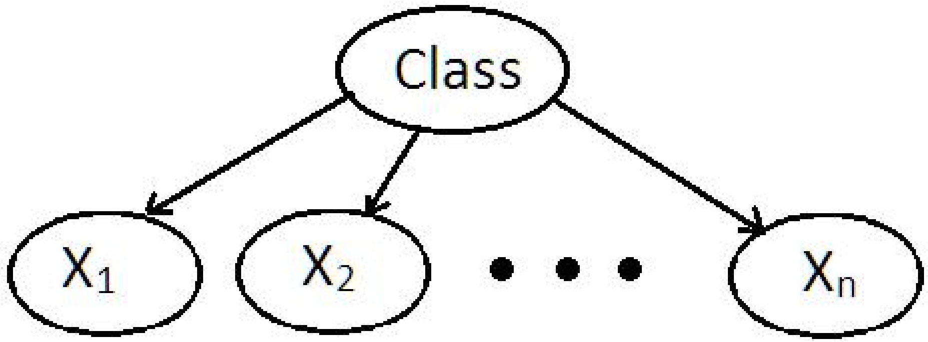
Simplificando a representação tabelas de probabilidade condicional (CPT)

- Vimos que que o número de entradas de uma CPT cresce exponencialmente
  - Para o caso binário e K pais, a CPT de um nó terá 2^k probabilidades a serem calculadas

- Vejamos duas abordagens para simplificar a rede através da adoção de hipóteses simplificadoras
  - Bayes Ingênuo e
  - OU-ruidoso

Naïve Bayes (Bayes Ingênuo)

- Uma classe particular e simples de redes bayesianas é chamada de Bayes Ingênuo (Naïve Bayes)
- Ela é simples por supor independência condicional entre todas as variáveis X dada a variável Class
- As vezes, chamado também de classificador Bayes, por ser frequentemente usado como abordagem inicial para classificação



Naïve Bayes (Bayes Ingênuo) - 2

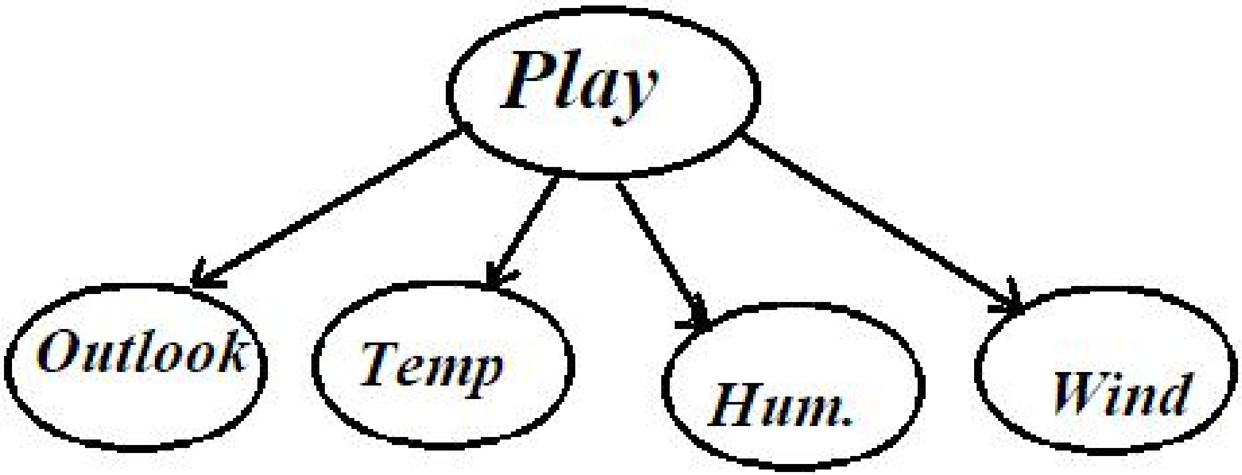
- A topologia simples traz a vantagem da representação concisa da Distribuição Conjunta Total.
- Como todo os nós tem no máximo um pai, cada CPT de no X tem apenas duas entradas e uma entrada no nó classe. Logo, (2n-1) entradas para toda a rede. Naïve Bayes é **linear** em relação ao número de nós (n) !!!!
- "Na prática, sistemas de Bayes ingênuos podem funcionar surpreendentemente bem….". pg. 438

Exemplo de Naïve Bayes

- Vamos retomar o exemplo do jogo de tênis

Ex	Céu	Temperatura	Umidade	Vento	JogarTênis
X1	Ensolarado	Quente	Alta	Fraco	NÃO
X2	Ensolarado	Quente	Alta	Forte	NÃO
X3	Nublado	Quente	Alta	Fraco	SIM
X4	Chuvoso	Boa	Alta	Fraco	SIM
X5	Chuvoso	Fria	Normal	Fraco	SIM
X6	Chuvoso	Fria	Normal	Forte	NÃO
X7	Nublado	Fria	Normal	Forte	SIM
X8	Ensolarado	Boa	Alta	Fraco	NÃO
X9	Ensolarado	Fria	Normal	Fraco	SIM
X10	Chuvoso	Boa	Normal	Fraco	SIM
X11	Ensolarado	Boa	Normal	Forte	SIM
X12	Nublado	Boa	Alta	Forte	SIM
X13	Nublado	Quente	Normal	Fraco	SIM
X14	Chuvoso	Boa	Alta	Forte	NÃO

Usando a abordagem Bayes ingênuo



- Problema a resolver:

Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

Solução:

- P(Play|Outlook, Temp, Hum, Wind)=
- P(Outlook, Temp, Hum, Wind|Play)P(Play)/P(Outlook, Temp, Hum, Wind)=
- Regra da cadeia e indepêndencia:
- P(Outlook|Play)P(Temp|Play)P(Hum|Play)P(Wind|Play)P(Play)/ P(Outlook, Temp, Hum, Wind)

- O método de inferência por enumeração já visto é aplicável!!!
- Estima-se as probabilidades pelo conjunto de treinamento

Contagens e probabilides estimadas pelo conjunto de treinamento

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
~P(Outlook|Play)													
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- P(Play=s|Outlook=sunny,Temp=cool,Hum=high,Wind=true)=

- P(sunny|play)P(cool|play)P(high|play)P(windy|play)P(Play) /P(evidencia) = 2/9*3/9*3/9*3/9*9/14 / P(e) =0.0053/P(e)

Solução 3 - continuação

- Da mesma forma,
- P(sunny|play)P(cool|play)P(high|play)P(windy|play). P(Play)/P(e) = 3/5*1/5*4/5*3/5*5/14/P(e)=0.0206/P(e)
- Mas P(H,e) e P(not H,e) tem que somar 1, assim:

$$\text{Probability of } yes = \frac{0.0053}{0.0053+0.0206} = 20.5\%,$$

$$\text{Probability of } no = \frac{0.0206}{0.0053+0.0206} = 79.5\%.$$

Estimativas de Probabilides

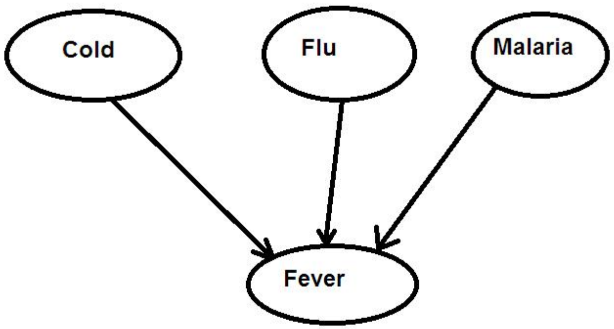
- Qual a estimativa da probabilidade P(Outlook=overcast|Play=no)?

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- Zero! Isto é razoável? Como resolver?
- Uma Solução: estimador de Laplace (Laplace smoothing). Seja V o número de valores possíveis para A, estima-se P(A|B) :
- P(A=a|B=b) = [N(A=a,B=b)+1]/[N(B=b)+V]

Criando Distribuições Condicionais Conjuntas Compactadas....

- Alguns problemas podem ser modelados com uma abordagem do tipo Noisy-OR (ou ruidoso). A técnica parte de duas hipóteses:
  - Todas as causas de uma variável ser acionada estão listadas (pode-se adicionar uma causa geral "outros")
    - Isto é, P (Fever | F,F,F) = 0
  - Há independência condicionais entre oque causa a "falha" da variável pai acionar a variável filho (efeito). Exemplo: o que impede a gripe de causar febre em alguém é independente do que impede o resfriado de causar febre.
    - Isto é, P (not Fever| Cold,Flu,Malaria) = P(not Fever|Cold)P(not Fever| Flu)P(not Fever | Malaria)
- Exemplo:
- P(Not fever |malaria) =0.1
- P(Not fever| flu) =0.2
- P(Not fever| cold)=0.6



Noisy -OR

Noisy-OR distributions model multiple noninteracting causes

1) Parents $U_1 \dots U_k$ include all causes (can add leak node)

2) Independent failure probability q_i for each cause alone

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

- $P(X \mid u_1, \cdots u_j, \neg u_{j+1}, \dots \neg u_k) = \langle 1 - \prod_{i=1}^{j} q_i ;\ \prod_{i=1}^{j} q_i \rangle$
- q_i is the probability of cause i fails !!

Noisy -OR

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

$Cold$	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	$0.012 = 0.6 \times 0.2 \times 0.1$

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

- $P(X \mid u_1, \cdots u_j, \neg u_{j+1}, \ldots \neg u_k) = \langle 1 - \prod_{i=1}^{j} q_i ; \prod_{i=1}^{j} q_i \rangle$
- q_i is the probability of cause i fails !!

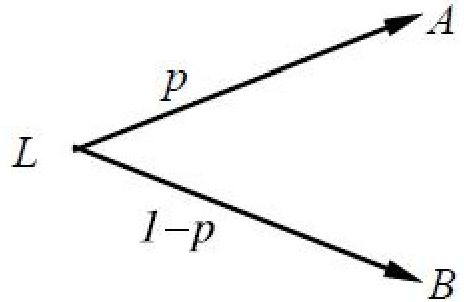
CES -161 - Modelos Probabilísticos em Grafos

[Rational] Decisions with Bayesian Networks

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac
Sala 110, IEC-ITA

Lotteries

An agent chooses among prizes (A, B, etc.) and lotteries, i.e., situations with uncertain prizes



Lottery $L = [p, A;\ (1-p), B]$

Notation:

$A \succ B$	A preferred to B
$A \sim B$	indifference between A and B
$A \succsim B$	B not preferred to A

Preferências Racionais

Idea: preferences of a rational agent must obey constraints.
Rational preferences $\Rightarrow$
behavior describable as maximization of expected utility

Constraints:

Orderability
$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$
Transitivity
$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$
Continuity
$$A \succ B \succ C \Rightarrow \exists p \; [p, A;\; 1-p, C] \sim B$$
Substitutability
$$A \sim B \Rightarrow [p, A;\; 1-p, C] \sim [p, B; 1-p, C]$$
Monotonicity
$$A \succ B \Rightarrow ([p, A;\; 1-p, B] \succsim [q, A;\; 1-q, B] \Leftrightarrow \; p \geq q)$$

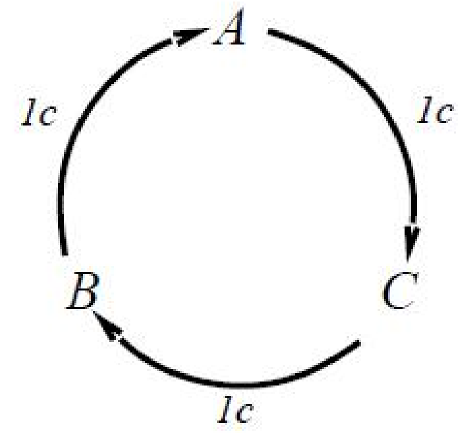
Violação das Restrições leva a "Irracionalidade"

For example: an agent with intransitive preferences can be induced to give away all its money

If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B

If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A

If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



Maximizing Expected Utility (MEU)

Theorem (Ramsey, 1931; von Neumann and Morgenstern, 1944):
Given preferences satisfying the constraints
there exists a real-valued function U such that

$$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$$
$$U([p_1, S_1; \ \ldots \ ; \ p_n, S_n]) = \Sigma_i \ p_i U(S_i)$$

MEU principle:

Choose the action that maximizes expected utility

Note: an agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities

E.g., a lookup table for perfect tictactoe

Estimando Utilidades

Utilities map states to real numbers. Which numbers?

Standard approach to assessment of human utilities:

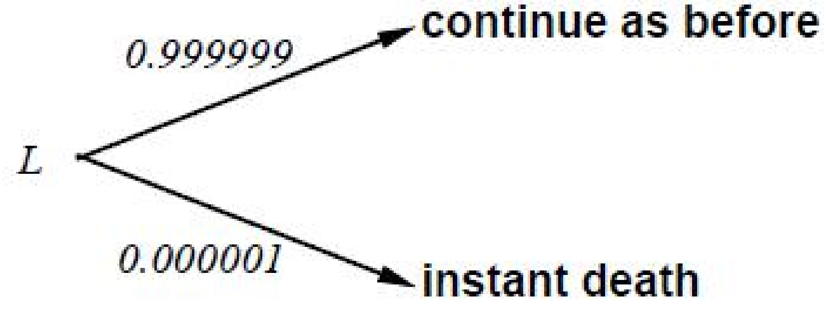
compare a given state A to a standard lottery L_p that has

"best possible prize" $u_\top$ with probability p

"worst possible catastrophe" $u_\bot$ with probability $(1-p)$

adjust lottery probability p until $A \sim L_p$

pay \$30 ~



Definindo Funções de Utilidades através de loterias

- Dado o intervalo [0,1] entre a "pior catastrofe possível" e "o melhor prêmio possível", ao encontrar uma loteria [p,1;1-p,0] que seja indiferente a uma situação S o número p é a utilidade de S

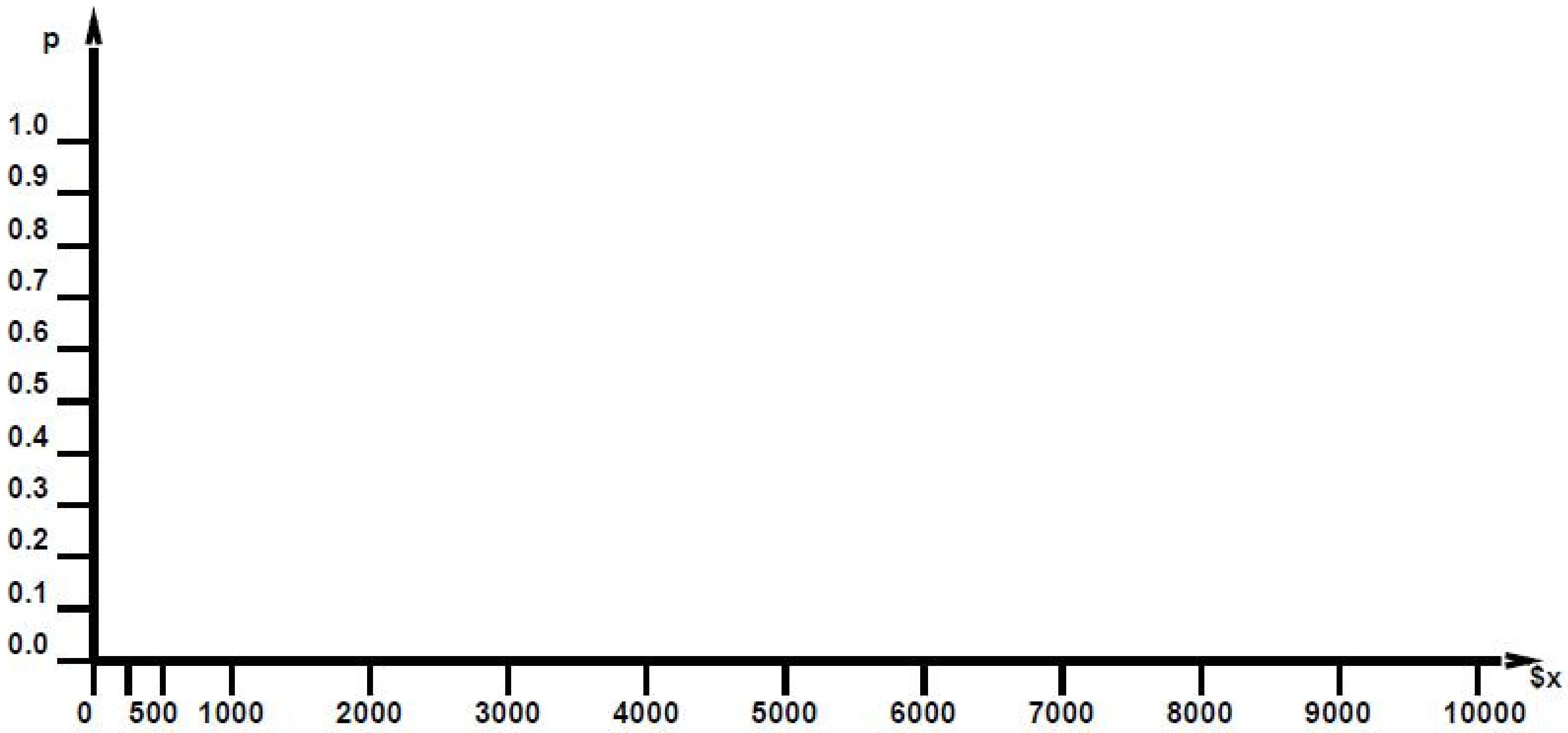
- Em ambientes, com prêmios determinísticos pode-se apenas estabelecer a ordem de preferências, nesse caso usa-se o termo utilidades ordinais

