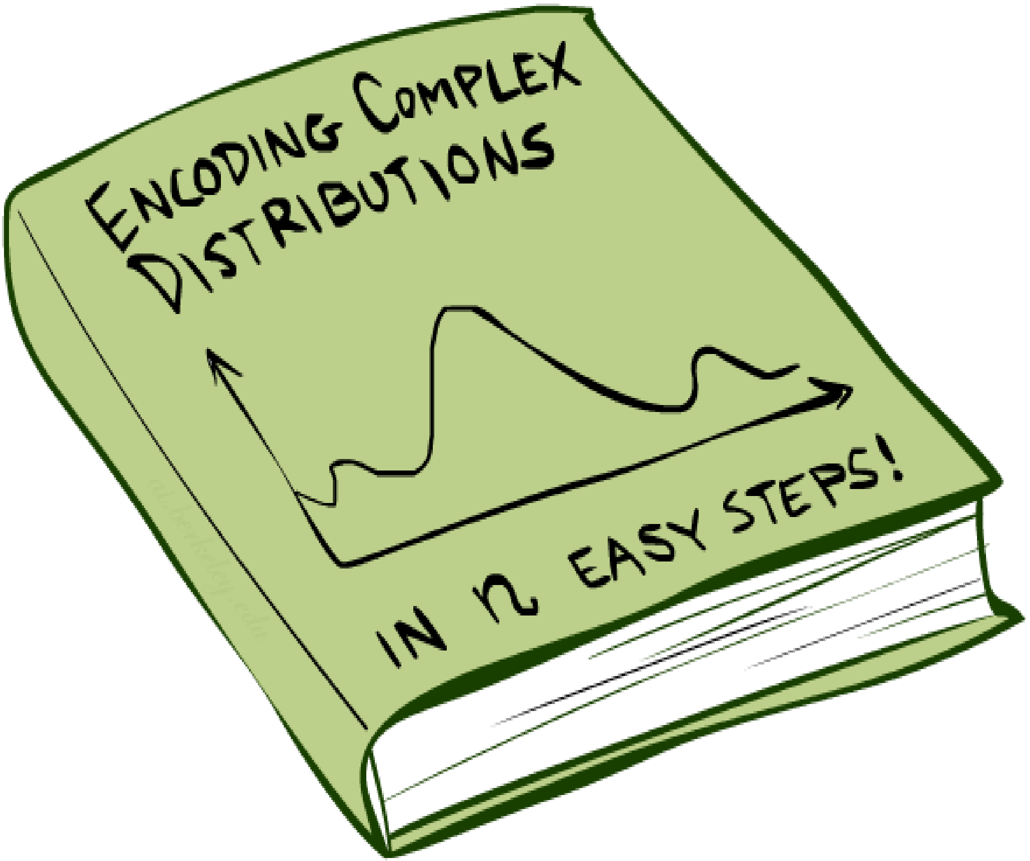
通知

- 作业3，已发布
  - 概率计算，贝叶斯网络属性
  - 12月8日上课时提交

今天的内容

- 贝叶斯网络：宏观介绍
- 贝叶斯网络：精确推理

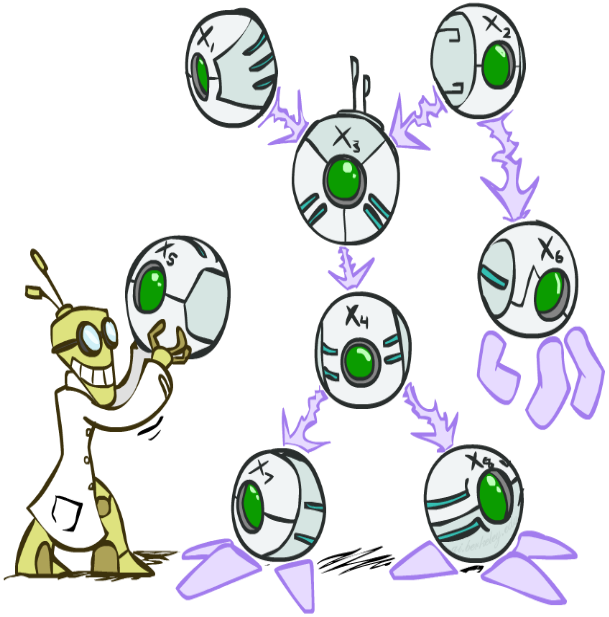
贝叶斯网络(BayesNets): 宏观介绍



