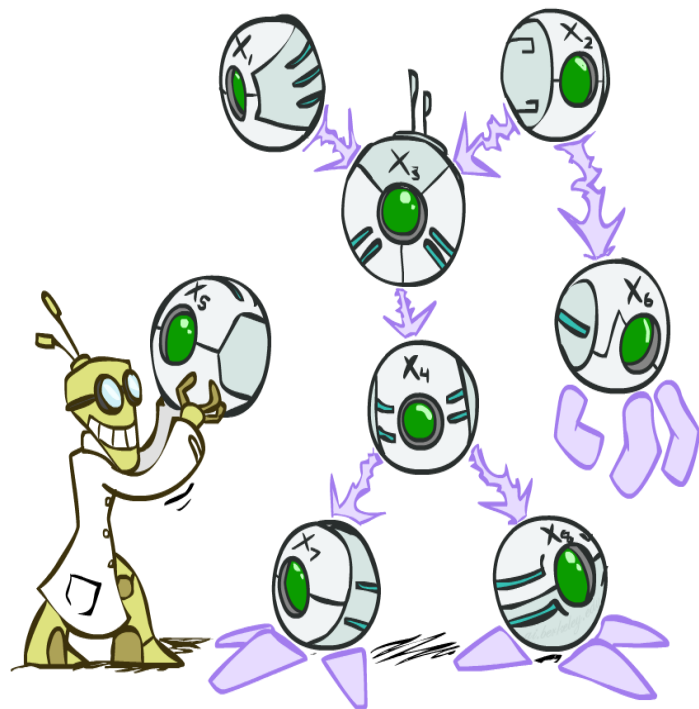


今天的内容

- 独立性(Independence)
- 条件无关性(Conditional independence)
- 贝叶斯网络(Bayes nets)
 - 语法和语义

贝叶斯网络(BAYES NETS)



概率基础总结

- 基本法则: $0 \leq P(\omega) \leq 1$ $\sum_{\omega \in \Omega} P(\omega) = 1$
- 事件: Ω 的子集: $P(A) = \sum_{\omega \in A} P(\omega)$
- 随机变量 $X(\omega)$ 对于每个 ω 有一个值
 - 分布 $P(X)$ 对每一个 x 值给出一个概率
 - 联合分布 $P(X,Y)$ 对每个 x,y 的组合给出概率

总结继续

■ 求和消除: $P(X=x) = \sum_y P(X=x, Y=y)$

■ 条件概率分布: $P(X|Y) = P(X, Y) / P(Y)$

通过列举法概率推理: $P(Q|e_1, \dots, e_k) = \alpha \sum_{h_1, \dots, h_m} P(Q, e_1, \dots, e_k, h_1, \dots, h_m)$

■ 这里 α (即是 $1/Z$) 是一个正规化因子, 使得 $P(Q|\dots)$ 之和为 1

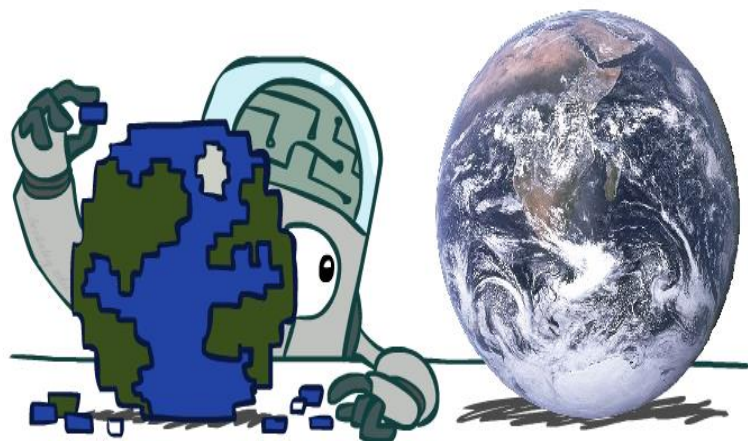
■ 乘法规则: $P(X|Y)P(Y) = P(X, Y) = P(Y|X)P(X)$

■ 推广到连锁法则: $P(X_1, \dots, X_n) = \prod_i P(X_i | X_1, \dots, X_{i-1})$

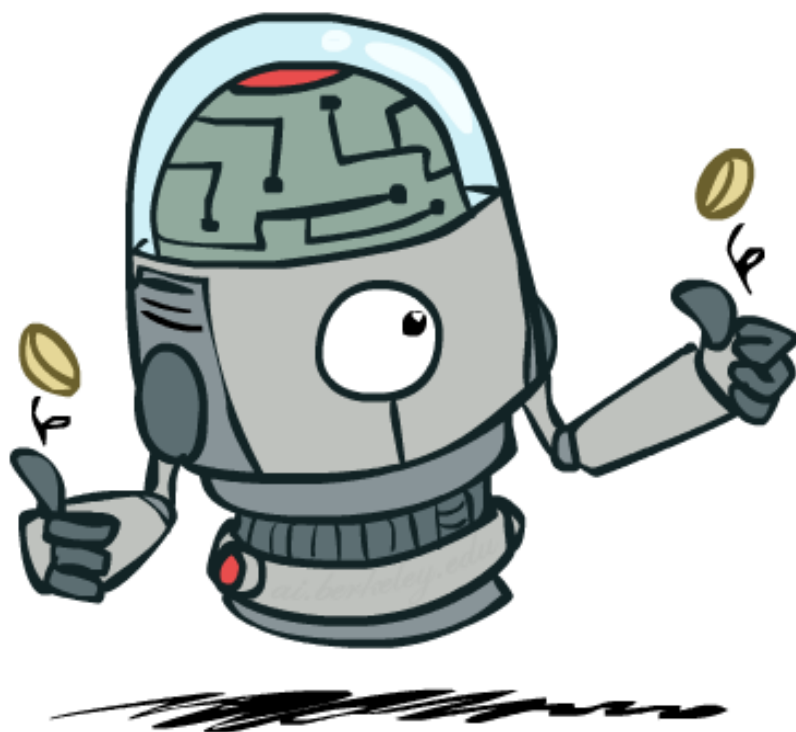
■ 贝叶斯规则: $P(X|Y) = P(Y|X)P(X) / P(Y)$

概率模型(Probabilistic Models)

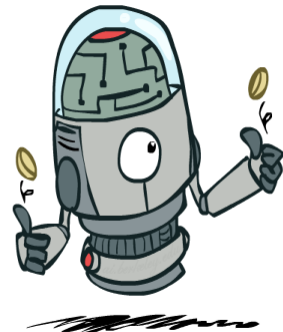
- 模型描述的是世界（或某一部分）是如何工作的
- 模型总是一种简化
 - 可能忽略了某些变量和它们之间的交互关系
 - “所有模型都是错的; 但某些是有用的.”
– George E. P. Box
- 概率模型能用来做什么?
 - 我们(或我们的人工智能体) 需要对未知变量进行推理, 当给定一些证据后
 - 例如: 解释 (诊断推理)
 - 例如: 预测 (因果推理)
 - 例如: 基于期望利益值的决策
- 如何建立模型, 并避免 d^n 的复杂性?



独立性



独立性



- 两个变量 X 和 Y 是 **独立的** 如果

$$\forall x, y \quad P(x, y) = P(x)P(y)$$

- 这说明他们的联合分布 **因式分解** 两个简单的分布的之乘积

- 结合乘法规则: $P(x, y) = P(x|y)P(y)$ 我们可以获得另一种形式:

$$\forall x, y \quad P(x|y) = P(x) \quad \text{or} \quad \forall x, y \quad P(y|x) = P(y)$$

- 举例: 两个骰子 $Roll_1$ 和 $Roll_2$

- $P(Roll_1=5, Roll_2=5) = P(Roll_1=5)P(Roll_2=5) = 1/6 \times 1/6 = 1/36$

- $P(Roll_2=5 \mid Roll_1=5) = P(Roll_2=5)$

举例：独立性

n 个公平，独立的硬币翻转：

$P(X_1)$

H	0.5
T	0.5

$P(X_2)$

H	0.5
T	0.5

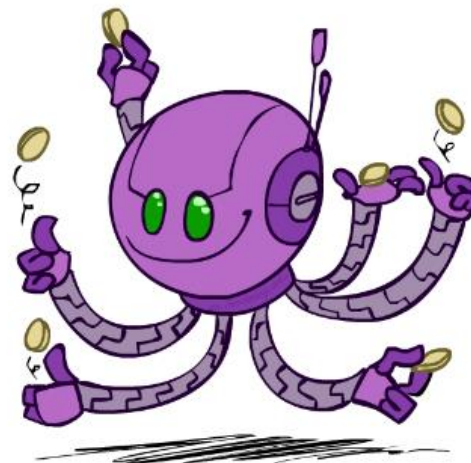
...

$P(X_n)$

H	0.5
T	0.5

2^n

$P(X_1, X_2, \dots, X_n)$



真实世界里的（概率事件）独立性

- 独立性是简化建模的假设
 - 有时对于真实世界的变量是合理的
 - 对于这些变量 {天气, 温度, 蛀牙, 牙疼}, 我们可以做什么样的假设?
 - 蛀牙和牙疼 **大致** 是独立于天气和温度的
 - 蛀牙和牙疼相互间 **不是** 独立的
 - 天气和温度相互间 **不是** 独立的

天气和温度（的独立性）？

计算边缘分布

$P_1(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(T)$

T	P
hot	0.5
cold	0.5

验证独立性，与
P1对比

$P_2(T, W)$

T	W	P
hot	sun	0.3
hot	rain	0.2
cold	sun	0.3
cold	rain	0.2

$P(W)$

W	P
sun	0.6
rain	0.4

举例：幽灵破坏者

- 变量和值域：

- G (幽灵位置) 在 $\{(1,1), \dots, (3,3)\}$

- $C_{x,y}$ (在方格 x,y 探测到的颜色；颜色越深离幽灵越近)：
 $\{\text{red}, \text{orange}, \text{yellow}, \text{green}\}$

- 假设我们有两个概率分布：

- **先验概率 (Prior distribution)** 幽灵的位置: $P(G)$

- 假设是均匀(uniform)分布

- **传感器模型 (Sensor model)**: $P(C_{x,y} \mid G)$ (只依赖于到 G 的距离)

- 例如 $P(C_{1,1} = \text{yellow} \mid G = (1,1)) = 0.1$

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

幽灵破坏者 计算: 第一步

当检测到 $C_{1,1} = \text{yellow}$ 时, 幽灵可能在哪儿?

给定证据, 概率分布 $P(G | C_{1,1} = \text{yellow})$ 是多少?

正规化
Normalize

可能性 Likelihood

先验
Prior

Bayes' rule:

$$P(G | C_{1,1} = \text{yellow}) = \frac{P(C_{1,1} = \text{yellow} | G) P(G)}{P(C_{1,1} = \text{yellow})}$$

○ $\propto P(C_{1,1} = \text{yellow} | G) P(G)$ (贝叶斯 **Bayesian** 更新)

0.11	0.11	0.11	0.17	0.10	0.10
0.11	0.11	0.11	0.09	0.17	0.10
0.11	0.11	0.11	0.01	0.09	0.17

幽灵破坏者 计算: 第二步

- 当看到 $C_{1,1} = \text{yellow}$, $C_{3,1} = \text{green}$, 幽灵在哪?
- 换句话说, $P(G \mid C_{1,1} = \text{yellow}, C_{3,1} = \text{green})$ 概率是什么?
- 我们的模型给出了 $P(C_{x,y} \mid G)$ 需要应用贝叶斯规则:

$$\begin{aligned} & \blacksquare P(G \mid C_{1,1}=y, C_{3,1}=g) \\ &= \alpha P(C_{1,1}=y, C_{3,1}=g \mid G) P(G) \\ &= \alpha P(C_{3,1}=g \mid C_{1,1}=y, G) P(C_{1,1}=y \mid G) P(G) \end{aligned}$$

- 条件后多了一项

应用乘法规则 | G

0.11	0.11	0.11	0.17	0.10	0.10	?	?	?
0.11	0.11	0.11	0.09	0.17	0.10	?	?	?
0.11	0.11	0.11	0.01	0.09	0.17	?	?	?

幽灵破坏者 计算: 第二步

- 如何计算 $P(C_{3,1}=g \mid C_{1,1}=y, G)$?
- 给定幽灵的位置, 在位置 1,1 观察到的黄色是否影响到在 3,1 为绿色的概率?
 - 不影响!
 - (只依赖于到幽灵的距离 distance from ghost)
- 在位置 3,1 的颜色是 **条件独立 (无关的)** (***conditionally independent***) 对于在位置 1,1 的颜色, 当给定幽灵的位置后
- $P(C_{3,1}=g \mid C_{1,1}=y, G) = P(C_{3,1}=g \mid G)$

幽灵破坏者 计算: 第二步

■ 观察到 $C_{1,1} = \text{yellow}$, $C_{3,1} = \text{green}$ 后, 幽灵的位置在哪?