- Funções de utilidades ordinais podem ser chamadas de funções de valor e são invariantes para qualquer transformação monotônica

Utilidade do dinheiro

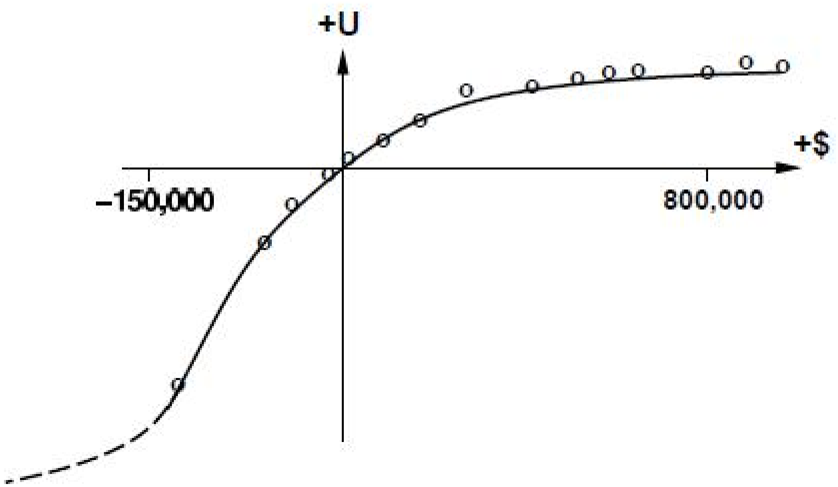
- Preferências de um grupo sobre dinheiro certo (x) e loteria [p,M;1-p,0]

For each x, adjust p until half the class votes for lottery (M=10,000)



Dinheiro vs Utilidade

- Dinheiro não tem uma relação linear (ou simples) com utilidade!
- Ao estimar a utilidade em vários experimentos, observa-se que dada uma loteria L com valor esperado EMV(L) tem-se $U(L) < U(EMV(L))$, isto é as pessoas são aversas a risco
- Um gráfico típico de dinheiro ($) vs Utilidade (U):



The Saint Petersburg Paradox

- The paradox is named from Daniel Bernoulli's presentation of the problem and his solution, published in 1738 in St. Petersburg

- A casino offers a game of chance for a single player in which a fair coin is tossed at each stage. The pot starts at 1 dollar and is doubled every time a head appears. The first time a tail appears, the game ends and the player wins whatever is in the pot.

- Thus the player wins 1 dollar if a tail appears on the first toss, 2 dollars if a head appears on the first toss and a tail on the second.
- Two questions:
  - How much would you accept to pay for playing this game?
  - What is the expected monetary value of the game?

Escolha A ou B

- A: 80% de chance de ganhar $4000
- B: 100% de chance de ganhar $3.000

Escolha C ou D

- C: 20% de chance de ganhar $4000
- D: 25% de chance de ganhar $3.000

Supondo U(0)=0

- Se maioria escolhe B em detrimento de A e C em detrimento de D,
  - De A e B, temos que $0,8*U(4000)<U(3000)$
  - De C e D temos que $0,8U(4000)>U(3000)$
- Contraditório!!!!

Teorias alternativas

- Em linhas gerais as pessoas divergem da teoria da maximização da utilidade esperada em situações de probabilidade muito alta e/ou muito baixa

- Há algumas teorias alternativas que se propõem a descrever o comportamento humano real. Uma das mais relevantes foi proposta por Kahneman e Tversky. Esta teoria propõe um modelo alternativo que descreve esse efeito "certeza" e outros

Decisões [Racionais] com Redes Bayesianas

- Preferências Racionais
- Utilidades x Dinheiro
- **Redes de Decisão**
- Classificação e Avaliação de classificadores

Decision Networks

- By now we know how to use Bayesian networks to represent uncertainty and do probabilistic inference.

- Now, we extend them to support decision making adding an explicit representation of both the **actions** under consideration and the value or **utility** of the resultant outcomes gives us **decision networks** (also called **influence diagrams** by Howard and Matheson, 1981).

- Bayesian decision networks combine probabilistic reasoning with utilities, helping us to make decisions that **maximize the expected utility**

Principle of Expected Utility

- The principle of maximum expected utility asserts that an essential part of the nature of rational agents is to choose that action which maximizes expected utility.

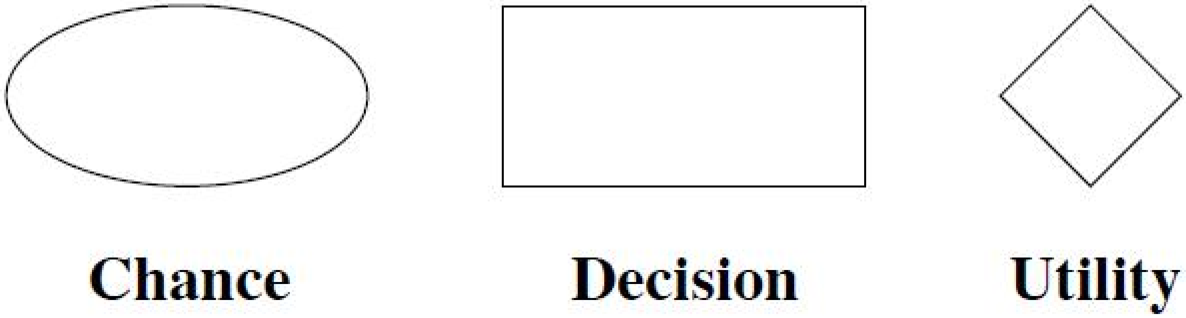
$$EU(A|E) = \sum_i P(O_i|E,A)\, U(O_i|A)$$

where

- E is the available evidence,
- A is a non-deterministic action with possible outcome states O_i,
- $U(O_i|A)$ is the utility of each of the outcome states, given that action A is taken,
- $P(O_i|E,A)$ is the conditional probability distribution over the possible outcome states, given that evidence E is observed and action A taken.

- In some cases $U(O_i \mid A) = U(O_i)$

Decision Network Node types



Decision network node types.

Rede de Decisão

- Nós de acaso: (elipses) representam variáveis aleatórias. Cada nó de acaso tem um distribuição de probabilidade condicional (dados os nós pais)

- Nós de Decisão : (retângulos) representam as possíveis ações

- Nós de utilidade: (losangos) representam as preferências do agente e podem ser usadas para definir as ações através da seleção da ação que maximiza a utilidade esperada

Football team example

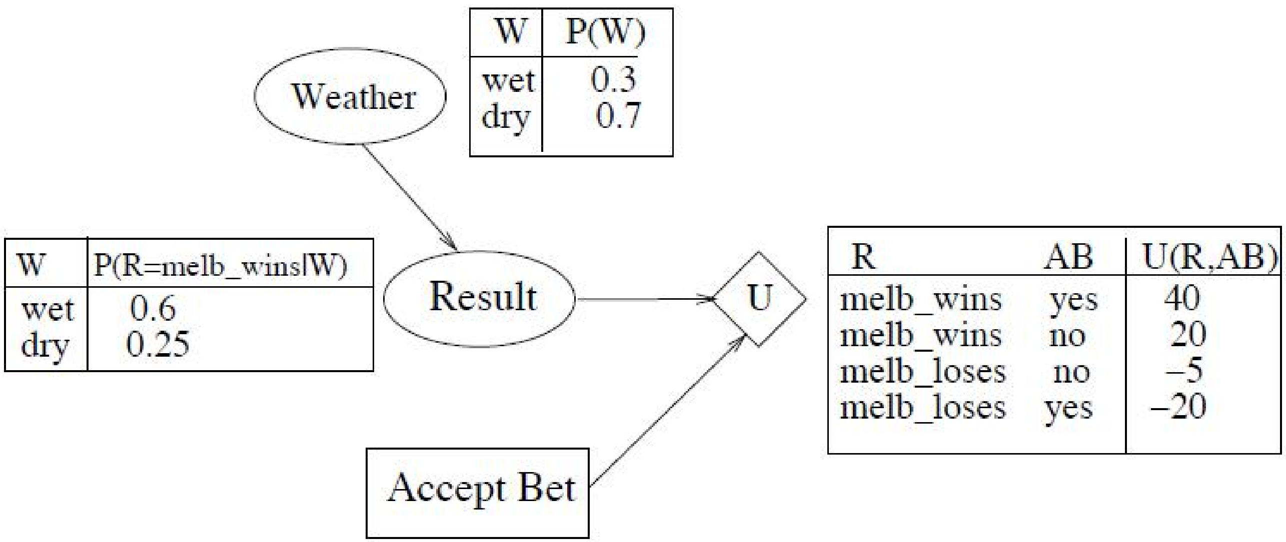
Clare's football team, Melbourne, is going to play her friend John's team, Carlton. John offers Clare a friendly bet: whoever's team loses will buy the wine next time they go out for dinner. They never spend more than $15 on wine when they eat out. When deciding whether to accept this bet, Clare will have to assess her team's chances of winning (which will vary according to the weather on the day). She also knows that she will be happy if her team wins and miserable if her team loses, regardless of the bet.

- Assume that there is 30% of raining and in that case Melbourne has 60% of wining, but only 25% if it is not wet…
- Make your decision network!
  - Variables, conditional dependence, action and utility node

Football team example

Clare's football team, Melbourne, is going to play her friend John's team, Carlton. John offers Clare a friendly bet: whoever's team loses will buy the wine next time they go out for dinner. They never spend more than $15 on wine when they eat out. When deciding whether to accept this bet, Clare will have to assess her team's chances of winning (which will vary according to the weather on the day). She also knows that she will be happy if her team wins and miserable if her team loses, regardless of the bet.

- Possible solution…



Estimating Expected Utility

- Without evidence added, the probability of Melbourne winning is

$$P(R = melb\_wins) = P(W = w) \times P(R = melb\_wins|W = w) + P(W = d) \times P(R = melb\_wins|W = d)$$

- And:

$$P(R = melb\_loses|E) = 1 - P(R = melb\_wins|E).$$

$$\begin{aligned} EU(AB = yes) &= P(R = melb\_wins) \times U(R = melb\_wins|AB = yes) \\ &+ P(R = melb\_loses) \times U(R = melb\_loses|AB = yes) \\ &= (0.3 \times 0.6 + 0.7 \times 0.25) \times 40 + (0.3 \times 0.4 + 0.7 \times 0.75) \times -20 \\ &= 0.355 \times 40 + 0.645 \times -20 = 14.2 - 12.9 = 1.3 \end{aligned}$$

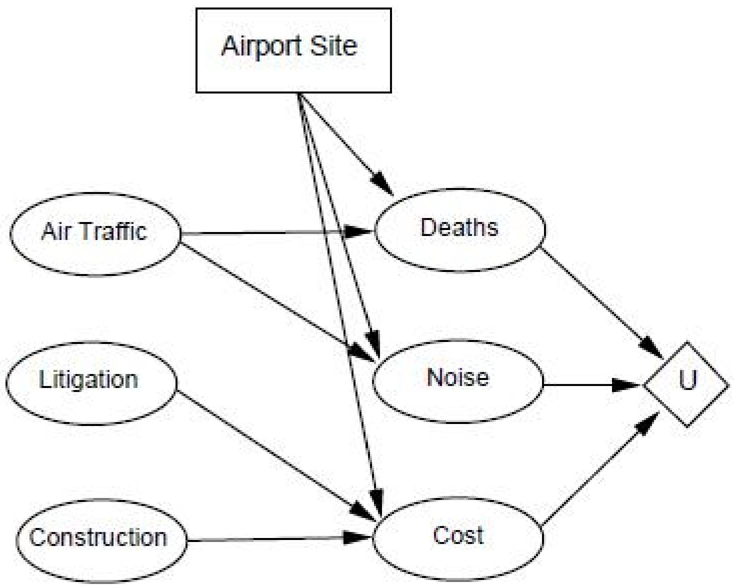
$$\begin{aligned} EU(AB = no) &= P(R = melb\_wins) \times U(R = melb\_wins|AB = no) \\ &+ P(R = melb\_loses) \times U(R = melb\_loses|AB = no) \\ &= (0.3 \times 0.6 + 0.7 \times 0.25) \times 20 + (0.3 \times 0.4 + 0.7 \times 0.75) \times -5 \\ &= 0.355 \times 20 + 0.645 \times -5 = 7.1 - 3.225 = 3.875 \end{aligned}$$

Outro Exemplo: Escolha da localização de um aeroporto

- Dependendo da posição pode-se alterar:
  - O risco de acidentes (logo, o número esperado de mortes.. **Deaths**
  - O incômodo causado pelo barulho dos aviões, quanto mais próximo de uma cidade pior….**Noise**
  - É fácil perceber que **Deaths** e **Noise** serão diretamente afetados pelo volume de **tráfego aéreo** no aeroporto.
  - Naturalmente, o custo também é alterado pela localização do aeroporto (Cost)
    - a desapropriação de um determinado terreno pode ser mais ou menos litigioso….e os custos de ligação de transportes do aeroporto a cidade podem ser maiores ou menores afetando a **construção**

Decision Network

Add action nodes and utility nodes to belief networks
to enable rational decision making

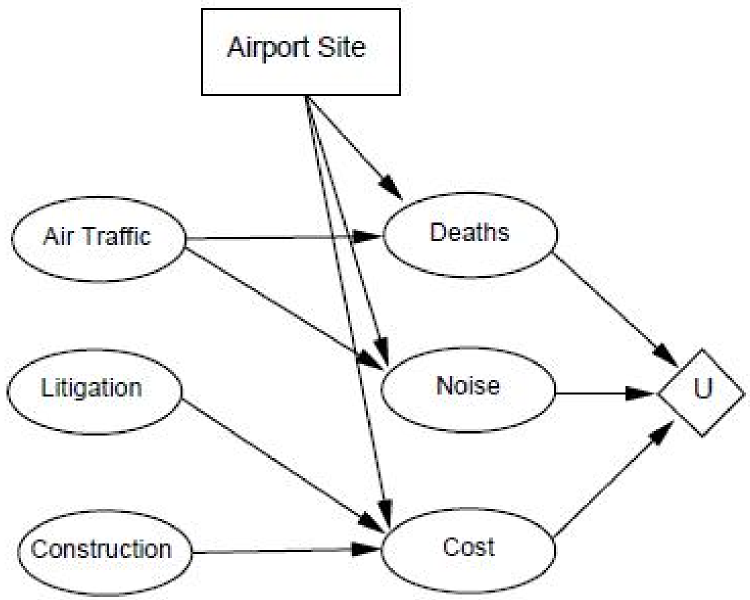


Algorithm:

For each value of action node
    compute expected value of utility node given action, evidence
Return MEU action

Podemos determinar distribuições de variáveis, mas como decidir?