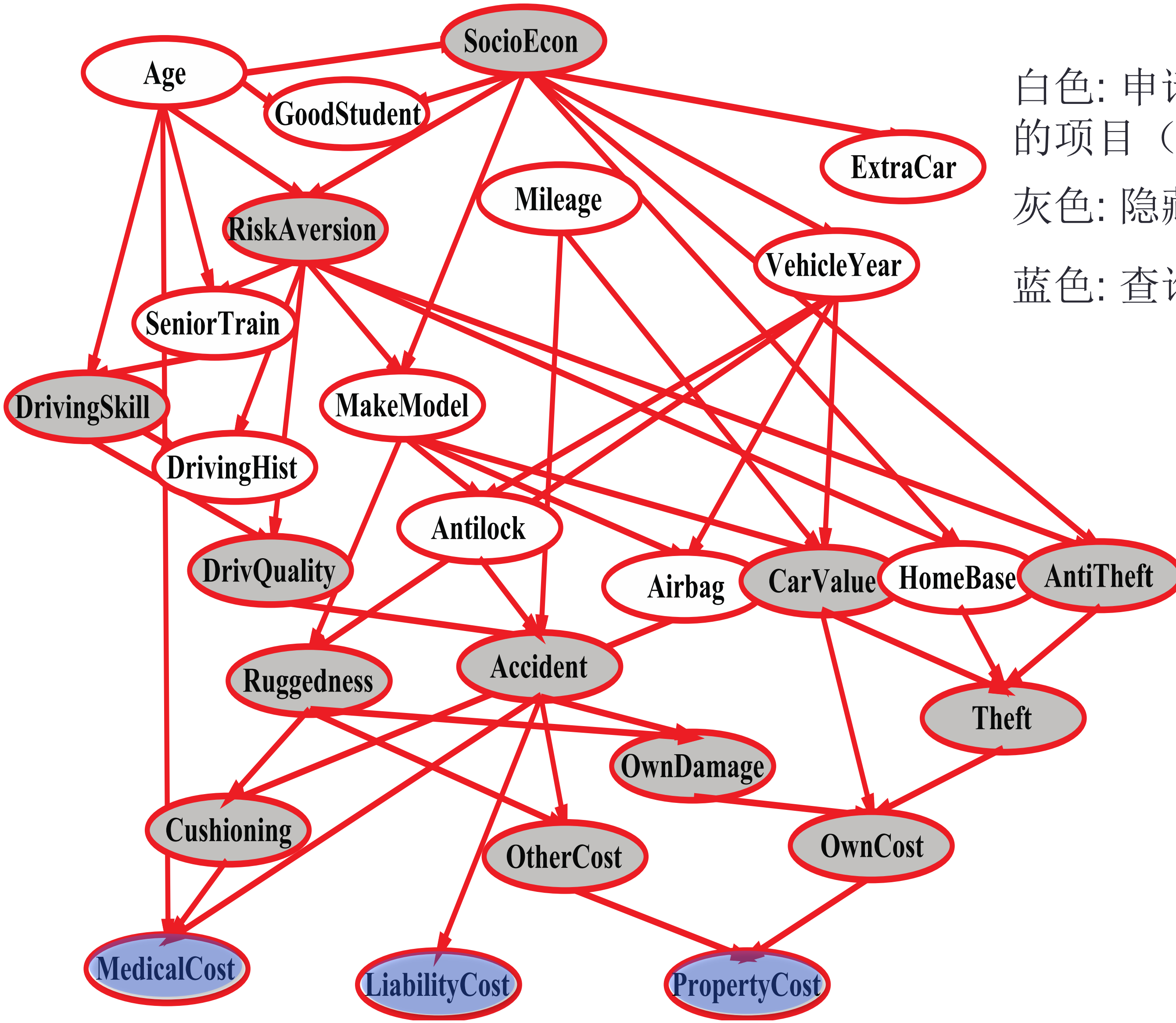
贝叶斯网络: 宏观介绍

- 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

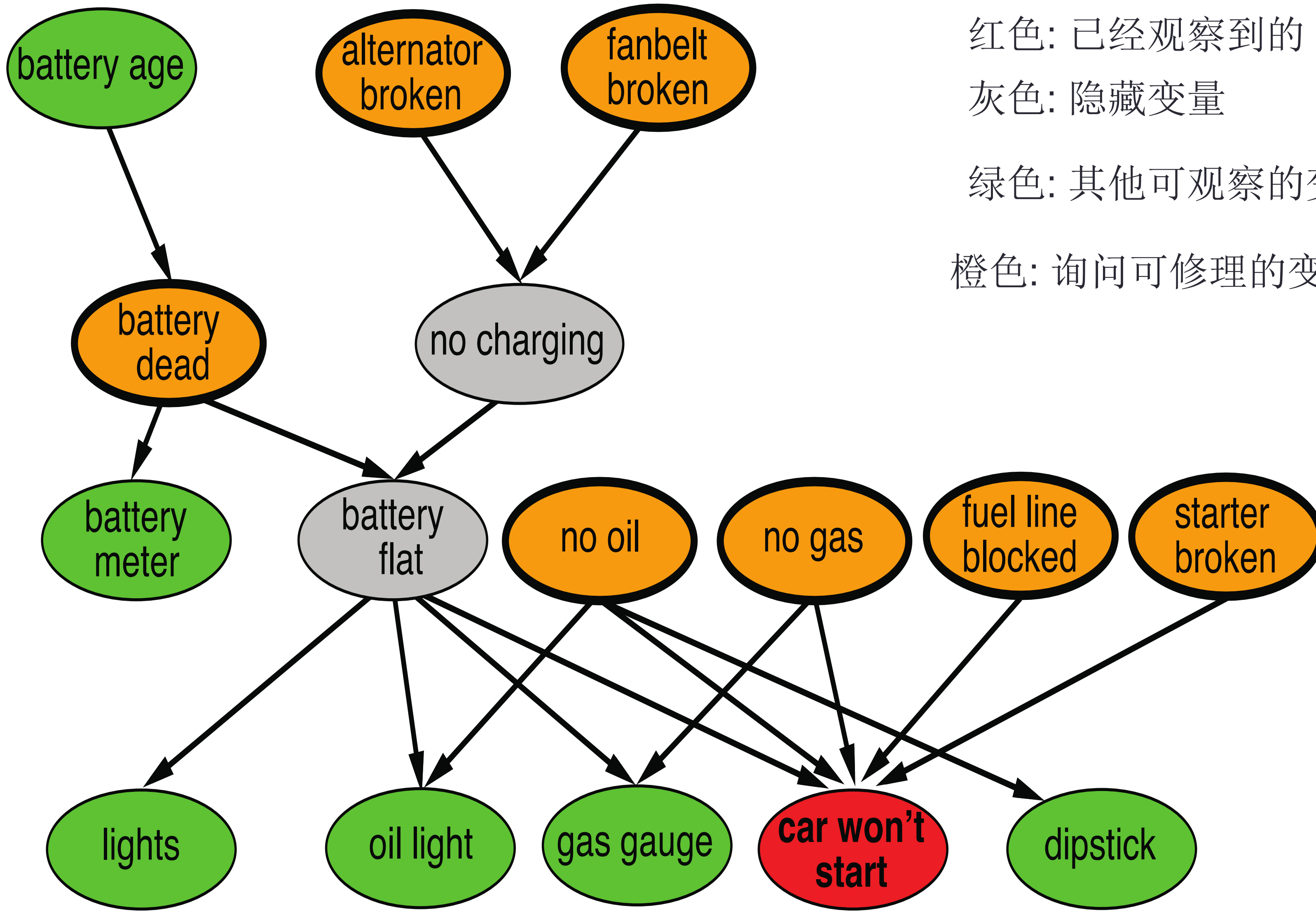


白色: 申请表格里出现的项目（观察到的）

灰色: 隐藏变量

蓝色: 查询变量