■ $P(G \mid C_{1,1} = \text{yellow}, C_{3,1} = \text{green})$?

■ 我们的模型给出 $P(C_{x,y} \mid G)$ 只需应用 Bayes' rule:

■ $P(G \mid C_{1,1}=y, C_{3,1}=g)$

$$= \alpha P(G) P(C_{1,1}=y, C_{3,1}=g \mid G)$$

$$= \alpha P(G) P(C_{1,1}=y \mid G) P(C_{3,1}=g \mid C_{1,1}=y, G)$$

$$= \alpha P(G) P(C_{1,1}=y \mid G) P(C_{3,1}=g \mid G)$$

依据 $C_{3,1}$ 和 $C_{1,1}$ 条件独立
(无关性), 当给定 G 后

距离	$P(\text{green} \mid \text{距离})$
0	0.01
1	0.1
2	0.2
3	0.3
4	0.4

0.11	0.11	0.11	0.17	0.10	0.10	0.34	0.15	0.10
0.11	0.11	0.11	0.09	0.17	0.10	0.13	0.17	0.05
0.11	0.11	0.11	0.01	0.09	0.17	0.01	0.04	0.01

条件独立性（条件无关）

Conditional Independence

- 无条件的 (绝对的) 独立性非常稀少 (为什么?)
- 条件独立性是我们对于不确定环境的最基本和鲁棒的知识蕴藏形式
- X 是 条件独立于 (conditionally independent) Y, 给定 Z
当且仅当:

$$\forall x, y, z \quad P(x \mid y, z) = P(x \mid z)$$

或, 等价地, 当且仅当

$$\forall x, y, z \quad P(x, y \mid z) = P(x \mid z)P(y \mid z)$$

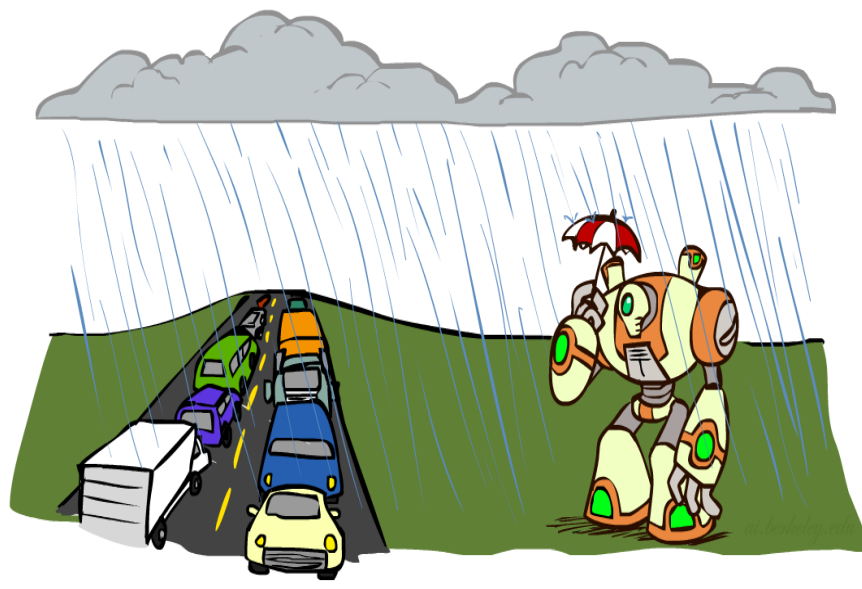
条件独立性（举例）

■ 关于以下环境的独立性是怎样的：

■ 交通流量

■ 雨伞

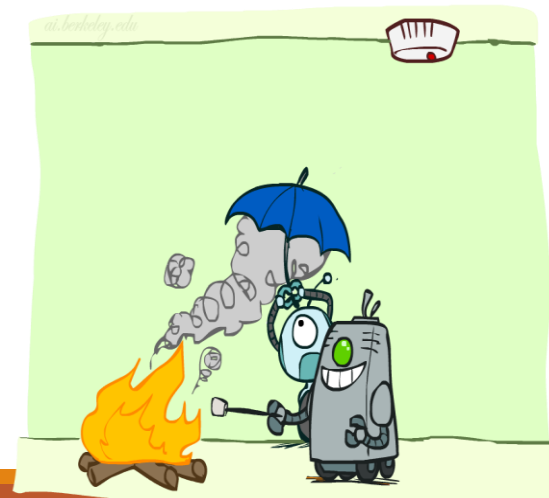
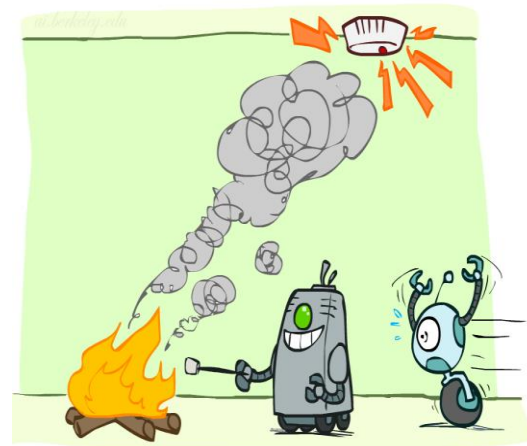
■ 下雨



条件独立性

■ 关于以下环境的独立性是怎样的：

- 燃火
- 冒烟
- 报警器



条件独立性与连锁法则(Chain Rule)

■ 连锁法则

$$P(x_1, x_2, \dots, x_n) = \prod_i P(x_i \mid x_1, \dots, x_{i-1})$$

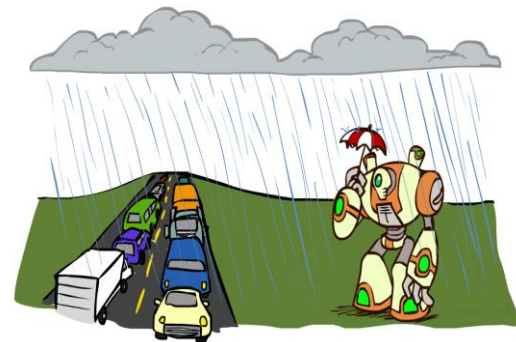
■ 简单的分解:

$$P(\text{Rain}, \text{Traffic}, \text{Umbrella}) = P(\text{Rain}) P(\text{Traffic} \mid \text{Rain}) P(\text{Umbrella} \mid \text{Rain}, \text{Traffic})$$

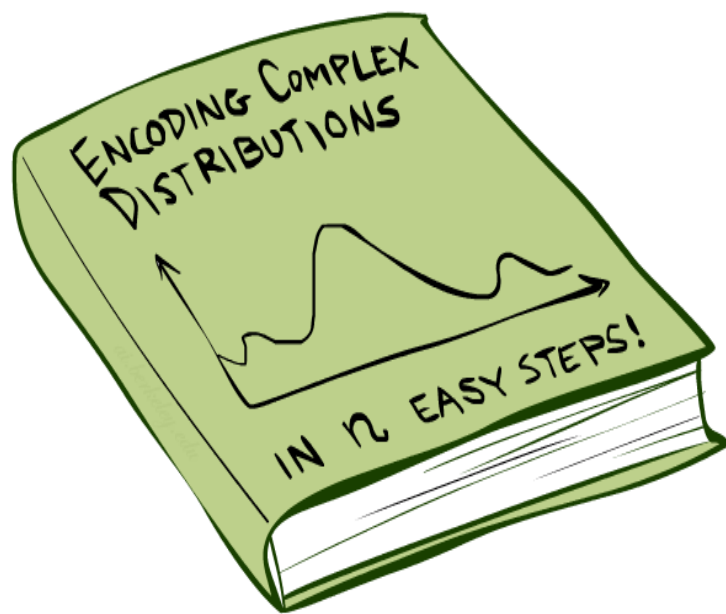
■ 利用了条件独立性的假设后:

$$P(\text{Rain}, \text{Traffic}, \text{Umbrella}) = P(\text{Rain}) P(\text{Traffic} \mid \text{Rain}) P(\text{Umbrella} \mid \text{Rain})$$

■ 贝叶斯网络 / 图形模型 帮助表达条件独立性的假设



贝叶斯网络(Bayes Nets)



贝叶斯网络

- 利用联合概率:

$$P(a | e) = \frac{1}{Z} P(a, e) = \frac{1}{Z} \sum_b \sum_c \sum_d P(a, b, c, d, e)$$

- 联合概率表可能很大

- 利用链锁法则

$$P(A, B, C, D, E) = P(A) P(B|A) P(C|A, B) P(D|A, B, C) P(E|A, B, C, D)$$

- 利用贝叶斯网络

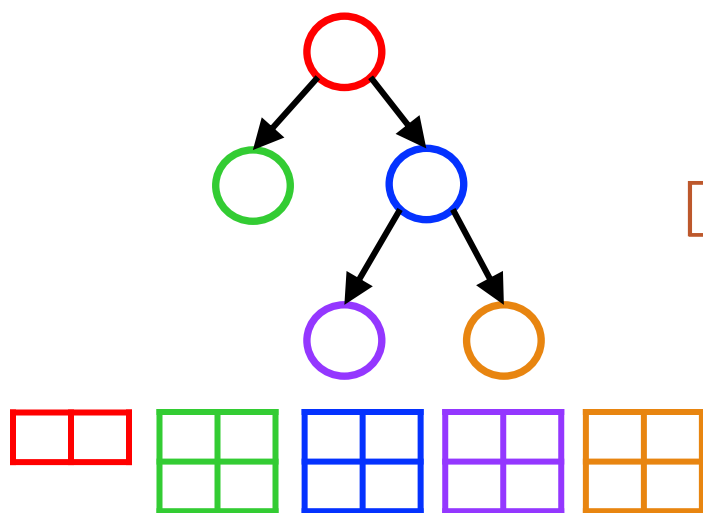
$$P(A, B, C, D, E) = P(A) P(B|A) P(C|A) P(D|C) P(E|C)$$