- Nós de ação e nós de utilidade na rede;

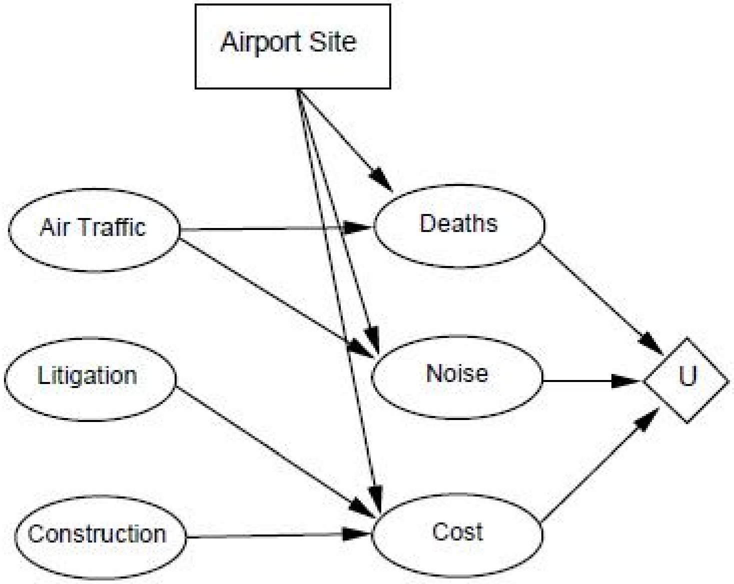


Optimal decisions: decision networks include utility information;
probabilistic inference required for $P(outcome|act\ on, ev\ dence)$

Processo de Decisão...

- pi = P(deaths=i| ASite=s, Noise=n) Ou
- P(outcome| action, evidence)

- Utilidade Esperada (action=a) = $\sum_i$ U(outcome$_i$)*P(outcome$_i$|action=a, evidence)
- Escolher ação que maximiza a utilidade esperada

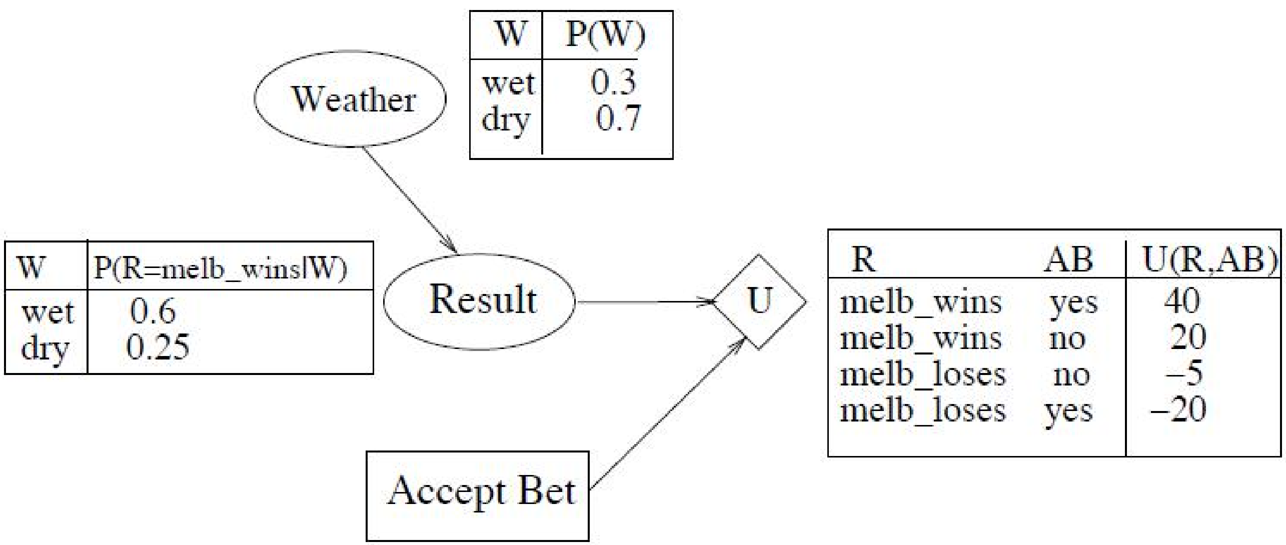


Decisões com Redes Bayesianas

- Preferências Racionais
- Utilidades x Dinheiro
- Redes de Decisão
  - Redes de Decisão e Decisão Sequenciada
- Modelo Decisório de Markov

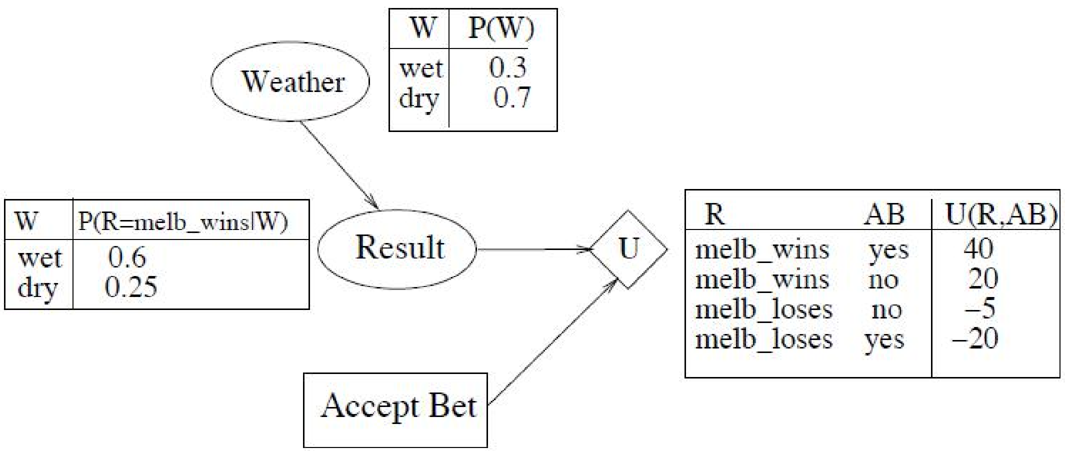
Information Link: Example

- To illustrate the use of information links, let’ s use again the Melbourne football team example, but now Clare was only going to decide whether to accept the bet or not after she heard the weather forecast
- Previously :



Previously, Clare decided don't accept the bet

- What if there is a reasonable **Forecast** about the weather **(rainy, cloudy, sunny)**? How to change the network?

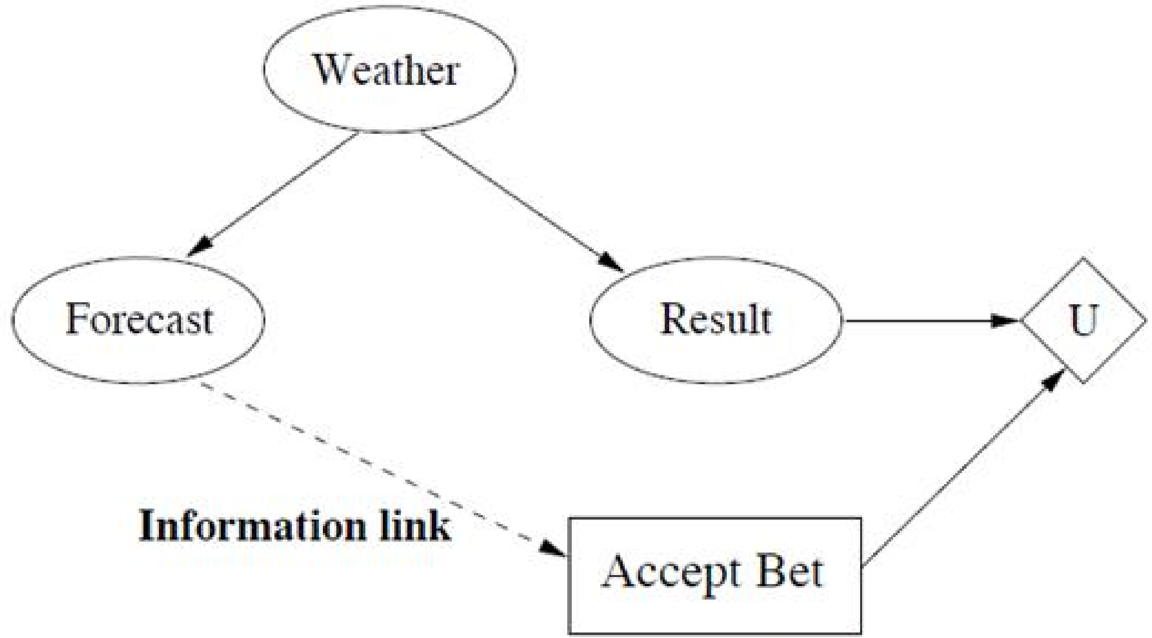


$$\begin{aligned} EU(AB=yes) &= P(R=melb\_wins) \times U(R=melb\_wins|AB=yes) \\ &+ P(R=melb\_loses) \times U(R=melb\_loses|AB=yes) = 1.3 \end{aligned}$$

$$\begin{aligned} EU(AB=no) &= P(R=melb\_wins) \times U(R=melb\_wins|AB=no) \\ &+ P(R=melb\_loses) \times U(R=melb\_loses|AB=no) = 3.875 \end{aligned}$$

Introducing the New Node, Information Link and Decision Table

W	F	P(F|W)
wet	rainy	0.60
	cloudy	0.25
	sunny	0.15
dry	rainy	0.10
	cloudy	0.40
	sunny	0.50



Decision Table

F	Accept Bet
rainy	?
cloudy	?
sunny	?

Decision Table obtained by following the algorithm

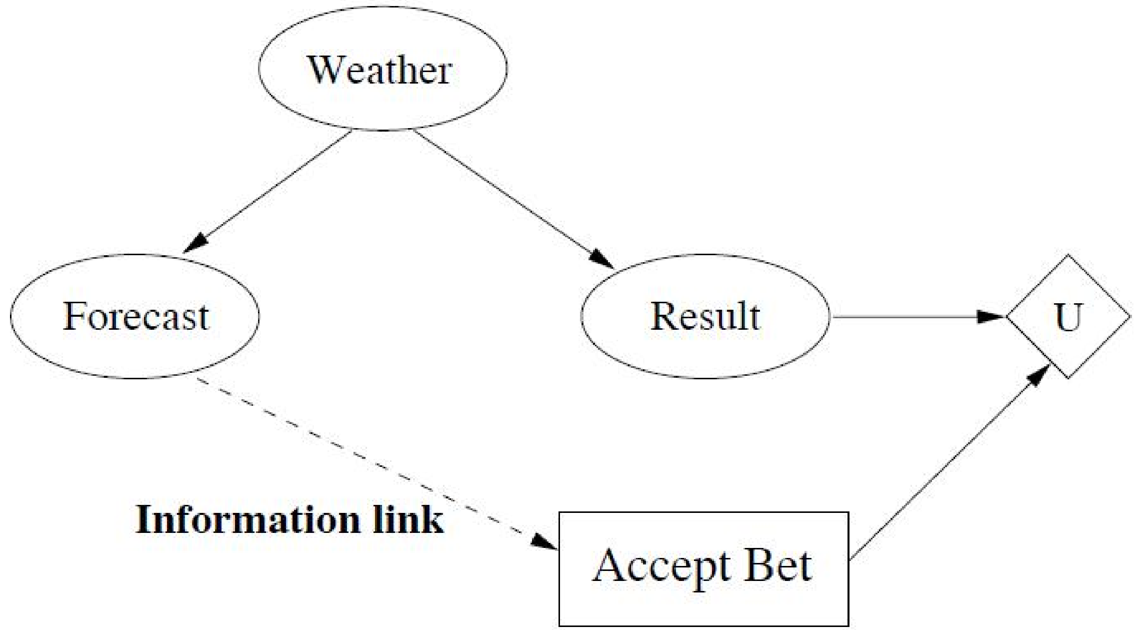
- **EU(AB=yes | F=rainy)** = P(R=melb_wins| F=rainy) x U(R=melb_wins| AB=yes,F=rainy)+ P(R=melb_loses| F=rainy) x U(R=melb_loses| AB=yes,F=rainy)
- **EU(AB=no | F=rainy)** = P(R=melb_wins| F=rainy) x U(R=melb_wins| AB=no, F=rainy)+ P(R=melb_loses| F=rainy) x U(R=melb_loses| AB=no, F=rainy)
- Analogus to other values of F…..

Decisions calculated for football team, given the new evidence node *Forecast*.

F	$Bel(W = wet)$	$Bel(R = melb\_wins)$	EU(AB=yes)	EU(AB=no)
rainy	0.720	0.502	**10.12**	7.55
cloudy	0.211	0.324	-0.56	**3.10**
sunny	0.114	0.290	-2.61	**2.25**

Decision Table with recorded actions (highest expected utility)

W	F	P(F|W)
wet	rainy	0.60
	cloudy	0.25
	sunny	0.15
dry	rainy	0.10
	cloudy	0.40
	sunny	0.50



Decision Table

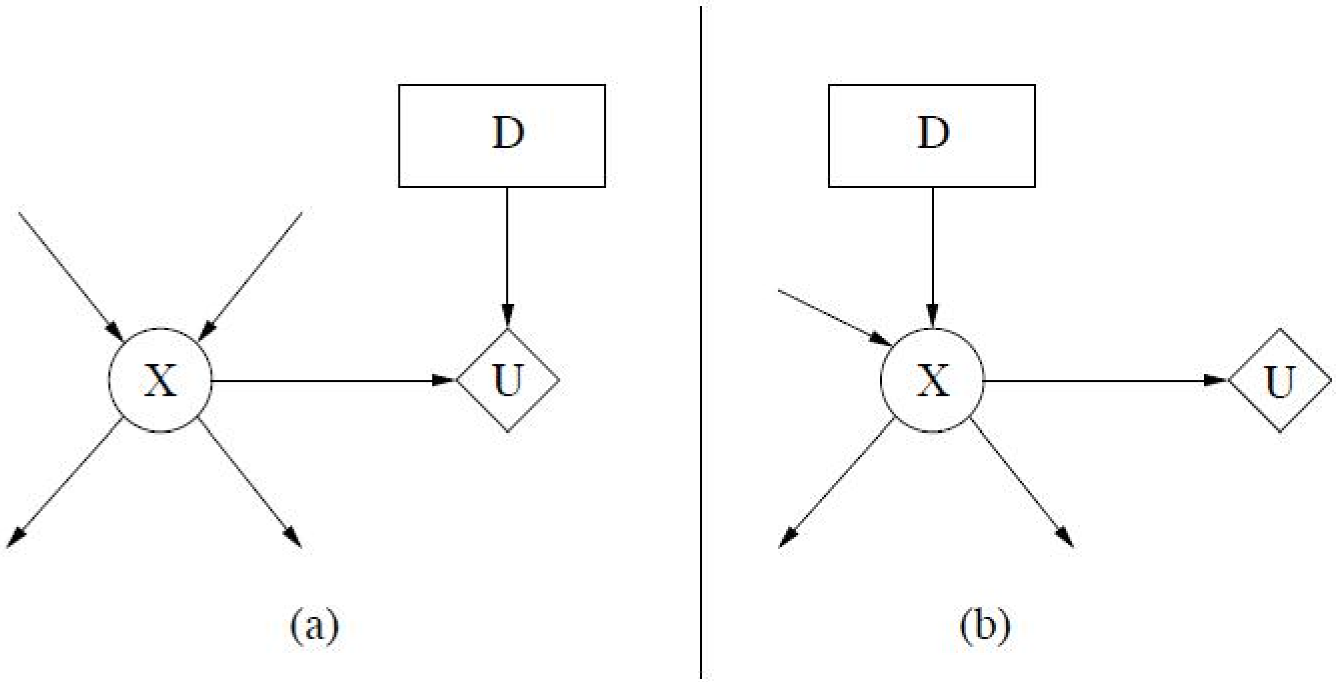
F	Accept Bet
rainy	yes
cloudy	no
sunny	no

Types of Actions

- There are two main types of actions in decision problems, **intervening** and **nonintervening**.
- **Non-intervening actions** do not have a direct effect on the chance variables being modeled, as in the Football team example.

- **Intervening actions** do have direct effects on the world, in the fever example, deciding to take aspirin will affect the later fever situation.

Types of actions



Generic decision networks for (a) non-intervening and (b) intervening actions.

CES -161 - Modelos Probabilísticos em Grafos

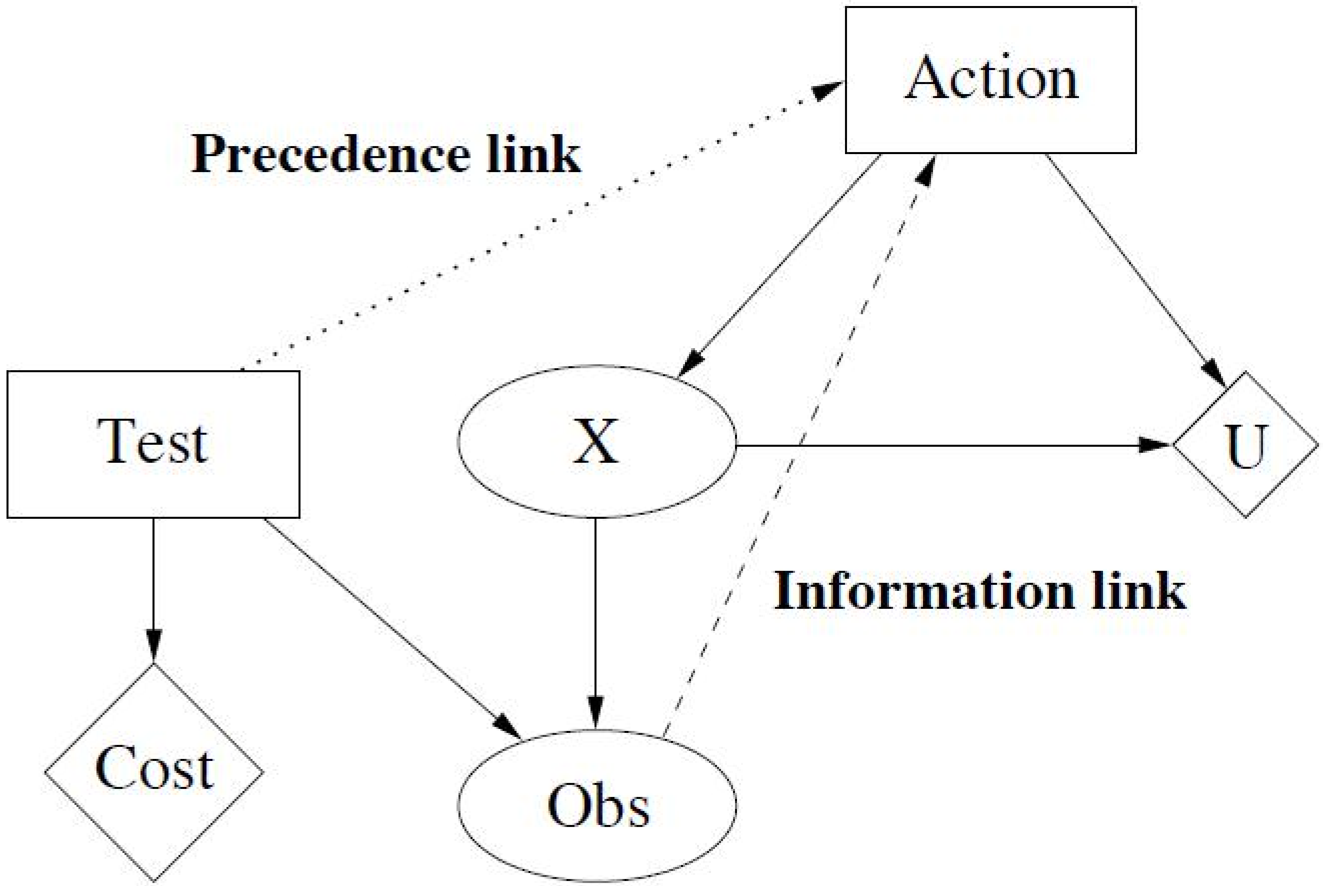
Sequential Decision Making with Bayesian Networks

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac Sala 110, IEC-ITA

Sequential Decision Making

- Thus far, we have considered only single decision problems. Often however a decision maker has to select a sequence of actions, or a plan. Sometimes, a sequence of action is not enough and it is required a function given observations
- In the football decision problem used before, Clare might have a choice as to whether to obtain the weather forecast (perhaps by calling the weather bureau)
- In the diagnostics example, the physician must decide whether to order another exam, before deciding on a treatment option.
- This type of decision problem has two stages:
  1. The decision whether to run a test or make an observation
  2. The selection of a final action

A decision network showing the general structure for these test-act decision sequences….



Test-action Decision Sequence

- If the decision is made to run the test, evidence will be obtained for the observation node Obs, before the Action decision is made; hence there is an information link from Obs to Action.

- The question then arises as to the meaning of this information link if the decision is made **not to run the test**. This situation is handled by adding an additional state, **unknown**, to the Obs node and setting the CPT for Obs

$$P(Obs = unknown | Test = no) = 1$$
$$P(Obs = unknown | Test = yes) = 0$$

Test-action Decision Sequence – 2

- In this generic network, there are **arcs from the Action node to both the chance node X and the utility node U**, indicating intervening actions with a direct associated cost. However, either of these arcs may be omitted, representing a non-intervening action or one with no direct cost, respectively

- There is an implicit assumption of no-forgetting in the semantics of a decision network. The decision maker remembers the past observations and decisions, indicated explicitly by the information and precedence links

Algorithm for Test-action Decision Sequence

1. *Evaluate decision network with any available evidence (other than for the Test result).*

   Returns Test decision.
2. *Enter Test decision as evidence.*
3. *If Test decision is 'yes'*

   Run test, get result;

   Enter test result as evidence to network.

   Else

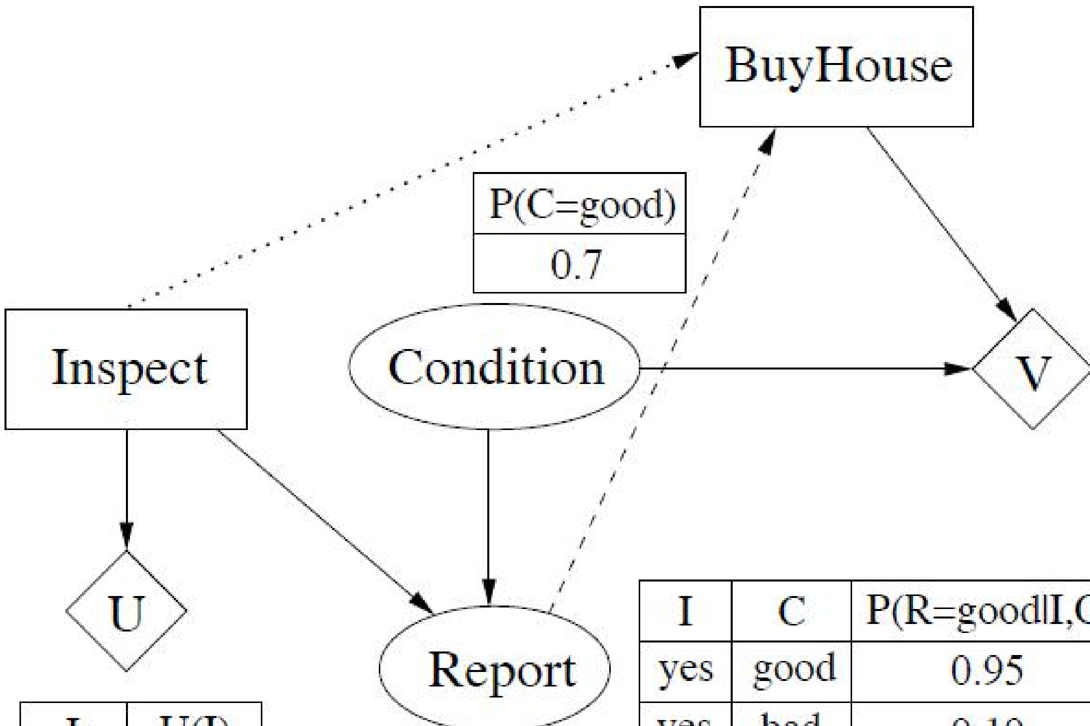
   Enter result 'unknown' as evidence to network.
4. *Evaluate decision network.*

   Returns Action decision.

Real estate investment example

Paul is thinking about buying a house as an investment. While it looks fine externally, he knows that there may be structural and other problems with the house that aren't immediately obvious. He estimates that there is a 70% chance that the house is really in good condition, with a 30% chance that it could be a real dud. Paul plans to re-sell the house after doing some renovations. He estimates that if the house really is in good condition (i.e., structurally sound), he should make a $5,000 profit, but if it isn't, he will lose about $3,000 on the investment. Paul knows that he can get a building surveyor to do a full inspection for $600. He also knows that the inspection report may not be completely accurate. Paul has to decide whether it is worth it to

A Decision Network



P(C=good)
0.7

BH	C	V(BH,C)
yes	good	5000
no	bad	0
no	good	0
yes	bad	–3000

I	U(I)
yes	–600
no	0

I	C	P(R=good|I,C)	P(R=bad|I,C)	P(R=unk|I,C)
yes	good	0.95	0.05	0.00
yes	bad	0.10	0.90	0.00
no	good	0.00	0.00	1.00
no	bad	0.00	0.00	1.00

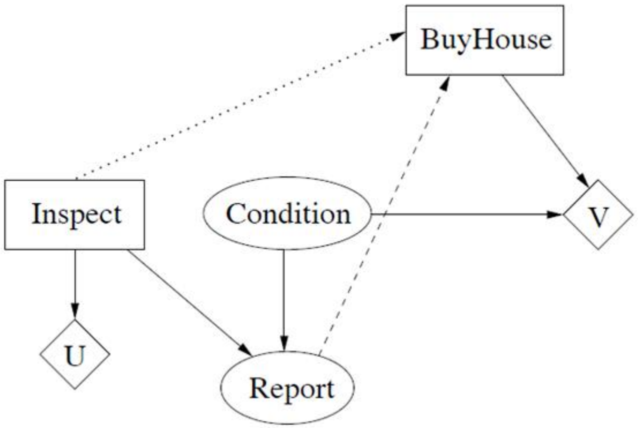
Evaluation using a decision tree model

- In order to show the evaluation of the decision network, we will use a decision tree representation

- To understand a decision tree, we start with the root node, which in this case is the first decision node, whether or not to inspect the house. Taking the path from the root to leaves each path means:
  - From a decision node, it indicates which decision is made
  - From a chance node, it indicates which value has been observed

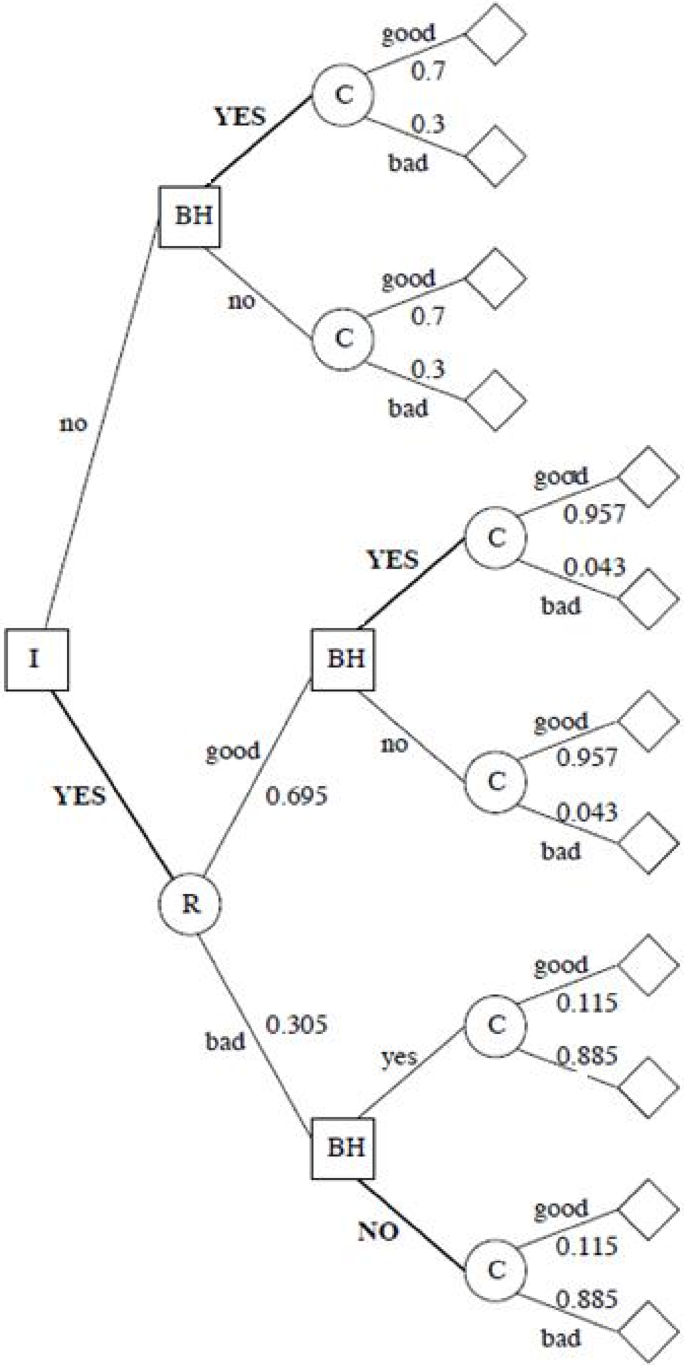
Evaluating by Decision tree

- When Paul decides about Inspection, he doesn' t have any information about the chance nodes, so there are no information links entering the Inspect decision node.



- When he decides whether or not to buy, he will know the outcome of that decision hence the **information link** from Report to BuyHouse.
- The temporal ordering of his decisions, first about the inspection, and then whether to buy, is represented by the **precedence link** from Inspect to BuyHouse.
- Note that there is a directed path from Inspect to BuyHouse (via Report) so even if there was no explicit precedence link added by the knowledge engineer for this problem, the precedence could be inferred from the rest of the network structure

- I: Inspect the house
- BH: Buy the house
- C: Condition
- R: Report



- Note: We could include
- R=unknown, when I=no
- But it wouldn’t change
- anything

How to decide? Decision Tree Evaluation Algorithm

1. *Starting with nodes that have only leaves (utility nodes) as children.*
2. *If the node X is a chance node, each outgoing link has a probability and each child has an associated utility. Use these to compute its expected utility*

$$EU(X) = \sum_{C \in Children(X)} U(C) \times P(C)$$

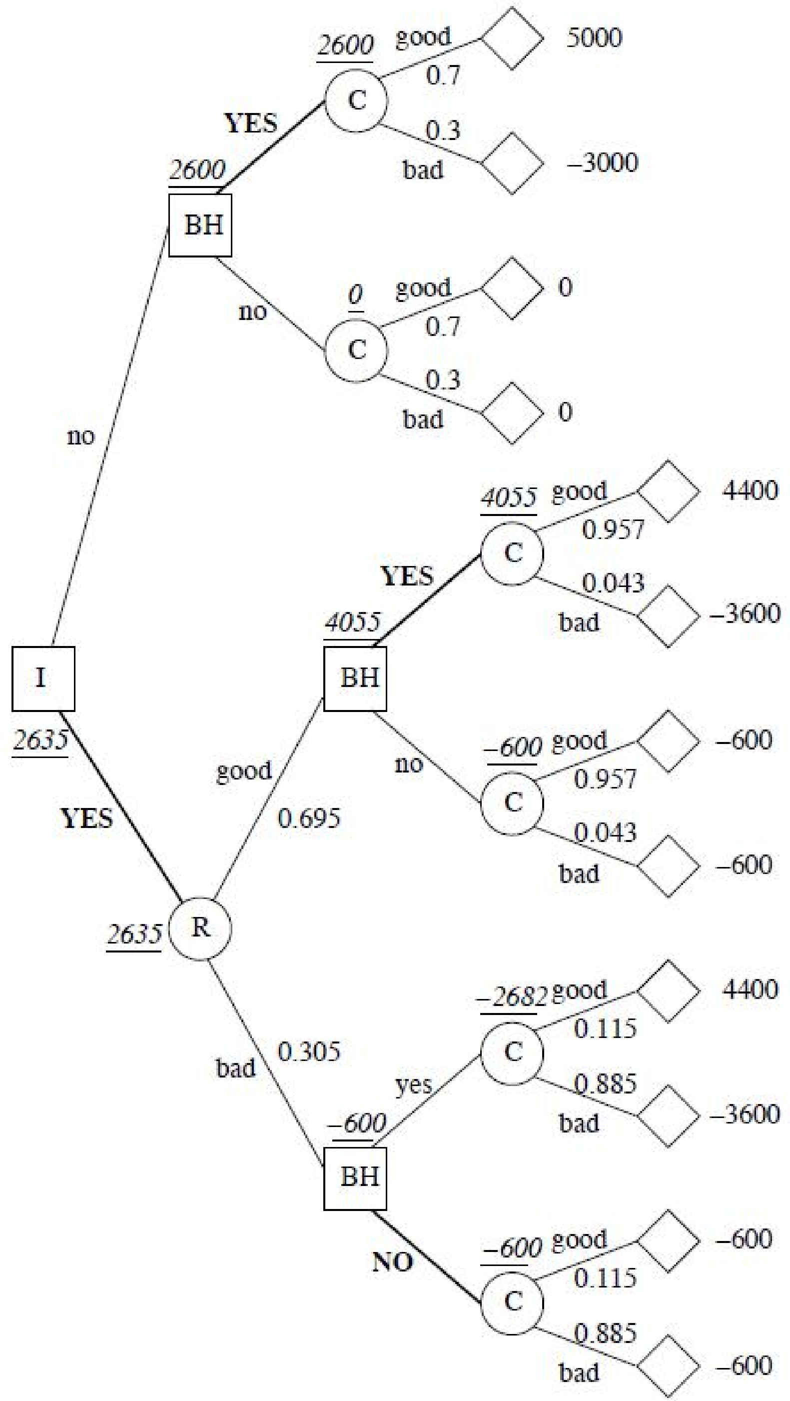
If the node is a decision node, each child has a utility or expected utility attached. Choose the decision whose child has the maximum expected utility and

$$EU(X) = max_{C \in Children(X)}(EU(C))$$

3. *Repeat recursively at each level in the tree, using the computed expected utility for each child.*
4. *The value for the root node is the maximal expected utility obtained if the expected utility is maximized at each decision.*

Evaluated Decision Tree

Expected Utilities are shown underlined



Decisions and their Expected Utilities

Decisions calculated for the real estate investment problem.