贝叶斯网络

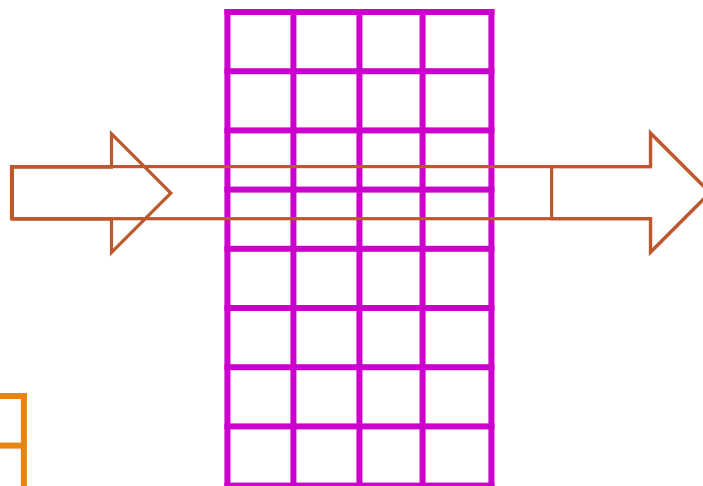
利用贝叶斯网络：

$$P(A, B, C, D, E) = P(A) P(B|A) P(C|A) P(D|C) P(E|C)$$

贝叶斯网络



联合分布

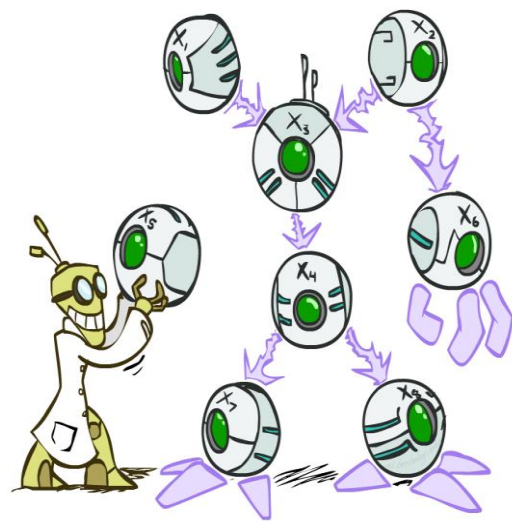


查询

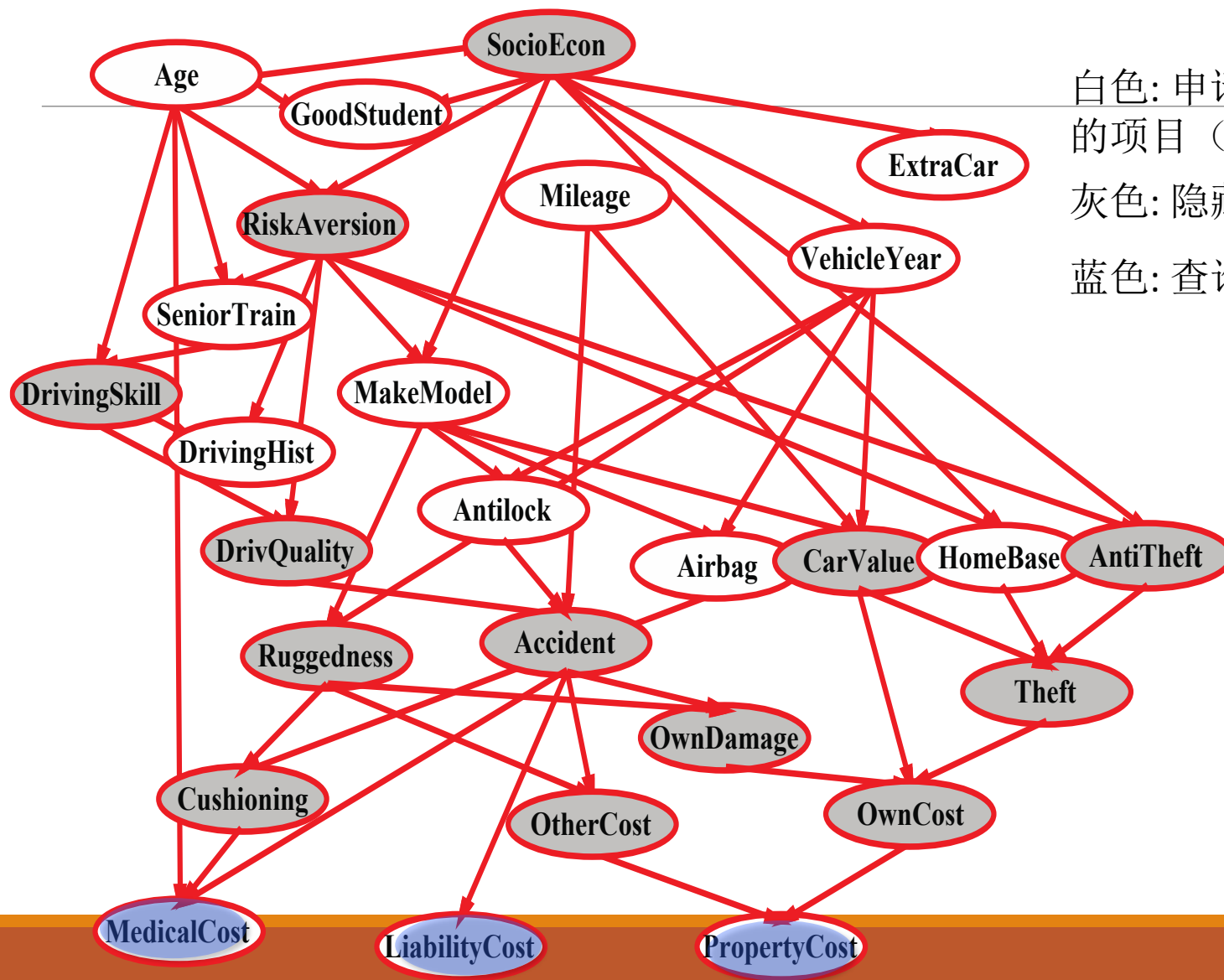
$$P(a | e)$$

贝叶斯网络: 宏观介绍

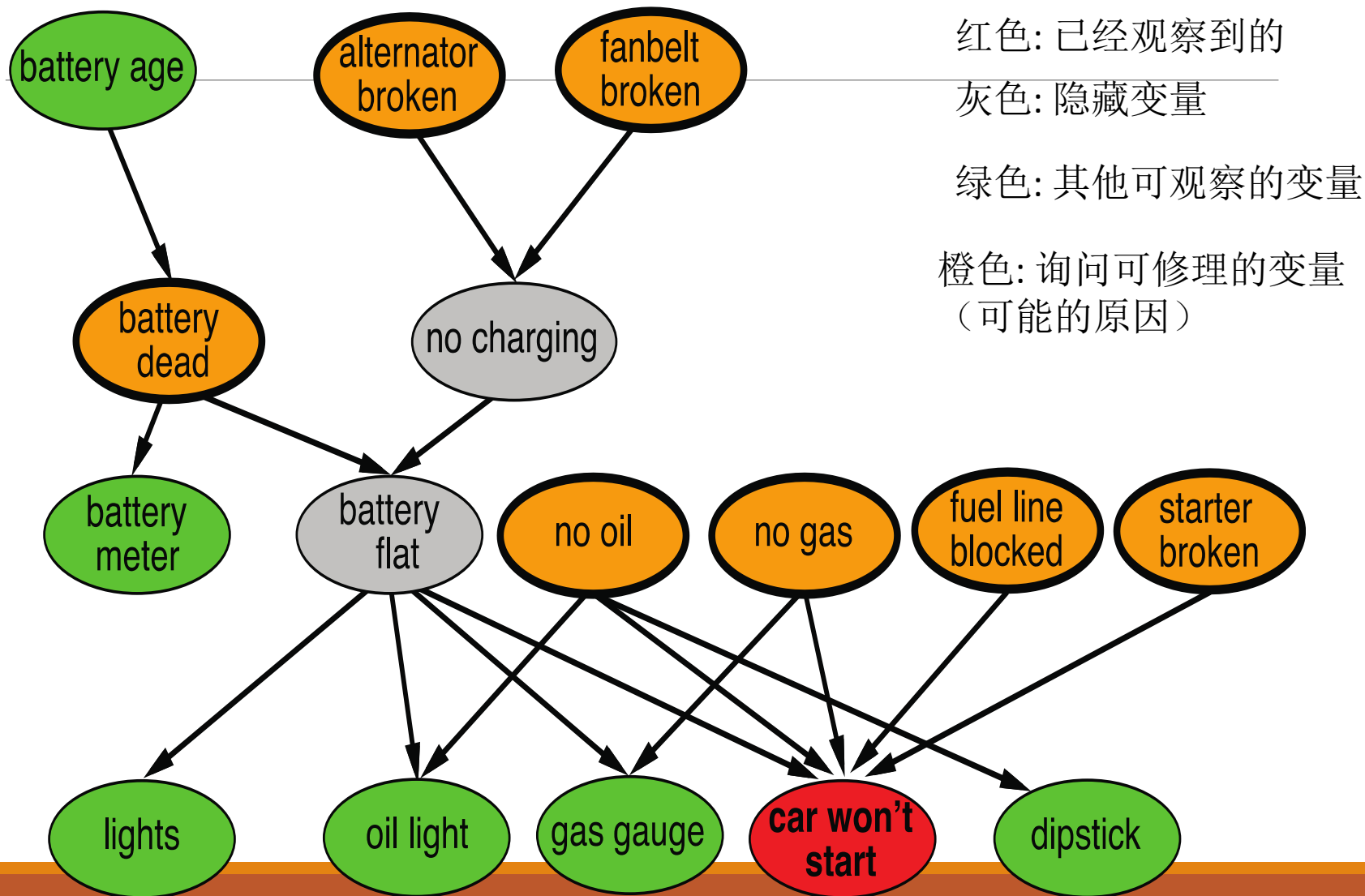
- 完全的联合分布表可以回答每一个问题, 但是:
 - 表的大小是变量数的指数级
 - 需要大量的例子来学习相应的概率
 - 用列举法 (加和消掉隐藏变量) 进行推理太慢
- 贝叶斯网络(Bayesian networks):
 - 表达了一个由变量组成的领域里所有的条件独立性关系
 - 联合分布因式分解为小规模条件概率分布的乘积
 - 分布表达的量级从指数减少为线性
 - 从较少的例子快速学习出模型
 - 快速推理 (在某些重要实例里可以达到线性时间复杂度)
 - “Microsoft’s competitive advantage lies in its expertise in Bayesian networks”
-- Bill Gates, quoted in LA Times, 1996



贝叶斯网络举例：汽车保险



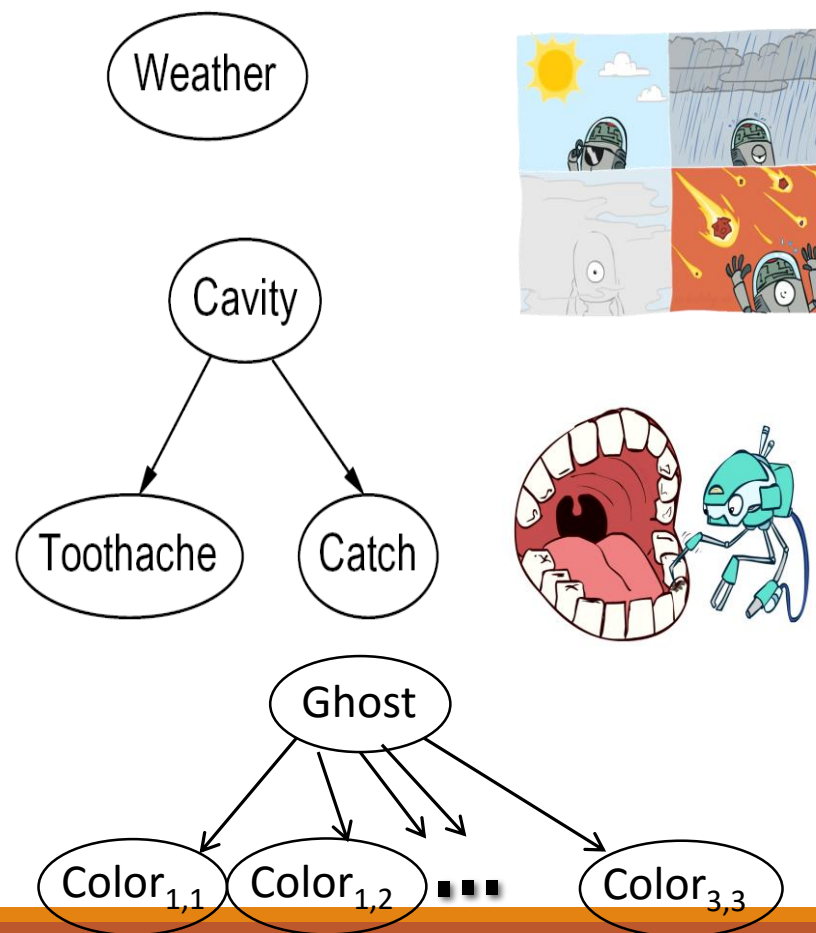
贝叶斯网络举例：简单汽车维修



图形模型的表达

- 节点: 变量 (每个变量有个值域)
- 弧/边: 相互作用
 - 指示 变量间的“直接的影响”
 - 正式的意思: 代表条件独立性
- 现在可以简单地: 箭头意味着直接的因果关系 (通常情况下, 它们并不一定是这样!)

Catch, 这里指的是牙医的探测器是否捕获到你的牙是否出问题; 有可能会有遗漏。



举例：硬币上抛翻转

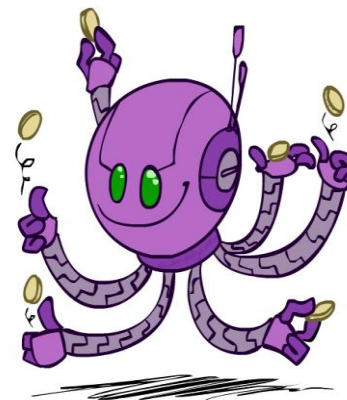
■ N 次硬币上抛（也可以理解为N个硬币抛一次）

X_1

X_2

...

X_n



■ 变量间没有相互作用：绝对独立（没有边连接）

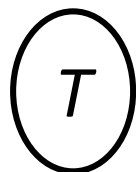
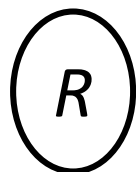
举例：交通流量预测

■ 变量：

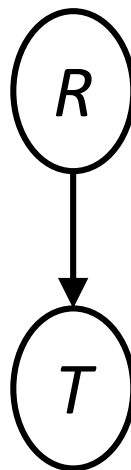
■ R : 下雨

■ T : 交通状况

■ 模型 1: 独立的



■ 模型 2: 下雨导致交通状况变化



■ 为什么用模型 2 更好？



贝叶斯网络语法和语义





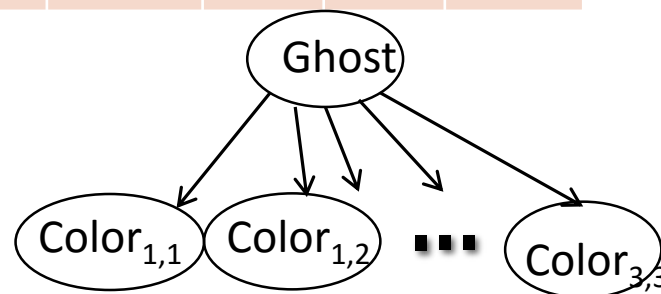
贝叶斯网络语法

- 一个节点对应一个变量 X_i
- 一个有向, 无环图
- 对每个节点 给定图中它的 **父节点** 有一个条件概率分布,
 - **CPT**: 条件概率分布表:
在给定父节点的一个配置以后,
每一行是子节点取值的一个分布
- 一个近似的“因果”过程的描述

P(Ghost)

(1,1)	(1,2)	(1,3)	...
0.11	0.11	0.11	...