Evidence	*Bel(C=good)*	EU(I=yes)	EU(I=no)	Decision
None	0.70	**2635**	2600	I=yes
Given *I=no*		EU(BH=yes)	EU(BH=no)	
Report=unknown	0.70	**2600**	0	BH=yes
Given *I=yes*		EU(BH=yes)	EU(BH=no)	
Report=good	0.957	**4055**	-600	BH=yes
Report=bad	0.115	-2682	**-600**	BH=no

- The report may change the decision of buying the house!

Value of information

- The decision of whether to gather new information is based on the value of the information.

$$EB(Test) = EU(Test = yes) - EU(Test = no)$$

- In the real estate investment problem,

$$\begin{aligned} EB(Inspect) &= EU(Inspect = yes) - EU(Inspect = no) \\ &= 2635 - 2600 = 35 \end{aligned}$$

- Note that it already computes the cost of the inspection. So the price is worth paying

Sequential Decision Making

- Sequential Decision Making may be approached using another type of Graph Probabilistic Model (Modelos Probabilísticos em Grafos)
  - Markov Chain (Redes ou Cadeias de Markov)

- Sequential Decision making is complex in Bayesian Networks, one way of dealing with it is using Dynamic Bayesian Network (DBN) which is going to be our next subject

Dynamic Bayesian networks

Dynamic Bayesian networks

- DBN are also called dynamic belief networks (Russell and Norvig, 1995, Nicholson, 1992), probabilistic temporal networks (Dean and Kanazawa, 1989, Dean and Wellman, 1991) and dynamic causal probabilistic networks (Kjærulff, 1992).

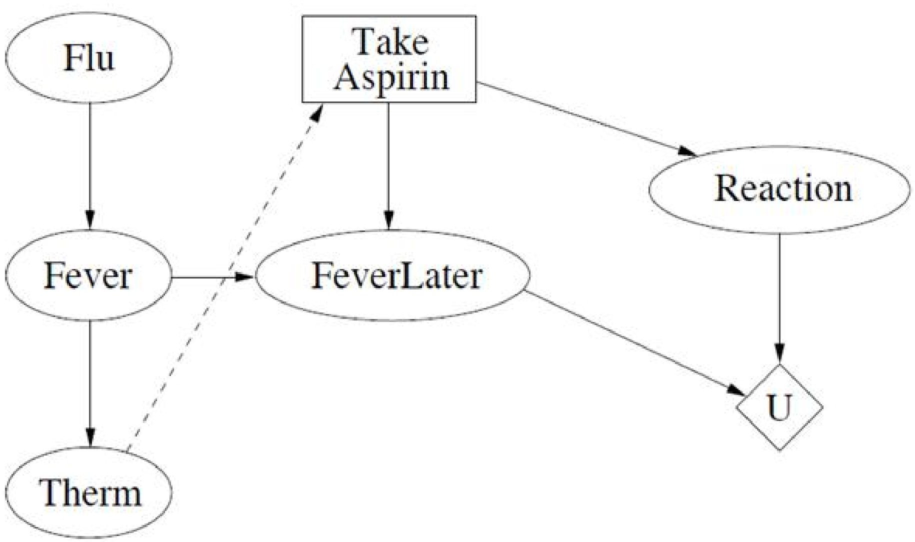
- Dynamic Bayesian networks (DBNs) explicitly model change over time.

- In the next chapter, we will extend these DBNs with decision and utility nodes, to give dynamic decision networks, which are a general model for sequential decision making or planning under uncertainty.

Bayesian Networks and time

- Although a causal relationship represented by an arc implies a temporal relationship, BNs do not explicitly model temporal relationships between variables.

- And the only way to model the relationship between the current value of a variable, and its past or future value, is by adding another variable with a different name. We saw an example of this with the fever example earlier with the use of the FeverLater node.



BN and time

- A Bayesian network is defined by a set of random variable and arcs connecting them

$$\mathbf{X} = \{X_1, \ldots, X_n\},$$

- When constructing a DBN for modeling changes over time, we include one node for each X_i for each time step. If the current time step is represented by t, the previous time step by t-1, and the next time step by t +1, then the corresponding DBN nodes will be:

  - Current: $\{X_1^t, X_2^t, \ldots, X_n^t\}$
  - Previous: $\{X_1^{t-1}, X_2^{t-1}, \ldots, X_n^{t-1}\}$
  - Next: $\{X_1^{t+1}, X_2^{t+1}, \ldots, X_n^{t+1}\}$

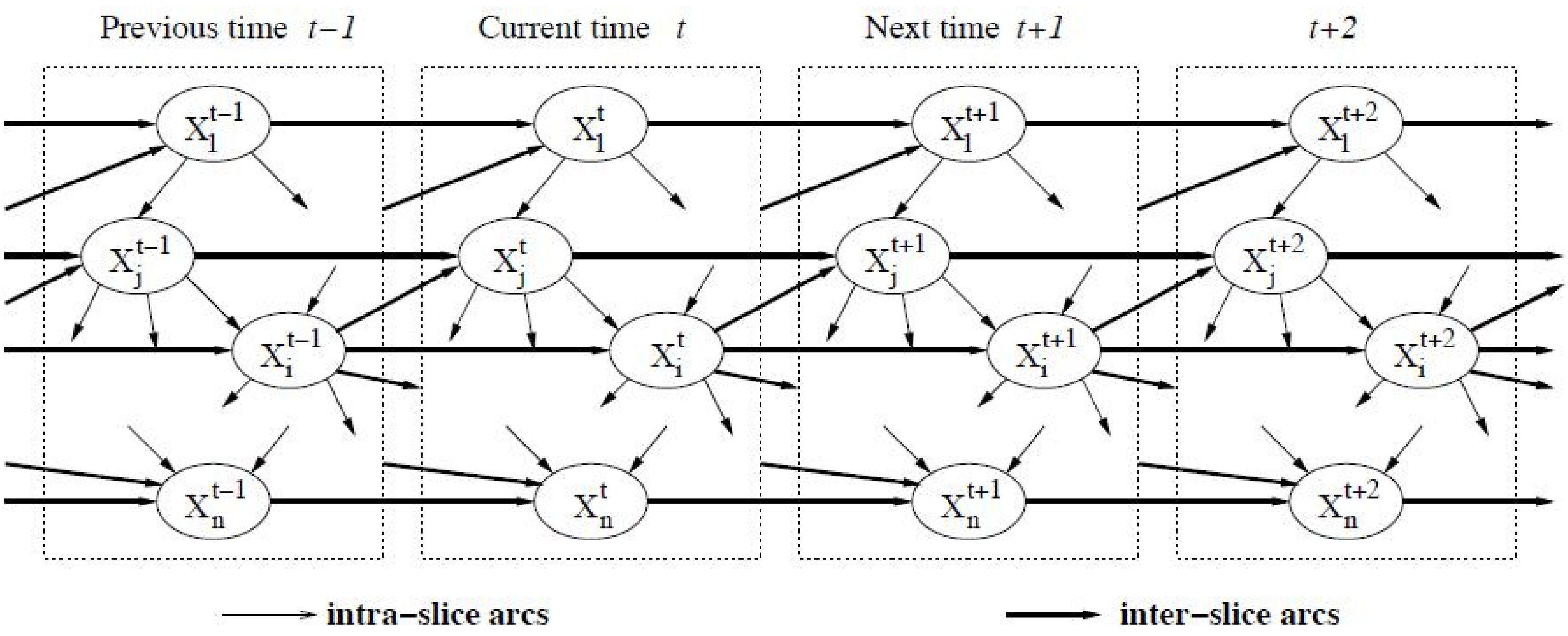
Dynamic Bayesian Network

- The relationships between variables in a time-slice are represented by **intra-slice arcs**, Although it is not a requirement, the structure of a time-slice does not usually change over time

- The relationships between variables at successive time steps are represented by **inter-slice arcs**, also called **temporal arcs**, including relationships between the same variable over time, i.e:

  - $X_i^t \rightarrow X_i^{t+1}$

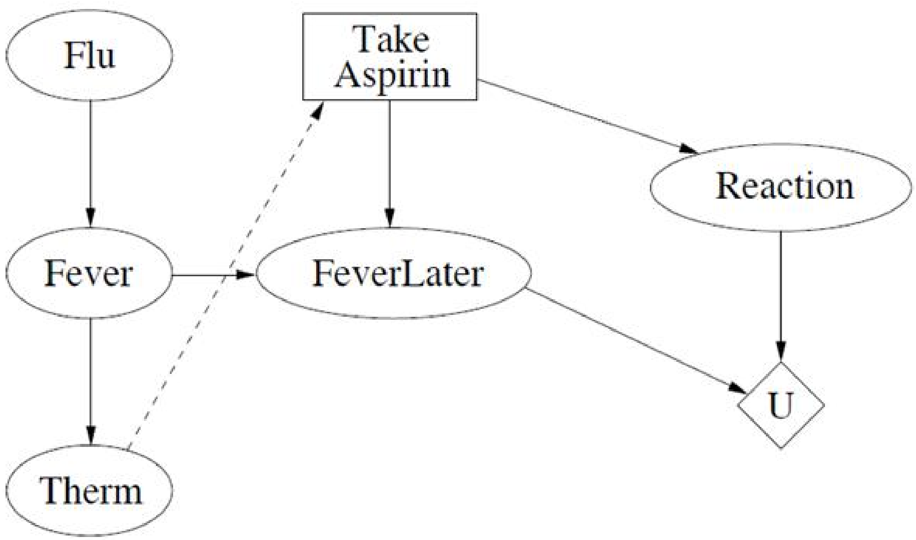
General structure of a Dynamic Bayesian Network



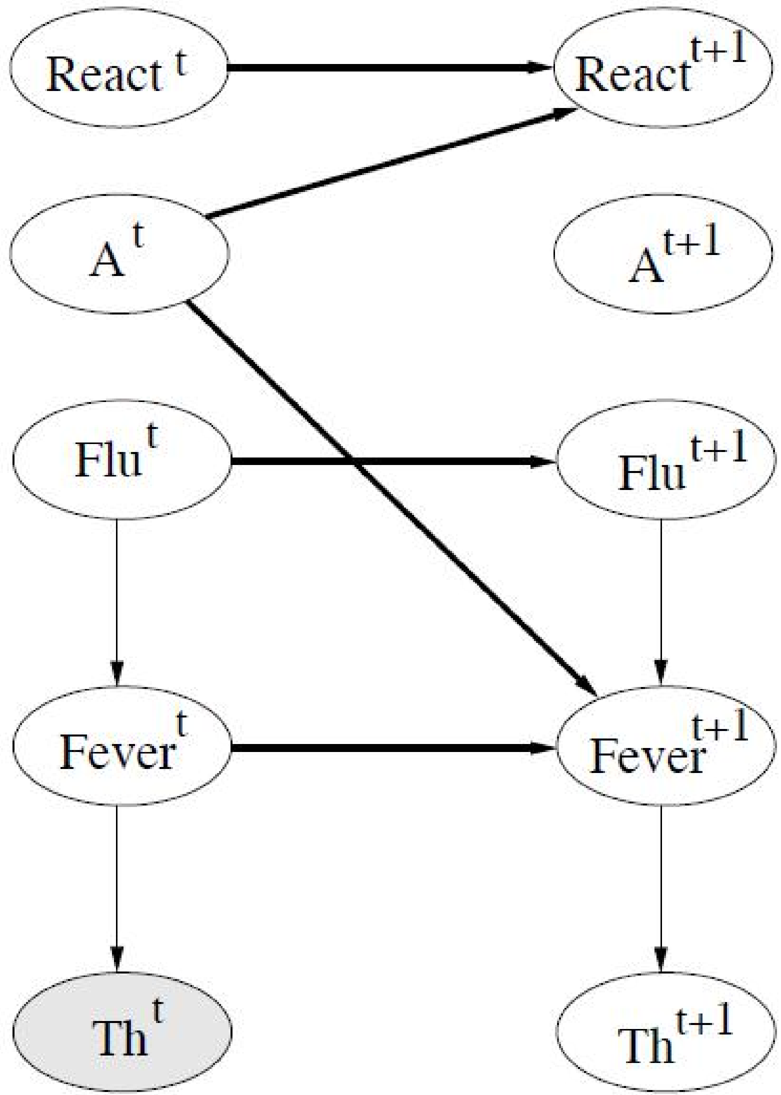
- Note that there are no arcs that span more than a single time step. This is another example of the Markov assumption

The Fever Aspirin Example

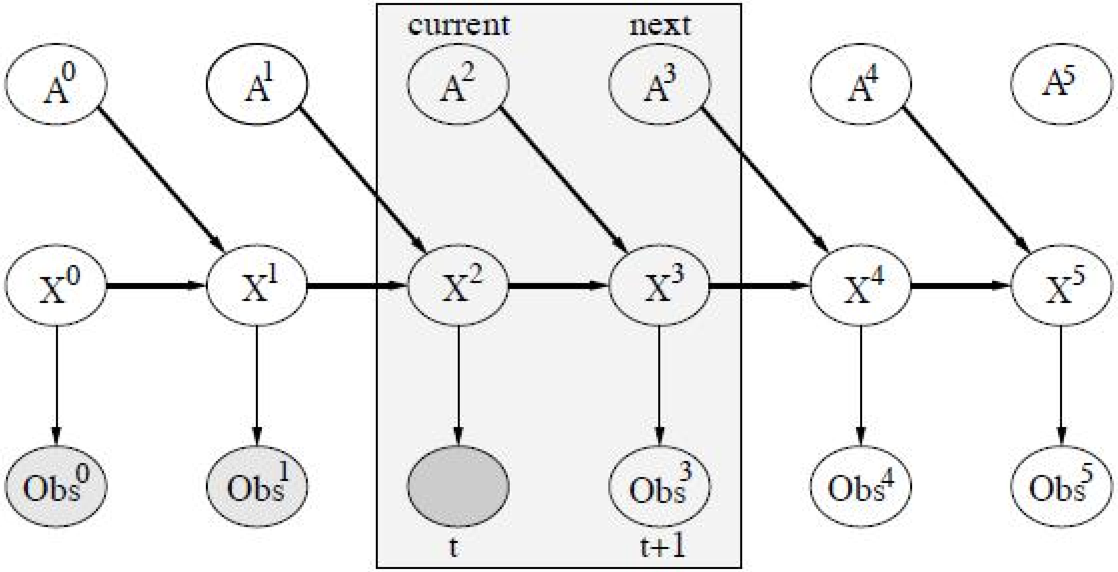
Suppose that you know that a fever can be caused by the flu. You can use a thermometer, which is fairly reliable, to test whether or not you have a fever. Suppose you also know that if you take aspirin it will almost certainly lower a fever to normal. Some people (about 5% of the population) have a negative reaction to aspirin. You'll be happy to get rid of your fever, as long as you don't suffer an adverse reaction if you take aspirin.

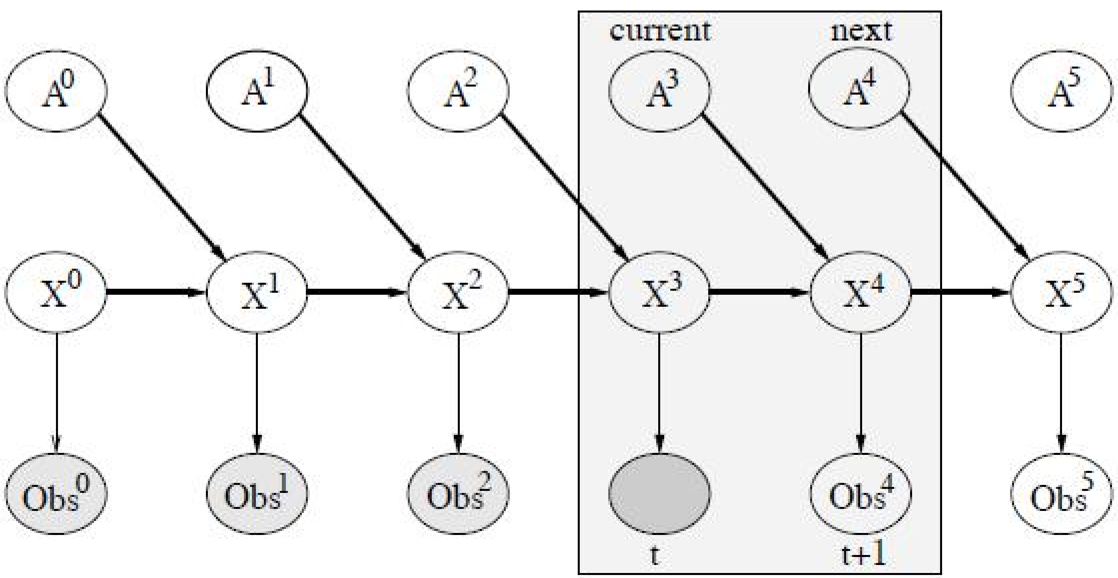


DBN Fever Aspirin Example



DBN and sliding "window" of two time-slices (Shading indicates evidence node.)





Sliding window…

- **As the reasoning process moves forward with time, one older time slice is dropped off the DBN, while another is added.**

- This use of a window means that every time we move the window along, the previous evidence received is no longer directly available. Instead, it is summarized taking the current belief for (root) nodes, and making these distributions the new priors

- The DBN updating process is given in the next Algorithm.
  - Note: steps of this DBN updating algorithm are exactly those of a technique used in classical control theory, called a Kalman Filter

DBN updating process

1. **Sliding:** *Move window along.*
2. **Prediction:**
   (a) *We already know* $Bel(X_{t-1}|\mathbf{E}_{\{1,t-1\}})$*, the estimated probability distribution over* X_{t-1}.
   (b) *Calculate the predicted beliefs,* $\widehat{Bel}(X_t|\mathbf{E}_{\{1,t-1\}})$,
3. **Rollup:**
   (a) *Remove time-slice* $t-1$.
   (b) *Use the predictions for the* t *slice as the new prior by setting* $P(X)$ *to* $\widehat{Bel}(X_t|\mathbf{E}_{\{1,t-1\}})$.
4. **Estimation:**
   (a) *Add new observations* $\mathbf{E}_t$.
   (b) *Calculate* $Bel(X_t|\mathbf{E}_{\{1,t\}})$*, the probability distribution over the current state.*
   (c) *Add the slice for* $t+1$.

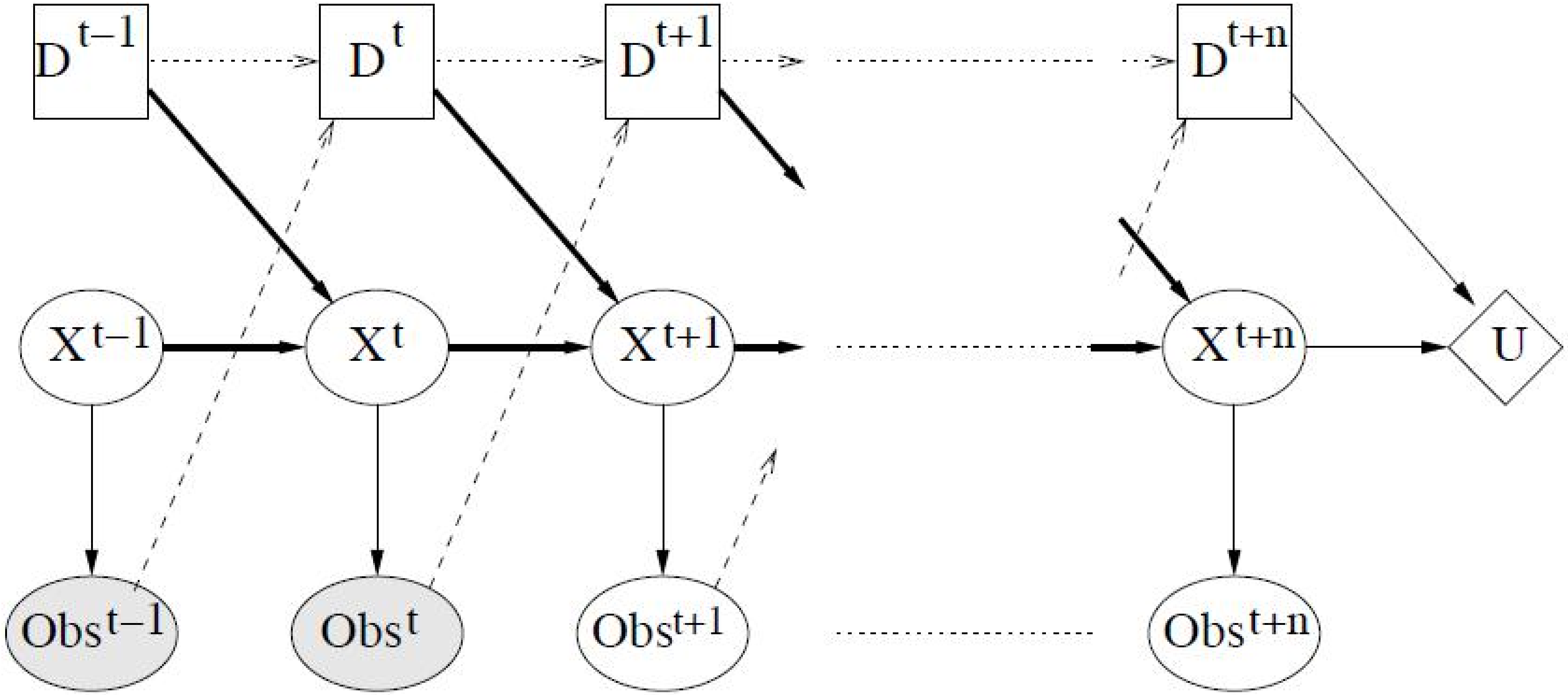
Inference in DBN

- Exact **clustering algorithms** can be applied to DBNs, particularly if the inference is restricted to two time-slices

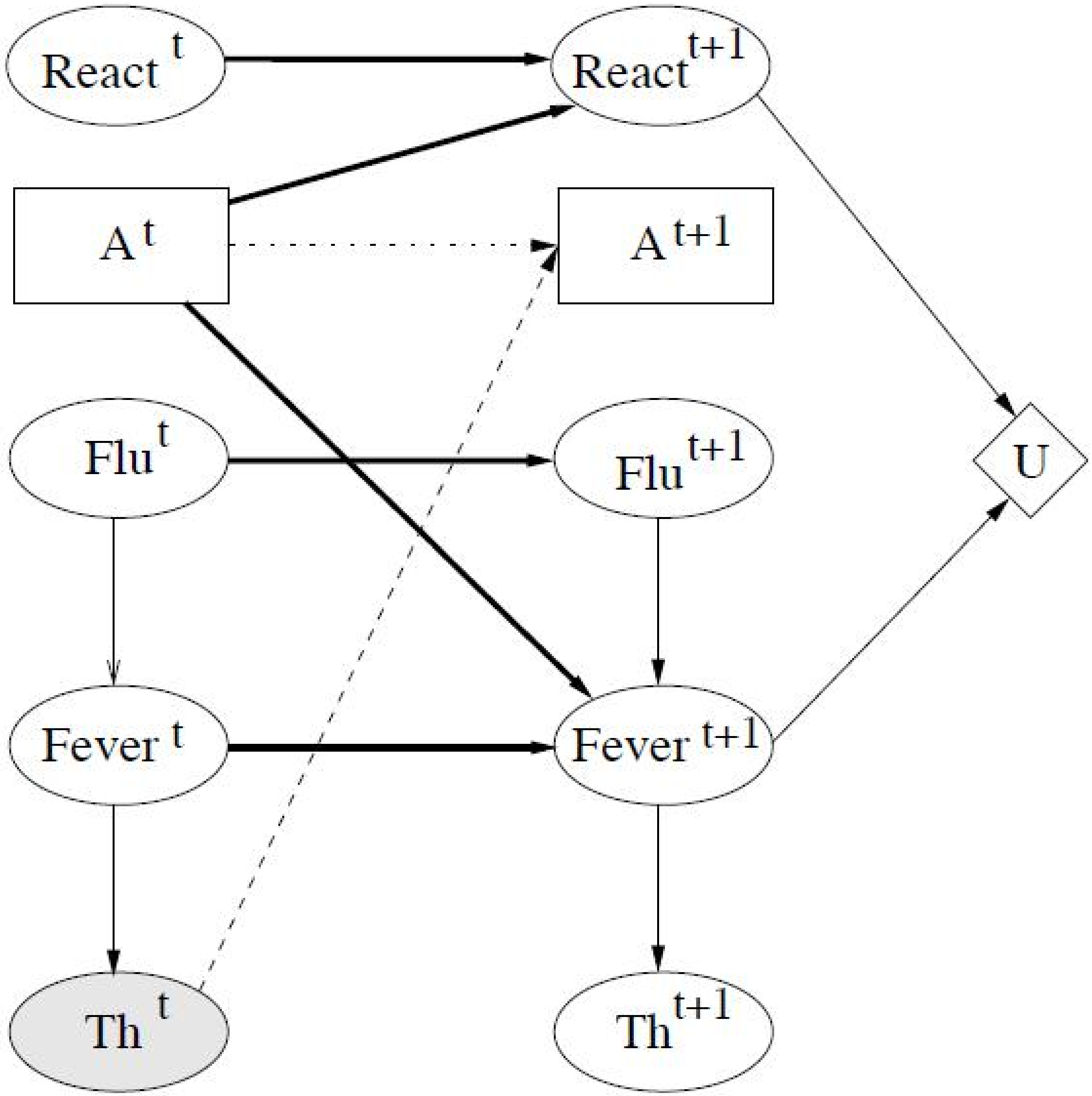
- Unfortunately, it is common that there is a cluster containing all the nodes in a time slice with inter-slice connections, so the clusters become hard computationally

Dynamic decision networks

- Just as Bayesian networks can be extended with a temporal dimension to give DBNs so can decision networks be extended to give dynamic decision networks (DDNs).



A DDN for the Fever problem

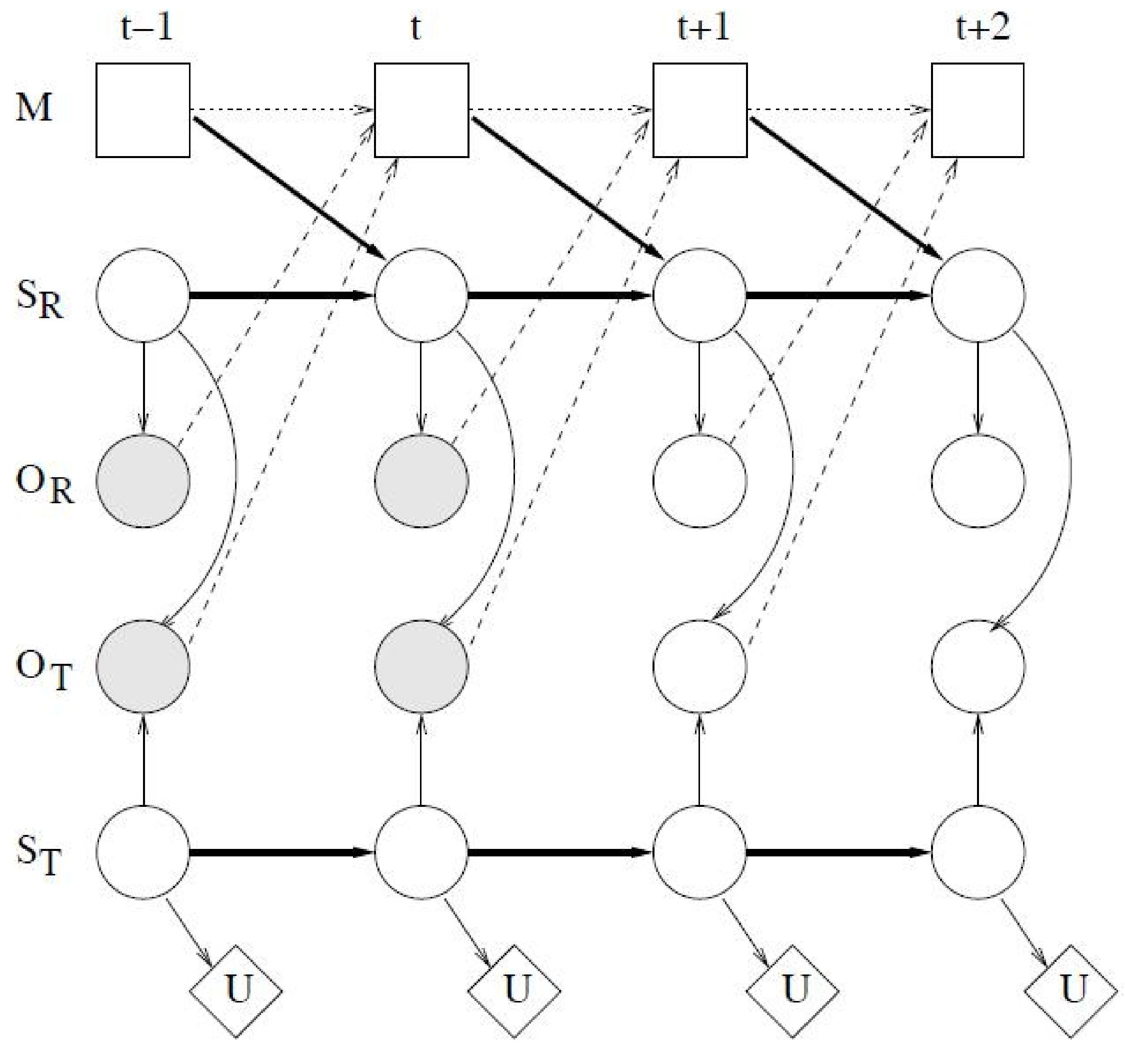


Mobile Robot Example

The robot's task is to detect and track a moving object, using sonar and vision sensor information, given a global map of the office floor environment. The robot must also continually reassess its own position (called localization) to avoid getting lost. At any point in time, the robot can make observations of its position with respect to nearby walls and corners and of the target's position with respect to the robot.

- The problem of a mobile robot that does localization and tracking can be modeled with a DDN as follows: The nodes S_T and S_R represent the locations of the target and the robot, respectively
- The decision node is M, representing the robot's movement actions options
- The nodes O_R and O_T represent the robot's observations of its own and the target's location, respectively
- The overall utility is the weighted sum over time of the utility at each step U_t , which is a measure of the distance between the robot and its target.