Ghost	P(Color _{1,1} Ghost)			
	g	y	o	r
(1,1)	0.01	0.1	0.3	0.59
(1,2)	0.1	0.3	0.5	0.1
(1,3)	0.3	0.5	0.19	0.01
...				

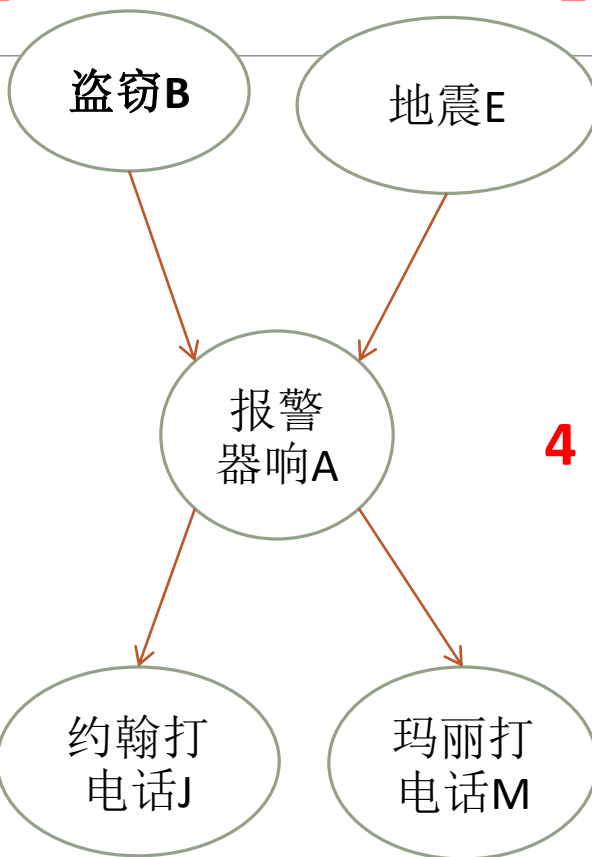


贝叶斯网络 = 拓扑结构(图形) + 局部条件概率

举例: 报警器网络

P(B)	
true	false
0.001	0.999

1



1

P(E)	
true	false
0.002	0.998

4

B	E	P(A B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J A)	
	true	false
true	0.9	0.1
false	0.05	0.95

2

A	P(M A)	
	true	false
true	0.7	0.3
false	0.01	0.99

2



条件概率分布表CPT
的自由参数的个数
总共有:

父变量的值域大小:

d_1, \dots, d_k

子变量的值域为 d
表中每一行概率值
之和为 1

$$(d-1) \prod_i d_i$$

对于稀疏的贝叶斯网络(BNs), 通用的规模计算公式

■ 假定:

- n 个变量

- 最大值域大小是 d

- 最大父节点数是 k

■ 完全的联合分布的规模: $O(d^n)$

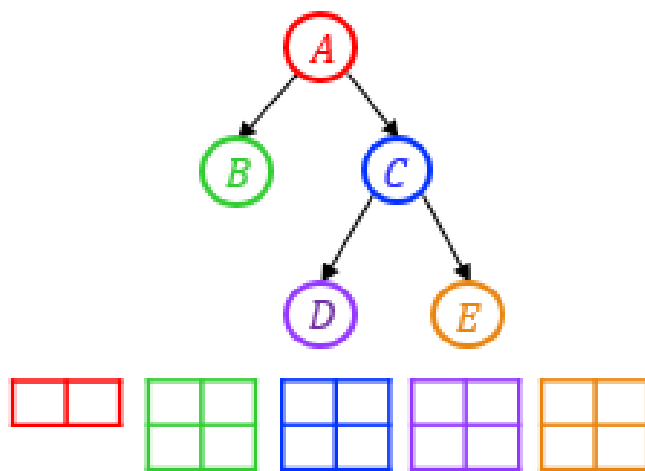
■ 贝叶斯网络的规模: $O(n \cdot d^k)$

- n 的线性比例(只要因果结构是局部的)

贝叶斯网络语法回顾

- 每个随机变量用一个节点来代表
- 有向无环图
- 每个节点有一个条件概率分布表
 - $P(\text{节点} \mid \text{该节点的父节点})$

贝叶斯网络：



贝叶斯网络的全局语法



- 贝叶斯网络整体表达了：
 - （编码）联合分布，作为每一个变量上条件分布的乘积：

$$P(X_1, \dots, X_n) = \prod_i P(X_i \mid \text{Parents}(X_i))$$

语义举例



联合概率分布因式分解举例

利用通用链式法则

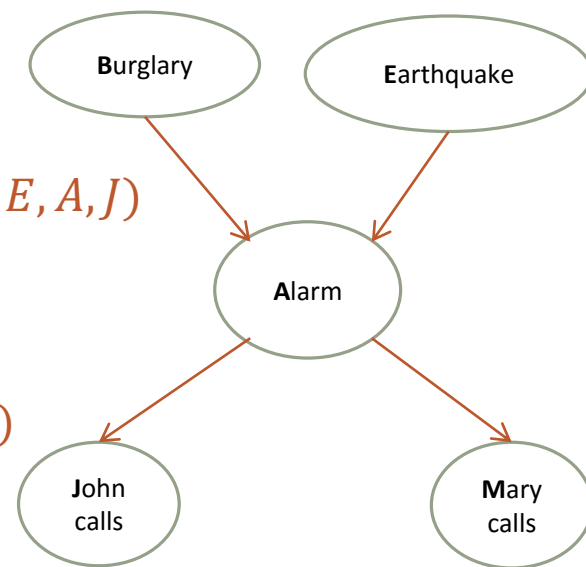
- $P(X_1 \dots X_n) = \prod_i P(X_i | X_1 \dots X_{i-1})$

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

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

贝叶斯网络

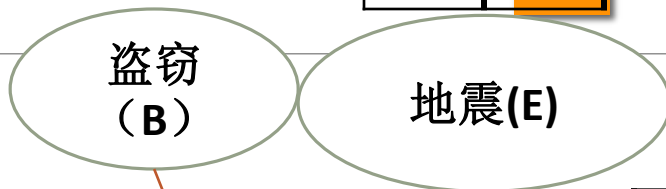
- $P(X_1 \dots X_n) = \prod_i P(X_i | \text{Parents}(X_i))$



举例

P(B)	
true	false
0.001	0.999

P(E)	
true	false
0.002	0.998



$$P(b, \neg e, a, \neg j, \neg m) =$$

$$P(b) P(\neg e) P(a|b, \neg e) P(\neg j|a) P(\neg m|a)$$

$$= .001 \times .998 \times .94 \times .1 \times .3 = .000028$$

B	E	P(A B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J A)	
	true	false
true	0.9	0.1
false	0.05	0.95

A	P(M A)	
	true	false
true	0.7	0.3
false	0.01	0.99

贝叶斯网络语法



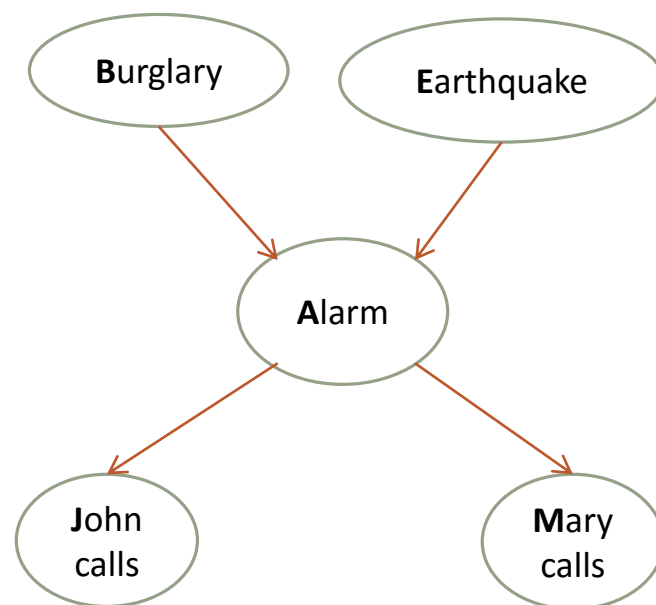
怎样采取这种捷径?

- 从链式法则

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

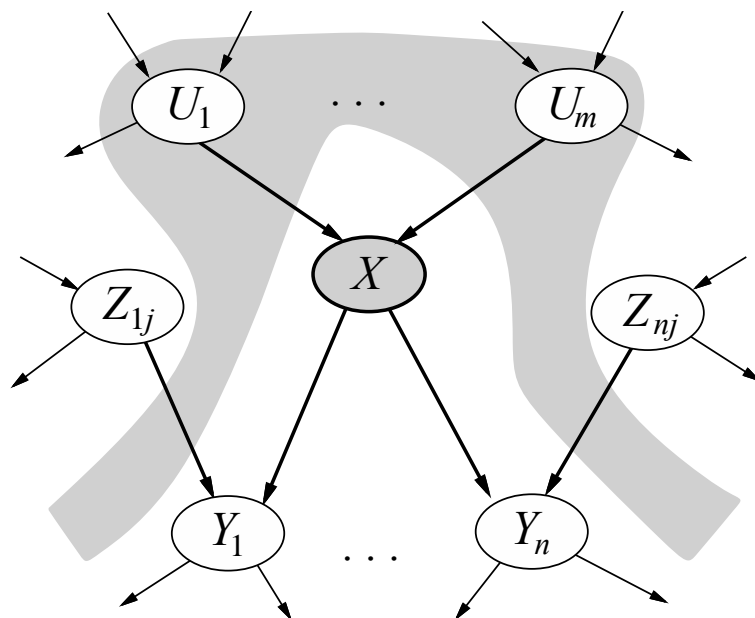
- 转为贝叶斯网络

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



条件独立性语义

当给定它的父节点取值后，每个变量都是条件独立于它的非后代变量



条件独立语法

对于下列贝叶斯网络, 写出联合分布 $P(A, B, C)$

1. 使用链式法则 (顺序为A,B,C)
2. 使用贝叶斯网络语法 (CPT 的乘积)



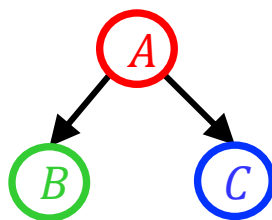
$$P(A) P(B|A) P(C|A, B)$$

$$P(A) P(B|A) P(C|B)$$

前提:

$$P(C|A, B) = P(C|B)$$

C 独立于A 当给定B后

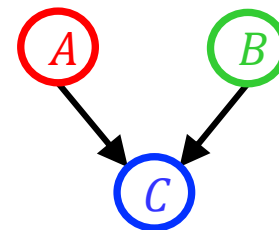


$$P(A) P(B|A) P(C|A, B)$$

$$P(A) P(B|A) P(C|A)$$

前提:

$$P(C|A, B) = P(C|A)$$



$$P(A) P(B|A) P(C|A, B)$$

$$P(A) P(B) P(C|A, B)$$

前提:

$$P(B|A) = P(B)$$

贝叶斯网络里的概率



- 为什么我们可以保证以下公式是正确的联合分布

$$P(X_1, \dots, X_n) = \prod_i P(X_i \mid \text{Parents}(X_i))$$

- 连锁法 (对所有分布有效): $P(X_1, \dots, X_n) = \prod_i P(X_i \mid X_1, \dots, X_{i-1})$

- 假定 条件独立性: $P(X_i \mid X_1, \dots, X_{i-1}) = P(X_i \mid \text{Parents}(X_i))$

- 当加入节点 x_i , 确保其父节点“屏蔽”它与其祖先节点的联系
- 给定它的父节点, 每个变量条件独立于它的非子孙节点变量

→ 结果: $P(X_1, \dots, X_n) = \prod_i P(X_i \mid \text{Parents}(X_i))$

- 所以, 网络的拓扑结构暗示着条件独立性

举例: 入室偷盗报警

■ 入室盗窃

■ 地震

■ 报警器

P(B)	
b	$\neg b$
0.001	0.999

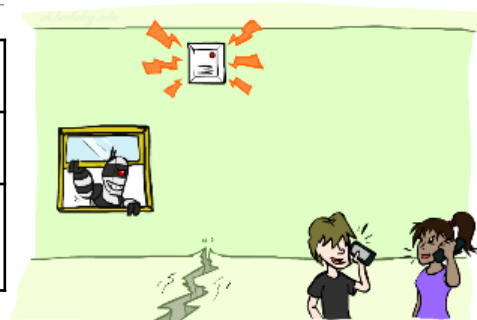
盗窃(B)