Mobile Robot Example



CES -161 - Modelos Probabilísticos em Grafos

Learning Probabilistic Models and

Knowledge Engineering with Bayesian Networks

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac
Sala 110, IEC-ITA

Learning Probabilistic Models

- Bayesian network structure may be constructed by:
  - hand, presumably during a process of eliciting causal and probabilistic dependencies from an expert or
  - learning via causal discovery or
  - a combination of both approaches
- However the structures are arrived at, they will be useless until parameterized, i.e., the conditional probability tables are specified, characterizing the direct dependency between a child and its parents.

Learning Probabilistic Models

- Learning probabilities
- Learning Models
- Bayesian Classifiers
- Evaluating Classifiers

Learning probabilities

- Parameterizing a categorial (or multinomial) model (variables with two or more possible values) can be done using a conjugate family of distributions, namely the Dirichlet family of distributions

- The Dirichlet distribution with τ possible values is written $D[\alpha_1,..,\alpha_i,..,\alpha_\tau]$ with α_i being the hyperparameter for value i and the probabilities of each value are given by:

$$P(X=i)=\frac{\alpha_i}{\sum_{j=1}^{\tau}\alpha_j}$$

Learning probabilities

- When we observe a data instance (or data point) where x=i then we can update the distribution from D[α_1 ,.., α_i,.., α_τ]   to D[α_1 ,.., α_i+1,.., α_τ] and the probabilities can still calculated in the same way, but now it reflects the observed instances

$$P(X = i) = \frac{\alpha_i}{\sum_{j=1}^{\tau} \alpha_j}$$

- In fact when α_i = 1 for all i, it is equivalent to the Laplace estimator (or smoothing).
- Note that small values of α_i make the data points more relevant and vice-versa

Multiple node

- The technique provides parameter estimation for BNs with a single node. What about multiple nodes?

- In order to parameterize an entire network, we can simply iterate over its nodes using a algorithm called Multinomial parameterization

- This algorithm is a very simple counting solution to the problem of parameterizing multinomial networks. This solution is certainly the most widely used and is available in the standard Bayesian network tools

Multinomial Parameterization

Algorithm *Multinomial Parameterization (Spiegelhalter and Lauritzen method)*

1. *For each node* X_j

   For each instantiation of $Parents(X_j)$*, assign some Dirichlet distribution for the* τ *states of* X_j $D[\alpha_1, \ldots, \alpha_i, \ldots, \alpha_\tau]$

2. *For each node* X_j

   For each joint observation of all variables $X_1, \ldots, X_k$

   (a) *Identify which state* i X_j *takes*

   (b) *Update* $D[\alpha_1, \ldots, \alpha_i, \ldots, \alpha_\tau]$ *to* $D[\alpha_1, \ldots, \alpha_i + 1, \ldots, \alpha_\tau]$ *for the distribution corresponding to the parent instantiation in the observation*

Problems

- So, there are two computationally difficult tasks these learners need to perform

- **First, they need to compute their metric**, scoring individual hypotheses. This scoring procedure may itself be computationally expensive

- **Second, they need to search the space of causal structures**, which, is known to be exponential

- Most of the search methods applied have been variants of greedy search

K2

- The first significant attempt at a Bayesian approach to learning discrete Bayesian networks without topological restrictions was made in 1991 (Cooper and Herskovits, 1991)
- Their approach is to compute the metric for individual hypotheses, $P(h_i|e)$ by brute force
- Since our goal is to find that h_i which maximizes P(hije), we can satisfy this by maximizing $P(h_i|e)$, as we can see from Bayes' theorem

$$
\begin{aligned}
P(h_i|e) &= \frac{P(e|h_i)P(h_i)}{P(e)} \\
&= \frac{P(h_i,e)}{P(e)} \\
&= \beta P(h_i,e)
\end{aligned}
$$

K2 - Simplifying assumptions

1. The data are joint samples and all variables are discrete
2. Samples are independently and identically distributed (i.i.d.).
3. The data contain no missing values. If, in fact, they do contain missing values, then they need to be filled in with some estimate method
4. For each variable Xk in hi and for each instantiation of its parents Parents(Xk), P(Xk = xjhi;q;p(Xk)) is uniformly distributed over possible values Xk = x. (where q is the parameter vector (e.g., conditional probabilities))
5. Assume the uniform prior over the causal model space; i.e. $P(h_i)=1/|\{h_i\}|$

Given these simplyfing assumptions…

- Given these simplifying assumptions, Cooper and Herkovits (1991) showed that the joint probability can be given by:

$$P(h_i, e) = P(h_i) \prod_{k=1}^{N} \prod_{j=1}^{|\Phi_k|} \frac{(s_k - 1)!}{(S_{kj} + s_k - 1)!} \prod_{l=1}^{s_k} \alpha_{kjl}!$$

Where

- N is the number of variables.
- $|\Phi_k|$ is the number of assignments possible to $\pi(X_k)$.
- s_k is the number of assignments possible to X_k.
- α_{kjl} is the number of cases in sample where X_k takes its l-th value and $\pi(X_k)$ takes its j-th value.
- S_{kj} is the number of cases in the sample where $\pi(X_k)$ takes its j-th value (i.e., $\sum_{l=1}^{s_k} \alpha_{kjl}$).

K2

- The computation of Cooper and Herkovits' s formula do $P(h_i, e)$ is polynomial, i.e., computing P given a particular h_i is tractable under the assumptions

- Ok, but the number of possible h_i is very big... Yes, it is!

- So, they made another simplifying assumption:
  - Assume we know the temporal ordering of the variables
- That way, the search space is greatly reduced. In fact, for any pair of variables either they are connected by an arc or they are not.

K2

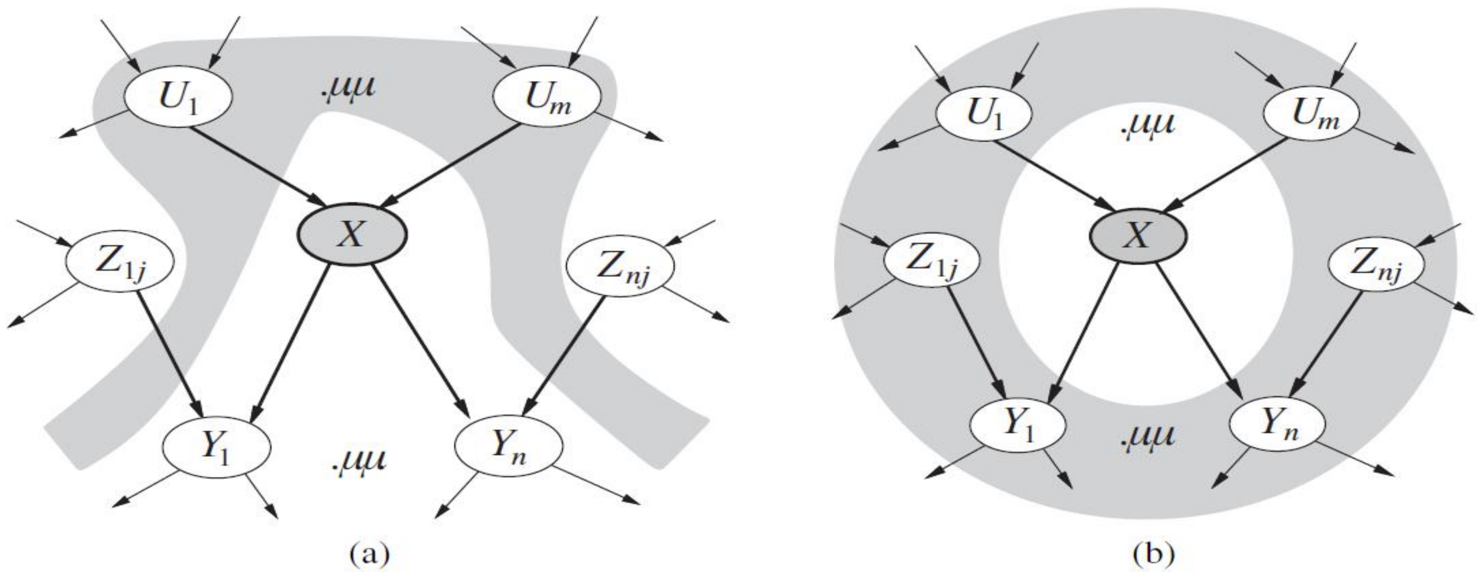
- Limitation of the maximum number of parents are also commonly required, as a way to reduce the search space
- Many variations can be used with K2, for instance change the metric from Bayesian to others metrics as: Minimum Description Length (MDL), Akaike Information Criterion (AIC), Entropy and others
- Machine learning frameworks provide several alternative methods for building Bayes networks,but K2 is one the most relevant methods

Problem with full causal

- The problem with full causal discovery for prediction, however, is the same as the problem with any kind of feature selection or model selection:
  - frequently the true model is not what is learned, but some similar, yet different, model is learned instead.

- The result may be that variables that should be in the target variable’s Markov blanket are not and so are ignored during prediction, with potentially disastrous consequences for predictive accuracy.

Indepence from Non-descendants and Markov blanket



- a) A node X is independent of its non-descendant given its parents (X is independent of Z..)
- b) Markov Blanket. A node is indenpendent of all other nodes given its Markov Blanket

Problems with NB and TAN

- NB and TAN, on the contrary, include all the attributes in their predictions, so this source of error is not even possible.

- To be sure, by being all-inclusive NB and TAN introduce a different potential source of error, namely **overfitting**.

- It may be that some variables are directly associated with the target variable only accidentally, due to noise in the available data

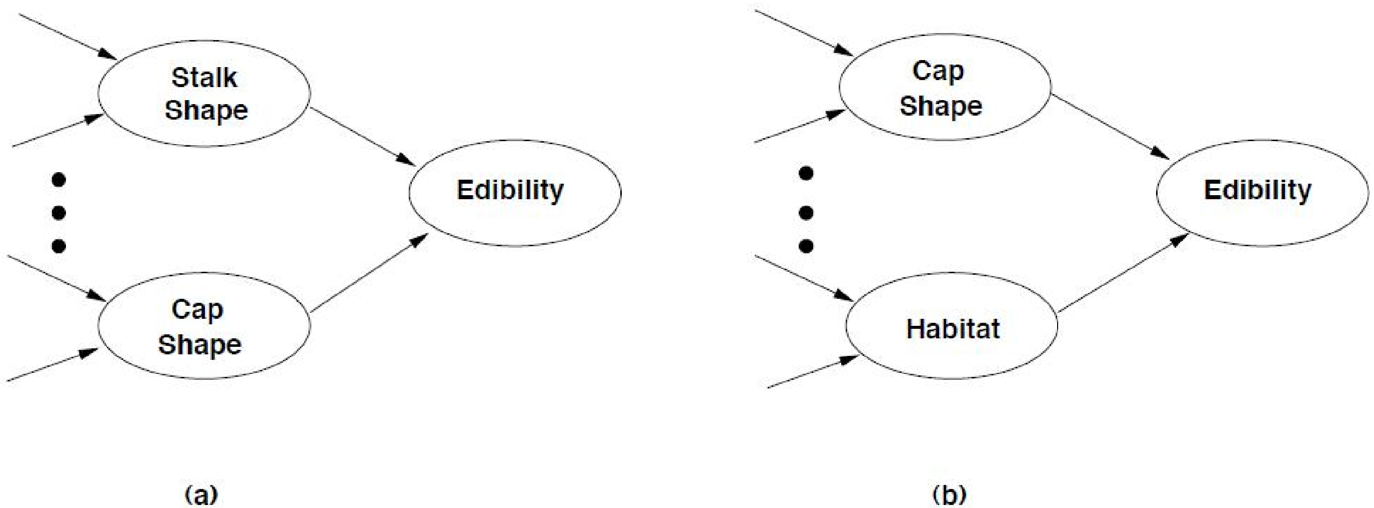
Problems with NB and TAN

- To compensate for this one may introduce **variable selection**, eliminating those attributes from the model which are contributing little to the prediction.

- In the latter case, again, incorrect variable selection returns us to the problem faced by causal discovery for prediction: variables missing from the target’s Markov blanket

Ensemble Bayes models

- Another response to the problem of incorrectly identifying the Markov blanket, aside from utilizing all attributes as predictors, is to move to ensembles of predictive models.
- This means mixing the predictions of some number of distinct models together, using some weighting over the models
- For example, (below) there are two alternative (partial) Bayesian networks for the mushroom problem (note that they are not NB models)



Two BN models explaining (and predicting) edibility.

Análise de Classificadores – Kappa

- O problema com a taxa de acerto (e outras..) é que não levam em consideração os acertos por puro acaso…
- Uma alternativa: estatística kappa (Cohen' s kappa)
  - $\kappa = (p_o - p_e)/(1 - p_e)$
  - p_o é a concordância observada
  - p_e é a concordância esperada
- Kappa mensura o ganho em relação a distribuição esperada aleatória, 0 significa que não faz melhor do que ela e 1 significa perfeita acurácia.
- Um problema da estatística kappa (e também da taxa de acerto) é que não leva em consideração o custo dos erros…que podem ser diferentes e bem mais significativos que outros
- Cada parâmetro de comparação é parcial e deve-se fazer uma análise vários parâmetros (há vários outros...precision, f-measure, etc) considerando as particularidades do domínio do problema

Exemplo

- O classificador A prevê uma distribuição de classes com <0.6;0.3;0.1>, se o classificador fosse independente da classe real haveria uma distribuição na mesma proporção (Expected distribution)

(a) Classifier A

		Predicted class a	b	c	Total
Actual class	a	88	10	2	100
	b	14	40	6	60
	c	18	10	12	40
	Total	120	60	20	

(b) Expected Distribution

		Predicted class a	b	c	Total
Actual class	a	60	30	10	100
	b	36	18	6	60
	c	24	12	4	40
	Total	120 60%	60 30%	20 10%	

- p_o = 88+ 40+12 = 140/200 (concordância observada)
- p_e = 60+18+4= 82/200 (concordância esperada)
- $\kappa = (p_o - p_e)/(1-p_e)$ = (140-82)/ (200-82) = 52 / 118 = 49,2%

Kappa statistic Interpretation

- *Cohen's Kappa*: Measures relative improvement on random predictor: 1 means perfect accuracy, 0 means we are doing no better than random

Value of Kappa	Level of Agreement
0-.20	None
.21-.39	Minimal
.40-.59	Weak
.60-.79	Moderate
.80-.90	Strong
Above .90	Almost Perfect

- **(McHugh, 2012) Interrater reliability: the kappa statistic.** Marry L. McHugh.