P(E)	
e	$\neg e$
0.002	0.998

地震(E)

报警器
响(A)

B	E	P(A B,E)	
		a	$\neg a$
b	e	0.95	0.05
b	$\neg e$	0.94	0.06
$\neg b$	e	0.29	0.71
$\neg b$	$\neg e$	0.001	0.999

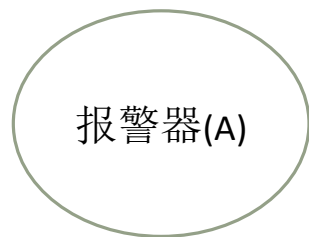


举例: 入室偷盗报警

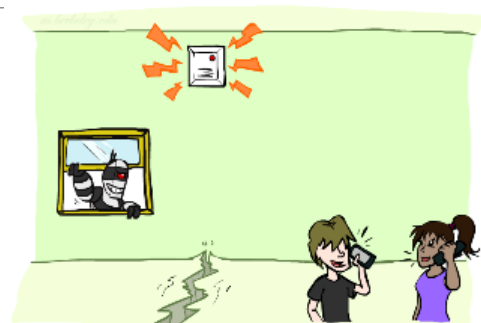
■ 报警器

■ 入室盗窃

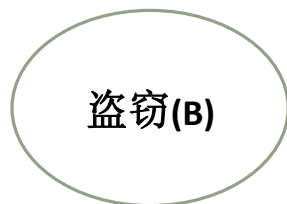
■ 地震



P(A)	
a	$\neg a$



A	P(B A)	
	b	$\neg b$
a	?	
$\neg a$		



A	B	P(E A,B)	
		e	$\neg e$
a	b	?	
a	$\neg b$		
$\neg a$	b		
$\neg a$	$\neg b$		

因果关系(Causality)?

■ 当贝叶斯网络反映了真实的因果关系模式时:

■ 通常更简单的网络 (较少的父节点, 较少的参数)

■ 通常更容易评估概率

■ 通常鲁棒性更强, 比如修改盗窃的频率后应该不影响模型里的其他部分!

■ BNs 不需要实际上表达因果关系

■ 有时没有因果网络存在于一个领域 (尤其是在一些变量丢失的情况下)

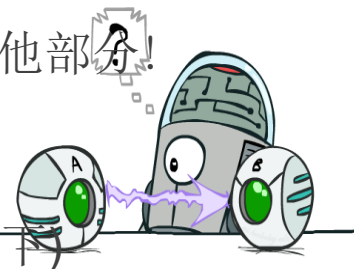
■ 其结果是箭头关联反映的是相关性 (correlation), 而不是因果关系

■ 箭头实际表示的是什么?

■ 拓扑结构可能碰巧表达的是因果关系

■ 拓扑结构真正表达 (编码) 的是条件独立性:

■
$$P(X_i | X_1, \dots, X_{i-1}) = P(X_i | \text{Parents}(X_i))$$



贝叶斯网络独立性举例

■ 给定报警器响，约翰打电话 是否 独立于 入室盗窃的发生？

■ 是的

■ 给定报警器响，约翰打电话 是否 独立于 玛丽打电话？

■ 是的

■ 盗窃 是否独立于 地震？

■ 是的

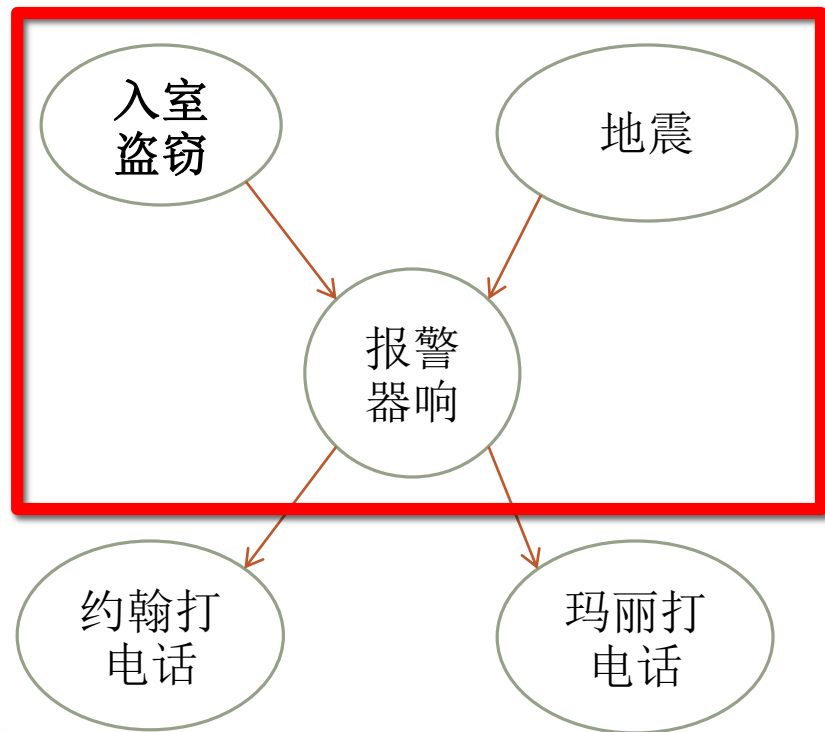
■ 当报警器响后, 盗窃 是否独立于 地震 ？

■ 不是！

■ 报警器已响，入室盗窃和地震都变得很有可能发生过

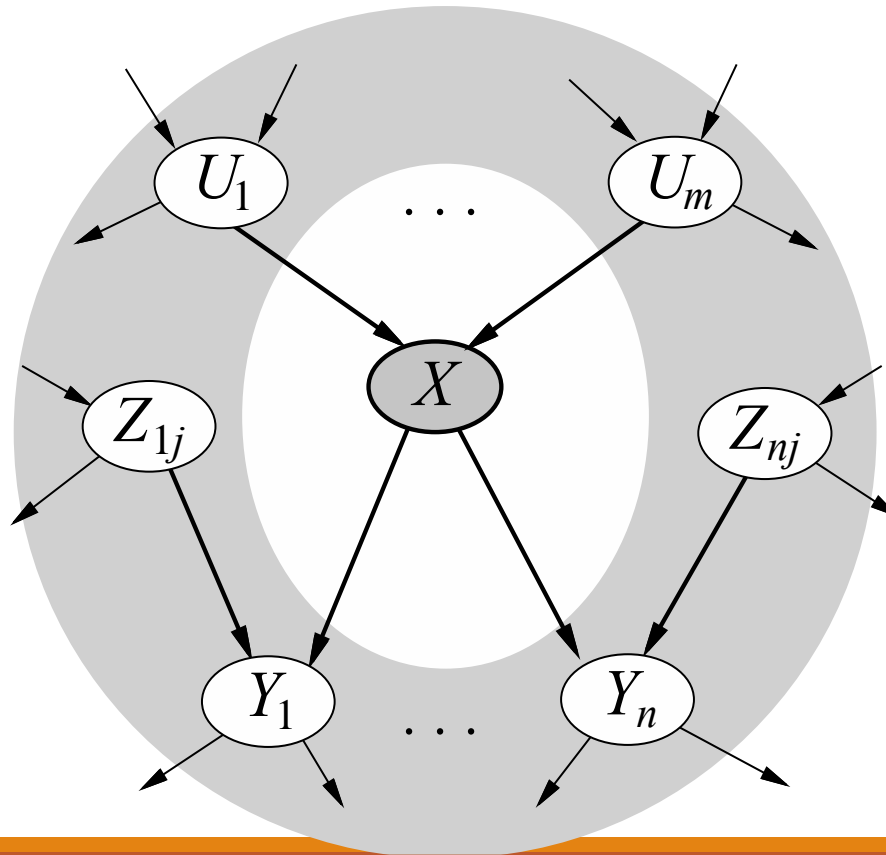
■ 但是，如果我们得知一个入室盗窃已经发生, 那么报警器响的原因被 **解释**，则地震发生的概率降低

V-结构



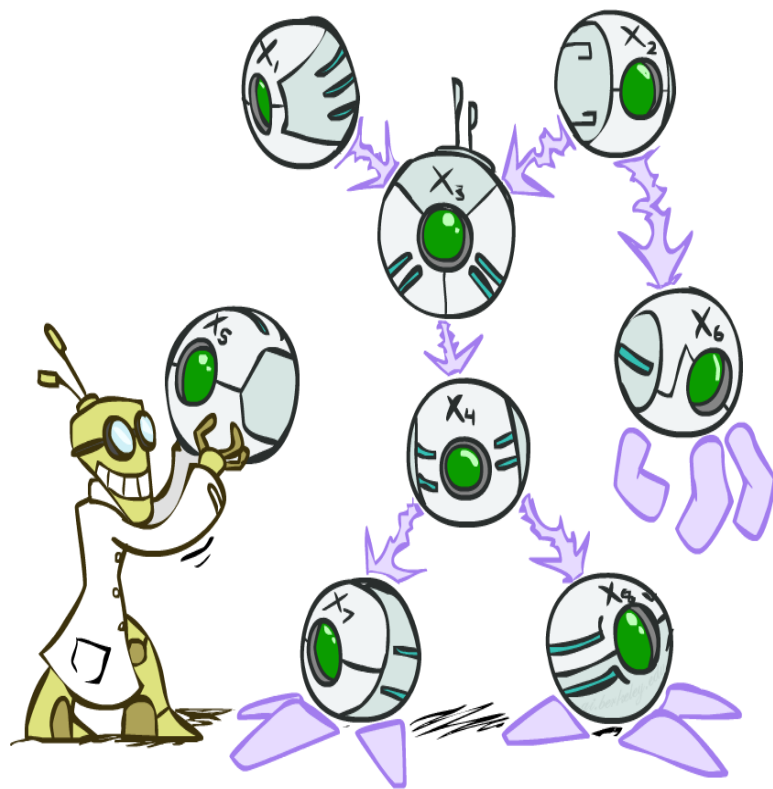
马尔科夫毯(Markov blanket)

- 一个变量的马尔可夫毯包括父节点, 子节点, 子节点的其他父节点
- 每个变量给定它的马尔科夫毯, 则是条件独立于所有其他变量



贝叶斯网络(Bayes Nets)

- 已经介绍: 贝叶斯网络如何实现了对联合分布的表达
- 下次: 如何回答查询, 计算查询变量在给定 (观察) 证据下的条件概率

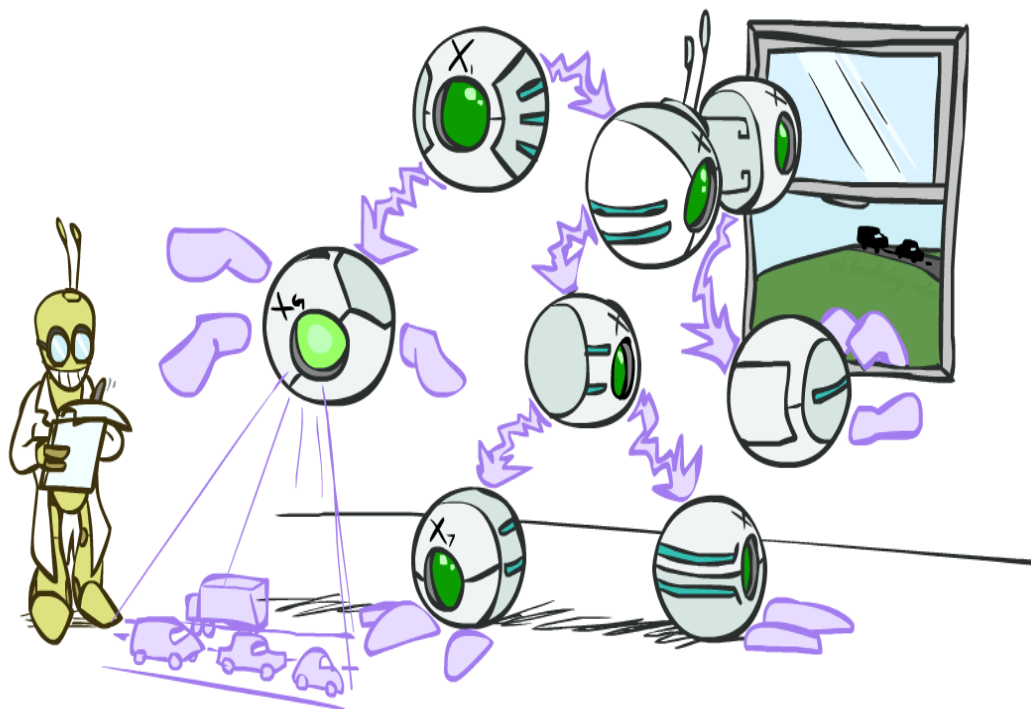


接下来的内容

■ 贝叶斯网络：精确推理

人工智能导论

贝叶斯网络：精确推理



贝叶斯网络 (Bayes Nets)