CES -161 - Modelos Probabilísticos em Grafos

Knowledge Engineering with Bayesian Networks

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac Sala 110, IEC-ITA

Knowledge Engineering with Bayesian Networks

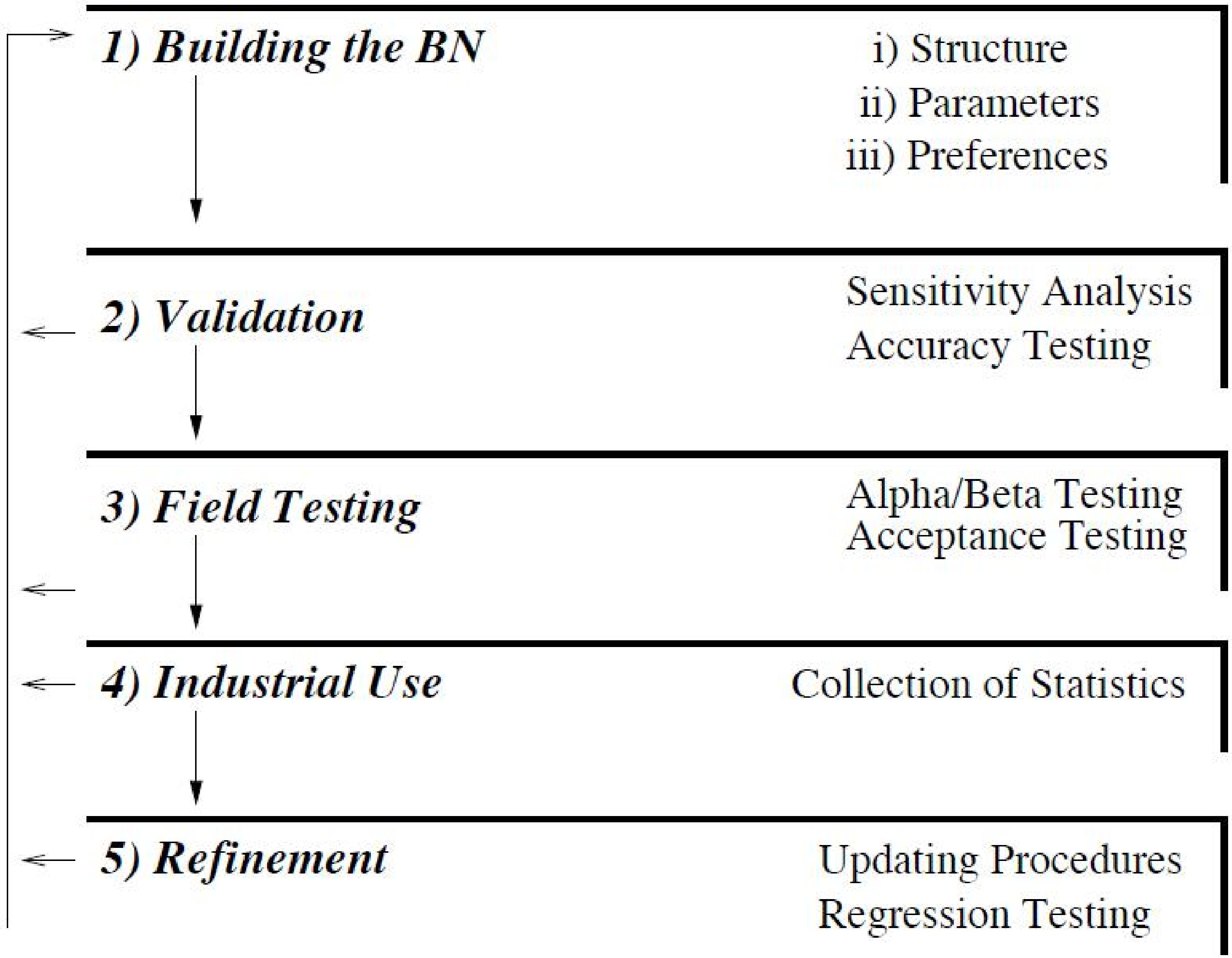
- When constructing a Bayesian network, the major modeling issues that arise are:
  - What are the variables? What are their values/states?
  - What is the graph structure?
  - What are the parameters (probabilities)?
- When building decision nets, the additional questions are:
  - What are the available actions/decisions, and what impact do they have?

KEBN life cycle model

- A simple view of the software engineering process construes it as having a lifecycle: the software is born (design), matures (coding), has a lengthy middle age (maintenance) and dies of old age (obsolescence).

- One effort at construing KEBN in such a lifecycle model (also called a "waterfall" model) is shown next.

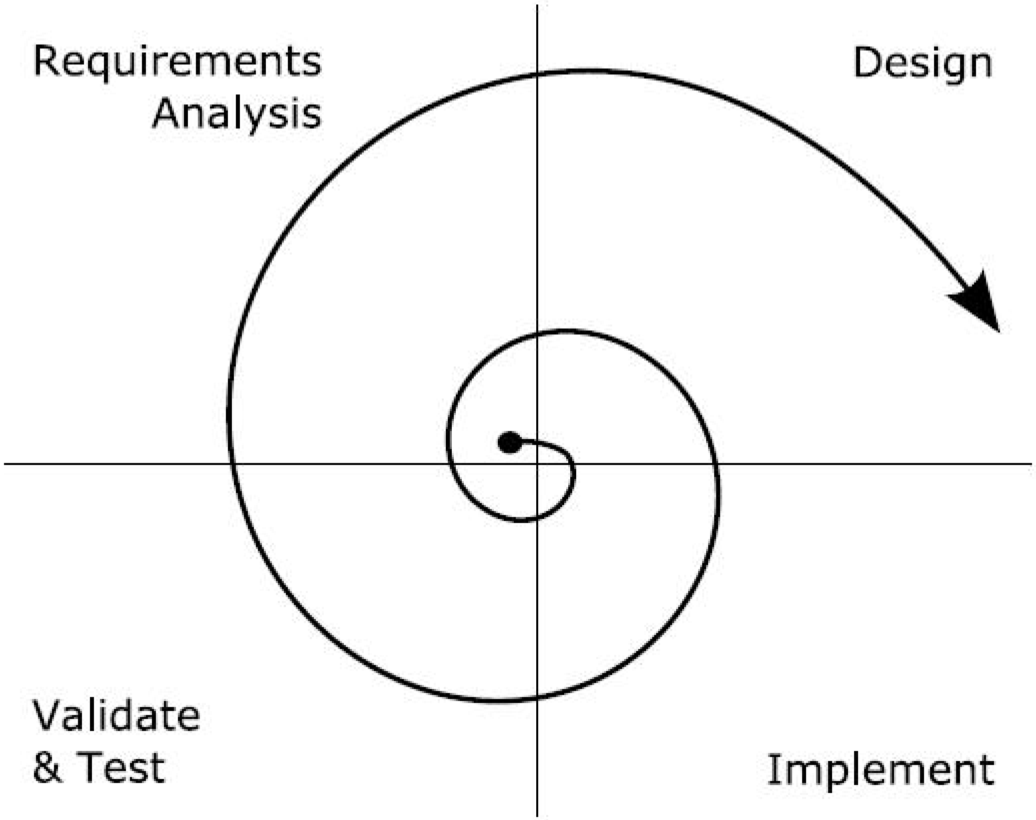
KEBN “waterfall” life cycle model



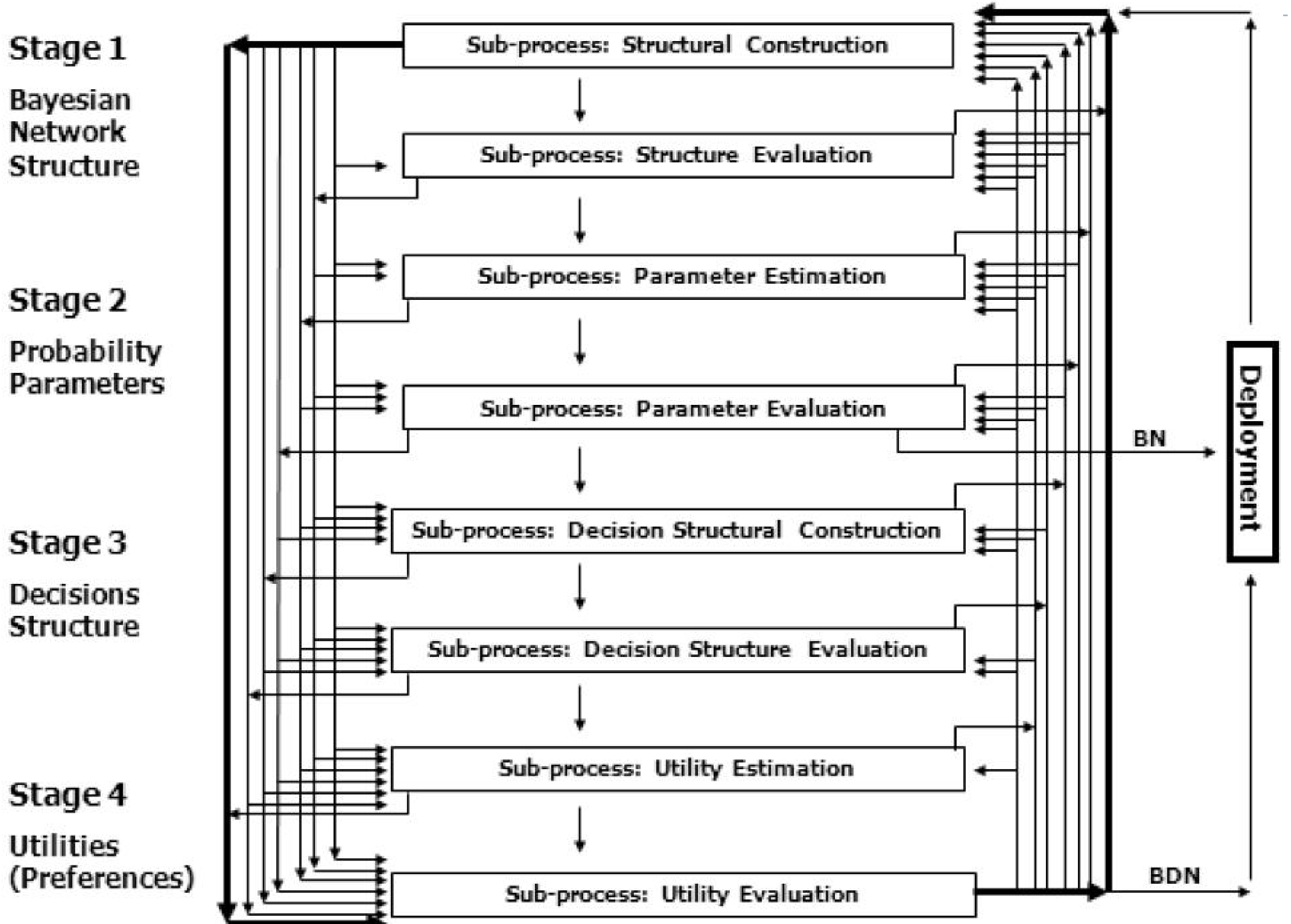
Iterative approach for KEBN

- An iterative and incremental approach for KEBN seems to be a better approach for us
- The software should grow by stages (prototypes) from childhood to adulthood, but at any given stage it is a self-sufficient, if limited, organism.
- Prototypes are functional implementations of software: they accept real input, such as the final system can be expected to deal with, and produce output of the type end-users will expect to find in the final system

A spiral model for KEBN (Korb, Nicholson, 2011)



Iterative lifecyle model for KEBN (Boneh, 2010)



Common mistakes

KEBN aspect	Mistake
The Process	Parameterizing before evaluating structure Trying to build the full model all at once
The Problem	Not understanding the problem context Complexity without value
Structure - Nodes	Getting the node values wrong Node values aren't exhaustive Node values aren't mutually exclusive Incorrect modeling of mutually exclusive outcomes Trying to model fuzzy categories Confusing state and probability Confusion about what the node represents
Structure - Arcs	Getting the arc directions wrong (a) Modeling reasoning rather than causation (b) Inverting cause and effect (c) Missing variables Too many parents
Parameters	Experts' estimates of probabilities are biased (a) Overconfidence (b) Anchoring (c) Availability Inconsistent "filling in" of large CPTs Incoherent probabilities (not summing to 1) Being dead certain

Other problems – Stage 3: Decision Structure

- First, we must model what decisions can be made, through the addition of one or more decision nodes.
- If the decision task is to choose only a single decision at any one time from a set of possible actions, only one decision node is required
- Combinations of actions can be modeled within the one node, for example, by explicitly adding a sequence of actions (ex. "surgery-medication")
  - This modeling solution avoids the complexity of multiple decision nodes, but has the disadvantage that the overlap between different actions
- Alternatively, you can use precedence links or Dynamic BN as you have seen, but you will have to deal with additional complexity!

Other problems – Stage 4: Utilities (Preferences)

- The next KE task for decision making is to model the utility of outcomes.
- The first stage is to decide what the unit of measure ("utile") will mean. Remember that money is not equal to utility (but it is related!)
- Remember the process to evaluate utilities of situation through lotteries as we have seen!

Stage 1: Bayesian Network Structure

- Common Modeling Mistake: Not understanding the problem context

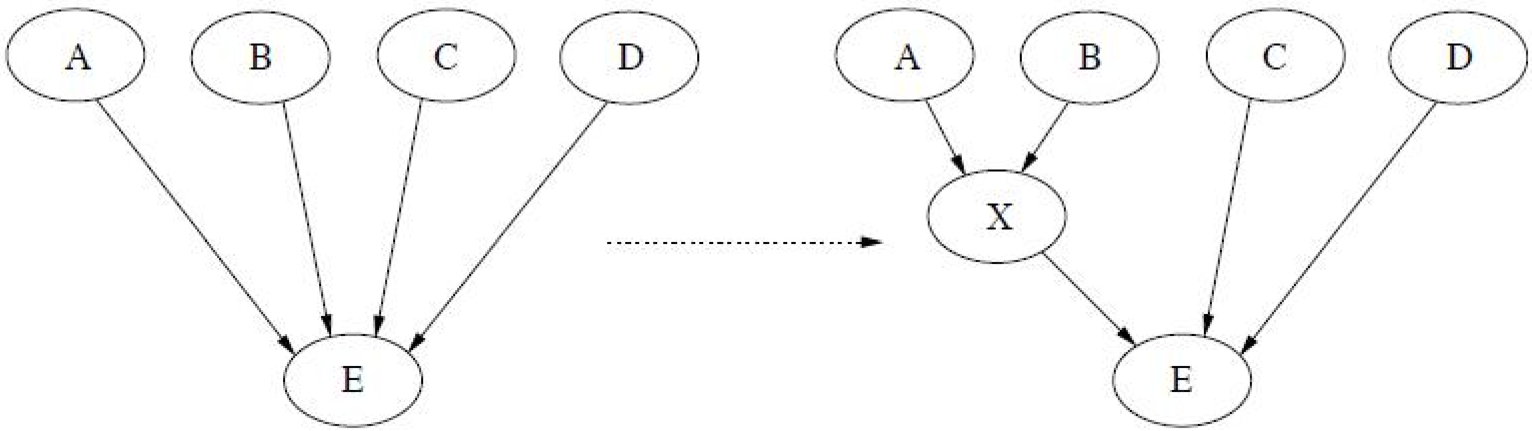
- It is crucial for the knowledge engineer to gain a clear understanding of the problem context. Ideally, this should be available in some form of project description. The knowledge engineer should ask questions like:

Q: *"What do you want to reason about?"*
Q: *"What don't you know?"*
Q: *"What information do you have?"*
Q: *"What do you know?"*

Stage 1: Bayesian Network Structure

- Other common mistake: Getting the arc directions wrong
  - (a) Modeling reasoning rather than causation
  - (b) Inverting cause and effect
  - (c) Missing variables
- Too many parentes
  - It is usually worse to have many parents than more parents. Modeling new nodes may help...

Reducing parents by intermediate nodes



Stage 1: Bayesian Network Structure

- Complexity without value
  - A very common impulse, when something is known about the problem, is to want to put it in the model.
  - But It may add complexity to the model without adding any value (and in fact often reduces value)
- Instead, the knowledge engineer must focus on the question:

Q: *"Which of the known variables are* most *relevant to the problem?"*

Stage 2: Probability Parameters

- Experts' estimates of probabilities are biased, including
  - Overconfidence: the tendency to attribute higher than justifiable probabilities to events that have a probability sufficiently greater than 0.5.
  - Anchoring: the tendency for subsequent estimates to be biased by an initial estimate (Kahneman and Tversky, 1973)
  - Availability: Assessing an event as more probable than is justifiable, because it is easily remembered or more salient (Kahneman and Tversky, 1973)

Stage 2: Probability Parameters

- Inconsistent "filling in" of large CPTs
  - For example, the expert uses 0.99 for "almost certain" in one part of the CPT, and 0.999 in another. Or it may be inconsistency across the CPT; for example, using different distributions for combinations of parents that in fact are very similar
- Incoherent probabilities (not summing to 1)
- Being dead certain

Stage 2 - Prob. Parameters - Another commons mistakes in structures

- Getting the node values wrong
- Node values aren’t exhaustive
- Node values aren’t mutually exclusive
- Incorrect modeling of mutually exclusive outcomes
  - Creation of separate nodes for different states of the same variable. For example, create both a FineWeather variable and a WetWeather variable (both Boolean). They are mutually exclusive!
- Trying to model fuzzy categories
- Confusing state and probability
- Confusion about what the node represents

State 2 - Prob. parameters - Discretization

- While it is possible to build BNs with continuous variables without discretization, the simplest approach is to discretize them

- Indeed, many of the current BN software tools available require this. They provides a choice between its doing the discretization for you crudely, (into even-sized chunks) or allowing the knowledge engineer more control over the process

Iterative lifecyle model for KEBN (Boneh, 2010)