Part I: 表达

Part II: 精确推理 (Exact inference)

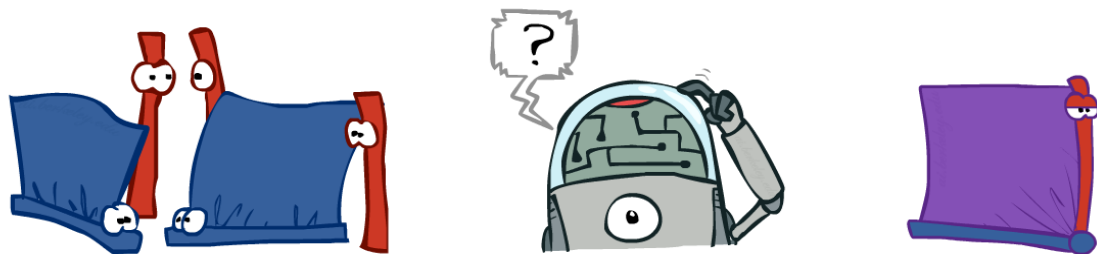
- 列举法 (总是指数复杂度)
- 变量消除法 (最差情况指数复杂度, 通常情况下更好)
- 推理问题是NP-hard

Part III: 近似推理 (Approximate Inference)

后面: 从数据学习构建网络结构

推理

- 从一个概率模型里（联合概率分布），
计算某些有用的数量
- 例如：
 - 后验边缘概率 (Posterior marginal probability)
 - $P(Q|e_1, \dots, e_k)$
 - 举例：给定一些症状，推理可能的疾病原因
 - 推理最有可能的解释是什么：
 - $\operatorname{argmax}_{q,r,s} P(Q=q, R=r, S=s | e_1, \dots, e_k)$



推理展望

随机变量 Q, H, E (询问, 隐藏, 证据)

我们知道如何在一个联合分布上做推理：

$$\begin{aligned} P(q|e) &= \alpha P(q, e) \\ &= \alpha \sum_{h \in \{h_1, h_2\}} P(q, h, e) \end{aligned}$$

我们知道贝叶斯网络能够分解联合分布成 CPTs

$$\begin{aligned} P(q|e) &= \alpha \sum_{h \in \{h_1, h_2\}} P(h) P(q|h) P(e|q) \\ &= \alpha [P(h_1) P(q|h_1) P(e|q) + P(h_2) P(q|h_2) P(e|q)] \end{aligned}$$

但是我们可以更有效率：

$$\begin{aligned} P(q|e) &= \alpha P(e|q) \sum_{h \in \{h_1, h_2\}} P(h) P(q|h) \\ &= \alpha P(e|q) [P(h_1) P(q|h_1) + P(h_2) P(q|h_2)] \\ &= \alpha P(e|q) P(q) \end{aligned}$$

现在可以扩展到更大的贝叶斯网络



列举法

变量消除法

通过列举法在贝叶斯网络里推理

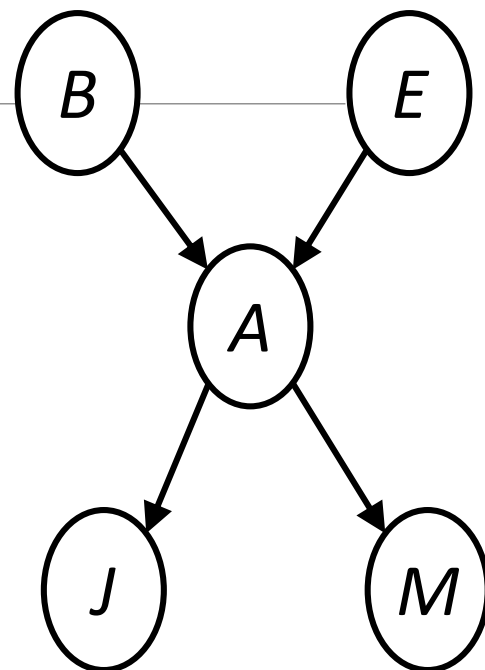
列举法推理回顾:

- 任何想要获知的概率值都可以通过加和(消除不相关的变量)联合概率分布里的项来计算出来
- 联合概率分布里的表项可以通过乘上贝叶斯网络里的相应的条件概率来计算获得

$$\begin{aligned} P(B \mid j, m) &= \alpha P(B, j, m) \\ &= \alpha \sum_{e,a} P(B, e, a, j, m) \\ &= \alpha \sum_{e,a} P(B) P(e) P(a \mid B, e) P(j \mid a) P(m \mid a) \end{aligned}$$

所以BN推理意味着对概率数乘积进行求和计算: 似乎很容易!!

问题: 要计算 指数增长的乘积项之和!



是否能做的更好?

■ 比如:

■ $x_1y_1z_1 + x_1y_1z_2 + x_1y_2z_1 + x_1y_2z_2 + x_2y_1z_1 + x_2y_1z_2 + x_2y_2z_1 + x_2y_2z_2$

■ 16 乘法, 7 个加法

■ 许多重复的子表达式!

■ 重写成:

■ $(x_1 + x_2)(y_1 + y_2)(z_1 + z_2)$

■ 2 乘法, 3 加法

■ $\sum_e \sum_a P(B) P(e) P(a|B, e) P(j|a) P(m|a)$ = $P(B) P(e) P(a|B, e) P(j|a) P(m|a)$
+ $P(B) P(\neg e) P(a|B, \neg e) P(j|a) P(m|a)$
+ $P(B) P(e) P(\neg a|B, e) P(j|\neg a) P(m|\neg a)$
+ $P(B) P(\neg e) P(\neg a|B, \neg e) P(j|\neg a) P(m|\neg a)$

■ 许多重复的子表达式!

变量消除法：基本思想

尽可能早的先做求和操作：

$$\begin{aligned} P(B|j, m) &= \alpha \sum_e \sum_a P(B, e, a, j, m) \\ &= \alpha \sum_e \sum_a P(j|a) P(e) P(m|a) P(a|B, e) P(B) \\ &= \alpha P(B) \sum_e P(e) \sum_a P(j|a) P(m|a) P(a|B, e) \end{aligned}$$

变量消除法: 基本思想

- 尽量把求和操作移到里面, 先消掉一些变量

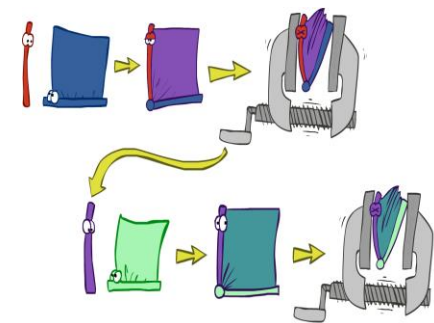
$$\begin{aligned} P(B \mid j, m) &= \alpha \sum_e \sum_a P(B) P(e) P(a \mid B, e) P(j \mid a) P(m \mid a) \\ &= \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) P(j \mid a) P(m \mid a) \end{aligned}$$

- 计算顺序由里向外

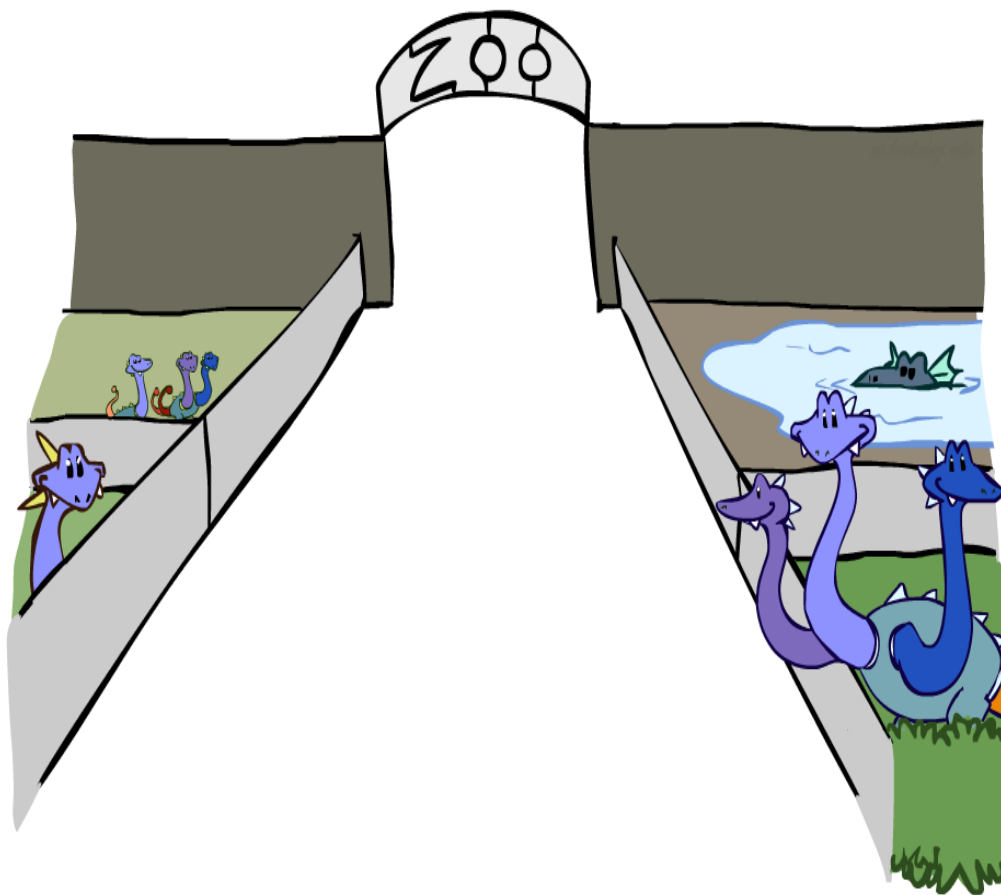
- 即, 先在 a 上求和, 再在 e 上求和

- 问题: $P(a \mid B, e)$ 不是单个数, 一组不同的数, 依赖于 B 和 e 的值

- 解决办法: 使用不同维度的数组, 以及相应的操作; 这些列表也叫作 **因子(factors)**



因子式的相关概念



因子概念I

■联合分布: $P(X,Y)$

- 表项 $P(x,y)$ 对任一 x, y
- $|X| \times |Y|$ 矩阵
- 表项之和 为 1

$P(A,J)$

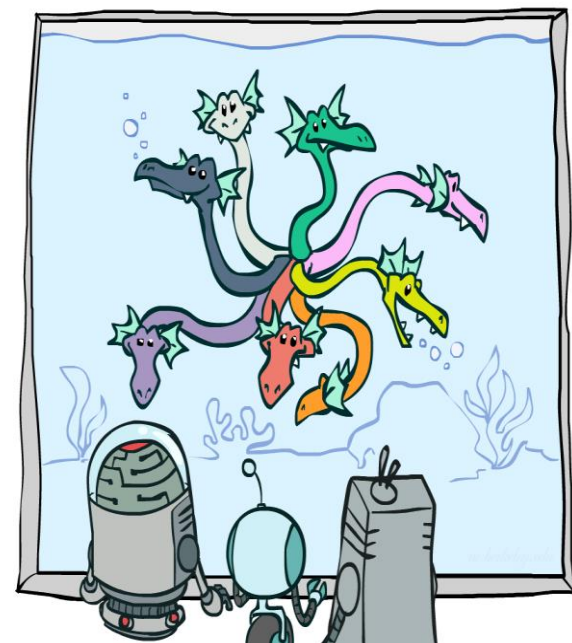
$A \setminus J$	true	false
true	0.09	0.01
false	0.045	0.855

■投射的联合分布概率： $P(x,Y)$

- 联合分布的一部分
- 表项 $P(x,y)$ 对于一个 x 值, 所有的 y 值
- $|Y|$ -个元素的向量数组
- 数组之和为 $P(x)$

$P(a,J)$

$A \setminus J$	true	false
true	0.09	0.01



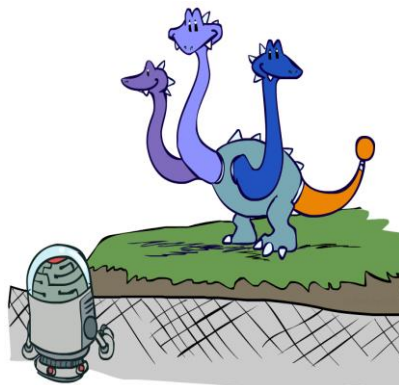
变量数之和 (大写的字符) = 表的维度

因子概念II

■ 单条件概率: $P(Y | x)$

■ 表项 $P(y | x)$, 对于固定的 x 值, 所有的 y 值

■ 表项之和为 1



$$P(J | a)$$

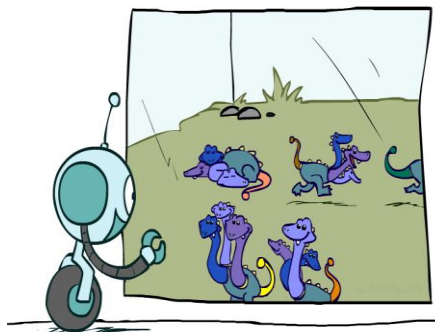
$A \setminus J$	true	false
true	0.9	0.1

■ 条件概率家族: $P(X | Y)$

■ 多个条件概率

■ 表项 $P(x | y)$, 对于所有 x, y

■ 表项之和为 $|Y|$



$$P(J | A)$$

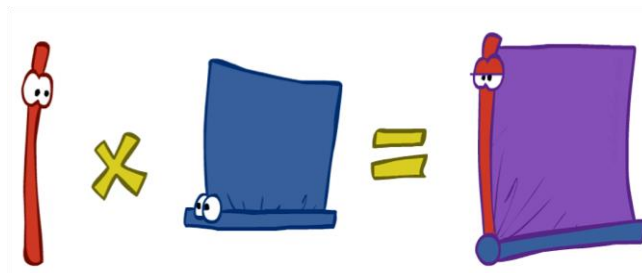
$A \setminus J$	true	false
true	0.9	0.1
false	0.05	0.95

} - $P(J | a)$

} - $P(J | \neg a)$

操作 1: 逐点乘积 (Pointwise product)

- 第一个基本操作: 因子的 **逐点乘积** (类似于一个数据库的联合 (join) 操作, **不是** 矩阵相乘!)
- 新的因子包含两个原始因子变量的**合集**
- 每个表项是原始因子相应项的乘积



- 例如: $P(J|A) \times P(A) = P(A,J)$

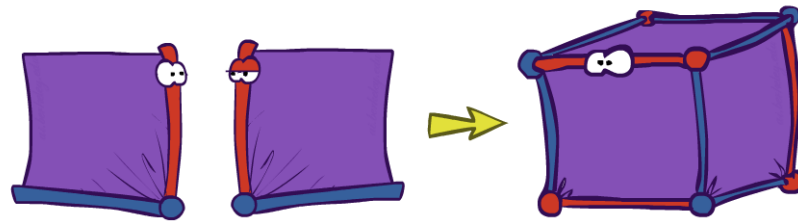
$P(A)$				$P(J A)$				$P(A,J)$	
true	0.1	\times	$=$	A \ J	true	false	A \ J	true	false
false	0.9			true	0.9	0.1	true	0.09	0.01
				false	0.05	0.95	false	0.045	0.855

逐点乘积举例

A	B	$\mathbf{f}_1(A, B)$	B	C	$\mathbf{f}_2(B, C)$	A	B	C	$\mathbf{f}_3(A, B, C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C) = \mathbf{f}_3(A, B, C)$.

举例：产生更大因子



■ 例如: $P(J/A) \times P(M/A) = P(J,M/A)$

$P(J/A)$

A \ J	true	false
true	0.99	0.01
false	0.145	0.855

\times

$P(M/A)$

A \ M	true	false
true	0.97	0.03
false	0.019	0.891

$=$

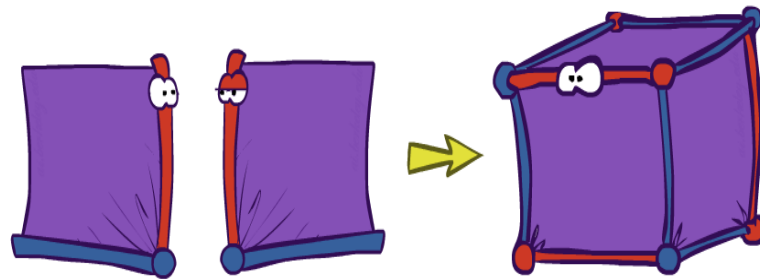
$P(J,M/A)$

		J \ M	true	false	
		J \ M	true	false	
	true				18
	false			.0003	

A=false

A=true

举例：产生更大因子



- 例如: $f_1(U,V) \times f_2(V,W) \times f_3(W,X) = f_4(U,V,W,X)$
- 因子表尺寸: $[10,10] \times [10,10] \times [10,10] = [10,10,10,10]$
- 即, 300 表项 增大为 10,000 个表项!
- 因子的迅速膨胀会导致变量相除法变得代价高昂

操作 2: 加和消掉一个变量

■ 第二个基本操作: 从因子表里 **加和去掉** 一个变量

■ 使一个因子变小

■ 例如: $\sum_j P(A, J) = P(A, j) + P(A, \neg j) = P(A)$

$P(A, J)$

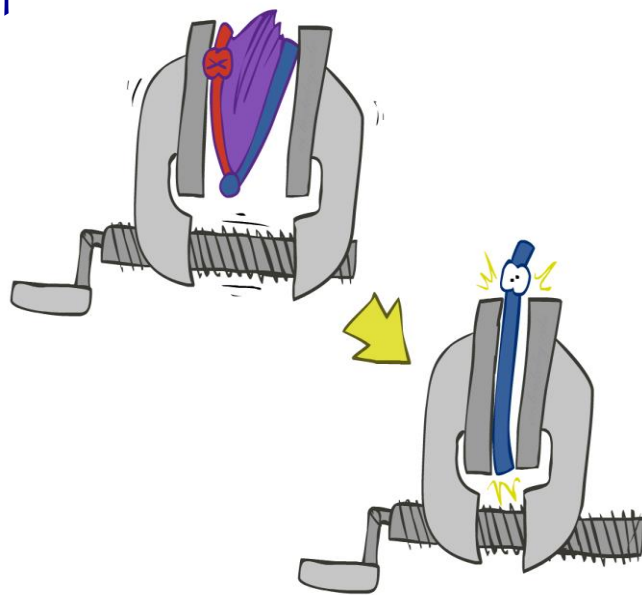
A \ J	true	false
true	0.09	0.01
false	0.045	0.855

加和消掉 J



$P(A)$

true	0.1
false	0.9

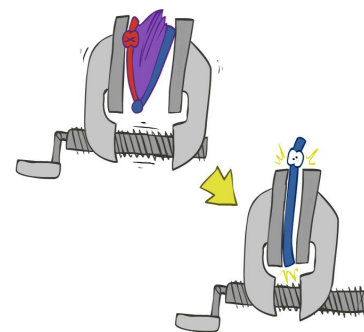


加和消除作用于因子乘积

■ 对每个因子表先根据实例化的变量值进行筛选（投射），然后再把乘积表达式取和

■ 例如：

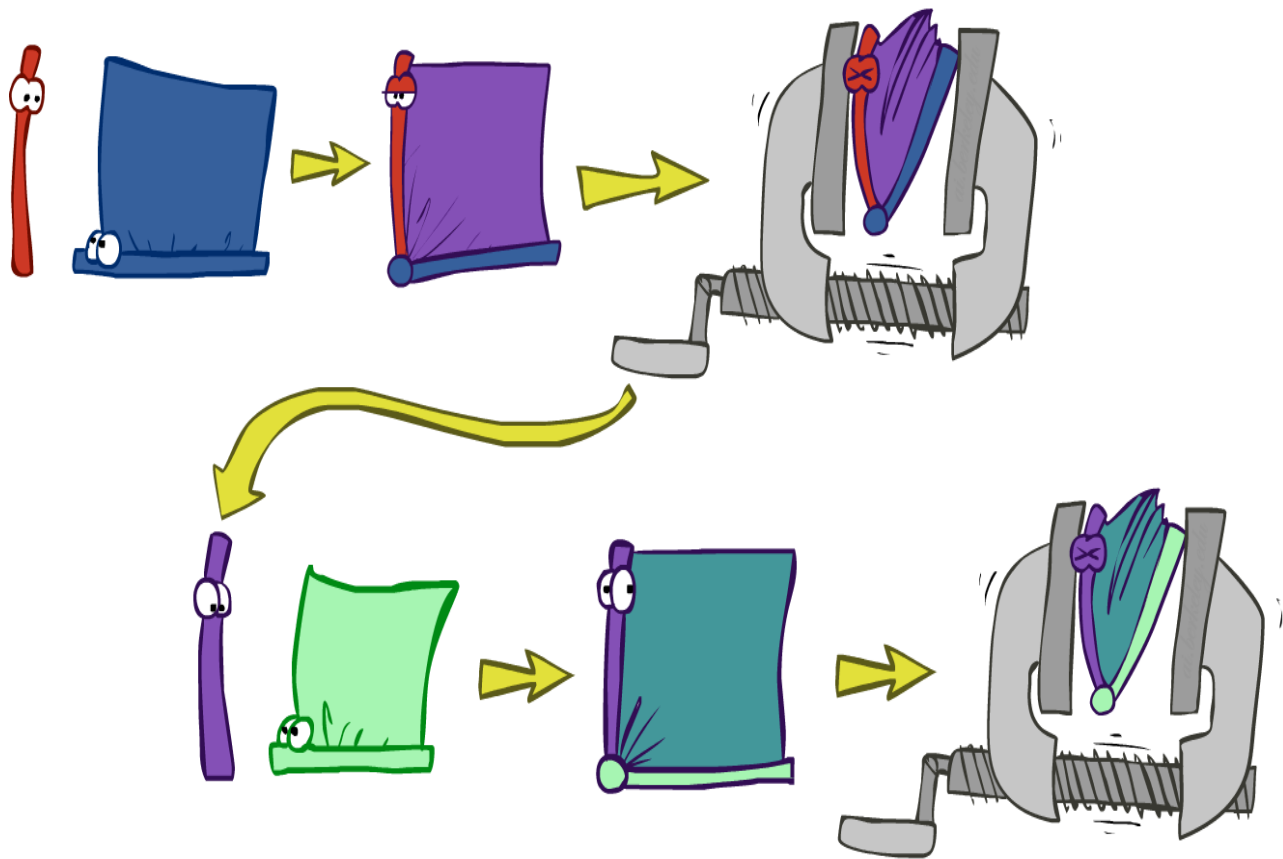
$$\begin{aligned} \circ \quad & \sum_a P(a|B,e) P(j|a) P(m|a) \\ &= P(a|B,e) P(j|a) P(m|a) + P(\neg a|B,e) P(j|\neg a) P(m|\neg a) \\ &= P(a,j,m|B,e) + P(\neg a,j,m|B,e) \\ &= P(j,m|B,e) \end{aligned}$$



举例：取和消除变量A

$$\begin{aligned}\mathbf{f}(B, C) &= \sum_a \mathbf{f}_3(A, B, C) = \mathbf{f}_3(a, B, C) + \mathbf{f}_3(\neg a, B, C) \\ &= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix} = \begin{pmatrix} .24 & .96 \\ .48 & .32 \end{pmatrix} .\end{aligned}$$

变量消除法(Variable Elimination)



变量消除法

■ 查询: $P(Q|E_1=e_1, \dots, E_k=e_k)$

■ 开始于初始的因子表:

■ 局部的条件概率表 (CPTs) (但经过观察变量E的实例化之后)

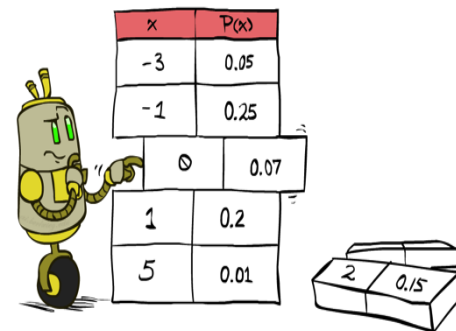
■ 当仍存在隐藏变量时 (既不是 Q 也不是E):

■ 选一个隐含变量 H

■ 合并所有包含 H 的因子表

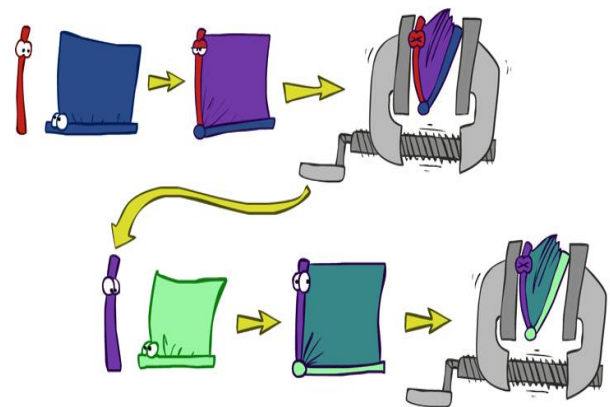
■ 消除变量 (通过取和) H

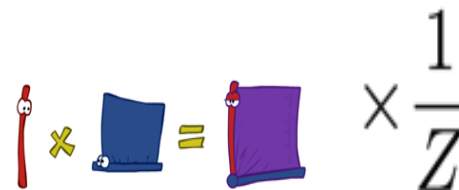
■ 合并所有剩余因子表, 并对结果进行正规化



X	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

2 0.15




$$\text{stick} \times \text{blue square} = \text{purple square} \times \frac{1}{Z}$$

变量消除法

function **VariableElimination**(Q, e, bn) **returns** 一个 Q 上的分布

$factors \leftarrow []$

for each var **in** **ORDER**($bn.vars$) **do**

$factors \leftarrow [MAKE-FACTOR(var, e) | factors]$

if var 是一个隐含变量 **then**

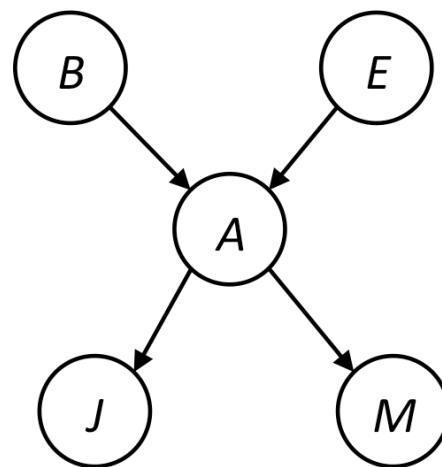
$factors \leftarrow SUM-OUT(var, factors)$

return **NORMALIZE**(**POINTWISE-PRODUCT**($factors$))

举例：之前的防盗报警器网络

■ 要查询 $P(B \mid j, m)$

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{f_1(B)} \sum_e \underbrace{P(e)}_{f_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{f_3(A, B, E)} \underbrace{P(j \mid a)}_{f_4(A)} \underbrace{P(m \mid a)}_{f_5(A)}$$



举例：之前的防盗报警器网络

查询 $P(B \mid j, m)$

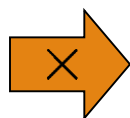
$P(B)$	$P(E)$	$P(A \mid B, E)$	$P(j \mid A)$	$P(m \mid A)$
--------	--------	------------------	---------------	---------------

选择 A

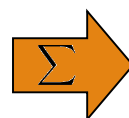
$P(A \mid B, E)$

$P(j \mid A)$

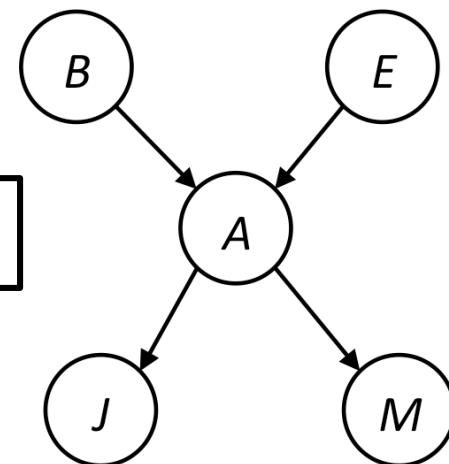
$P(m \mid A)$



$P(A, j, m \mid B, E)$



$P(j, m \mid B, E)$



$P(B)$	$P(E)$	$P(j, m \mid B, E)$
--------	--------	---------------------

举例

$P(B)$	$P(E)$	$P(j,m B,E)$
--------	--------	--------------

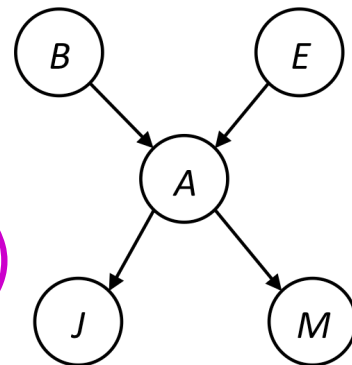
选择 E

$$\begin{array}{l} P(E) \\ P(j,m|B,E) \end{array} \xrightarrow{\times} P(E,j,m|B) \xrightarrow{\Sigma} P(j,m|B)$$

$P(B)$	$P(j,m B)$
--------	------------

最后查询 B

$$\begin{array}{l} P(B) \\ P(j,m|B) \end{array} \xrightarrow{\times} P(j,m,B) \xrightarrow{\text{正规化}} P(B|j,m)$$



选择变量的顺序有关系

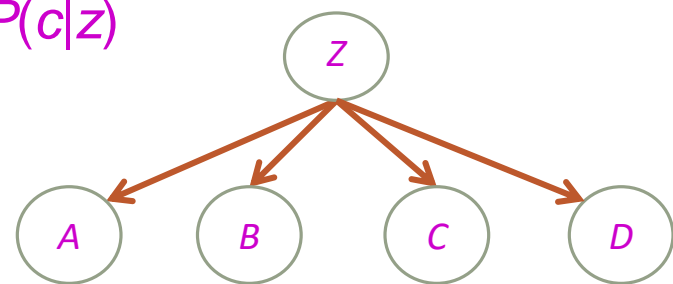
- 如果排序为 **D, Z, A, B C**

- $P(D) = \alpha \sum_{z,a,b,c} P(D|z) P(z) P(a|z) P(b|z) P(c|z)$
- $= \alpha \sum_z P(D|z) P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z)$
- 最大的因子有 2 个变量 (D,Z)

- 如果排序为 **A, B C, D, Z**

- $P(D) = \alpha \sum_{a,b,c,z} P(a|z) P(b|z) P(c|z) P(D|z) P(z)$
- $= \alpha \sum_a \sum_b \sum_c \sum_z P(a|z) P(b|z) P(c|z) P(D|z) P(z)$
- 最大的因子有 4 个变量 (A,B,C,D)

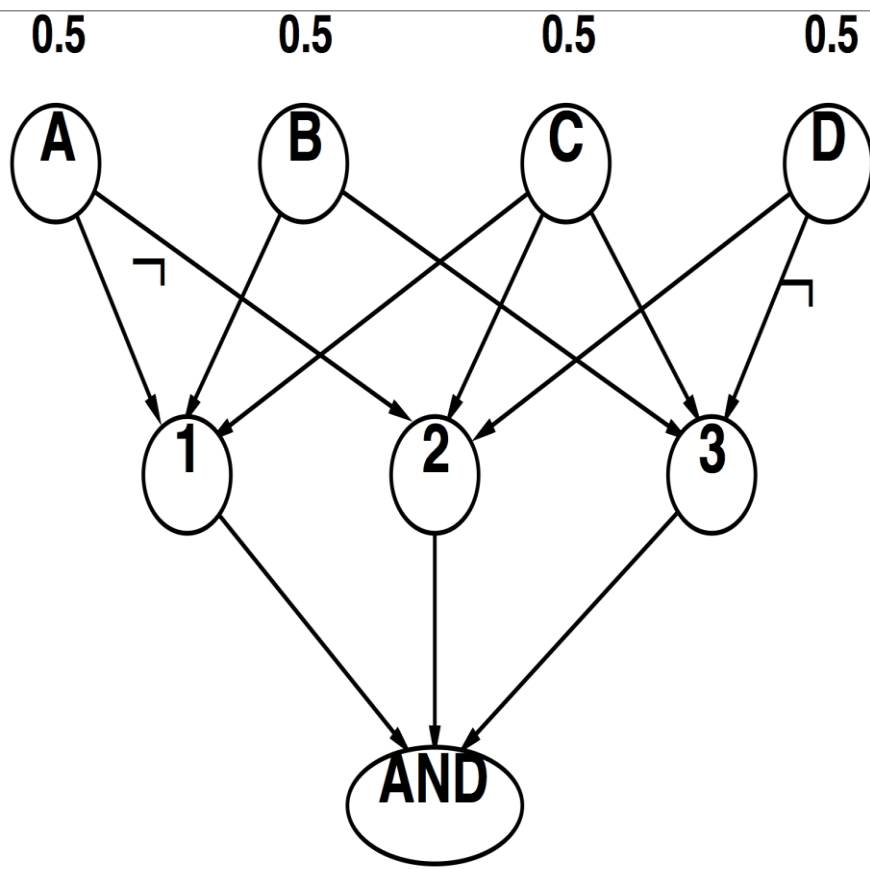
- 通常, 如果有 n 个叶节点, 因子表的大小是 2^n



变量消除法: 计算时间和空间复杂度

- 计算时间和空间复杂度是由最大因子表的大小来决定的 (存储空间要求有可能过大而难以存储)
- 变量去除的顺序可以很大程度上影响最大因子表的大小
 - 例如, 上一页举例中, 2^4 vs. 2^2
 - 其他原因影响因子表大小的是网络结构
- 是否存在一个最佳排序方法总是能够只导致小因子表 (变量数少)?
 - 不存在!

最差情况复杂度？从 SAT 问题约简过来



■ 合取范式(CNF)的子句:

■ $A \vee B \vee C$

■ $C \vee D \vee \neg A$

■ $B \vee C \vee \neg D$

■ $P(\text{AND}) > 0$ 当且仅当 所有子句是可满足的

■ \Rightarrow NP-难度

■ $P(\text{AND}) = S \times 0.5^n$, S 是使该合取范式满足的变量配值的组数

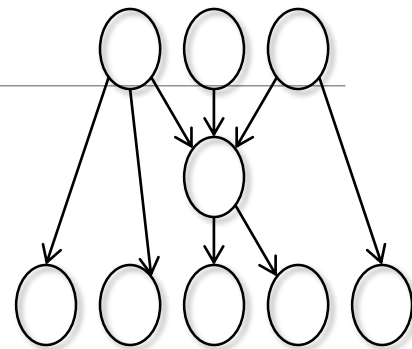
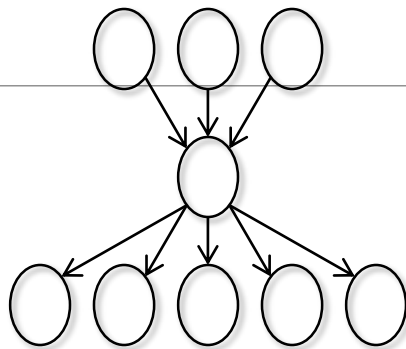
■ \Rightarrow #P-难度 (至少和其对应的NP-难度问题一样难, 或更难的)

最差情况复杂度？从 SAT 问题约简过来

- 如果我们能够回答 $P(\text{AND})=0$ 或大于0的话，那么我们就已经回答了这个问题是否存在一个解；
- 因此，贝叶斯网络里的推理难度是NP-hard，即没有已知的高效的概率推理方法，适用于所有情况。

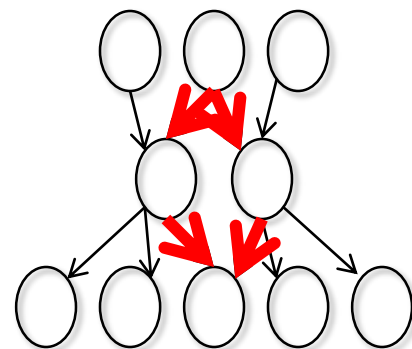
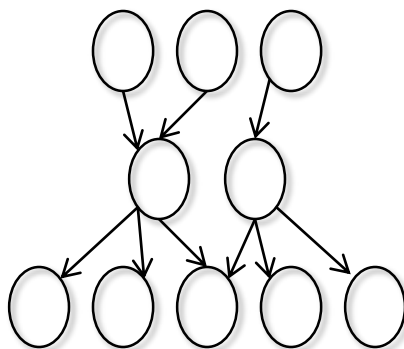
多树 (Polytrees)

■ 一个多树是一个有向无环图
对应的无向图是一个树（即无环）



■ 对于多树，当变量消除的顺序是从叶到根的话，变量消除法的复杂度是和网络的大小成线性关系的，

■ 本质上是与树结构的约束满足问题 (CSPs) 的求解是同一个原理



贝叶斯网络 (Bayes Nets)

✓ Part I: 表达

✓ Part II: 精确推理

- ✓ ◦ 列举法 (总是导致指数级复杂度)
- ✓ ◦ 变量消除法 (最差情况下指数级复杂度, 通常情况会更好)
- ✓ ◦ 通常情况下, 推理是 **NP-难度** (没有通用的最优解法)

Part III: 近似推理