贝叶斯网络举例: 简单汽车维修



红色: 已经观察到的

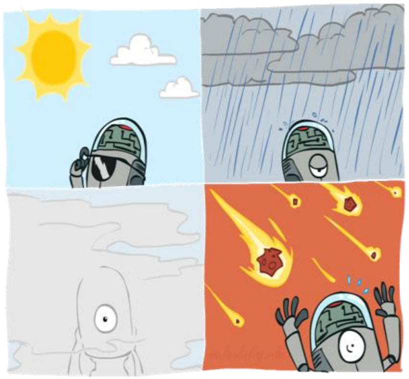
灰色: 隐藏变量

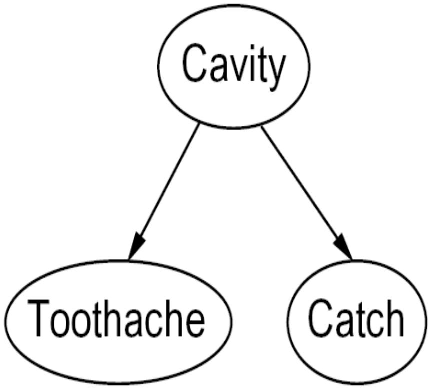
绿色: 其他可观察的变量

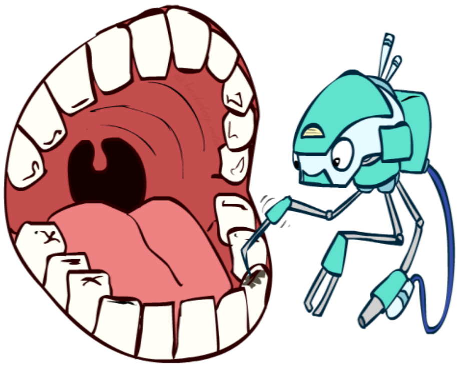
橙色: 询问可修理的变量

图形模型的表达

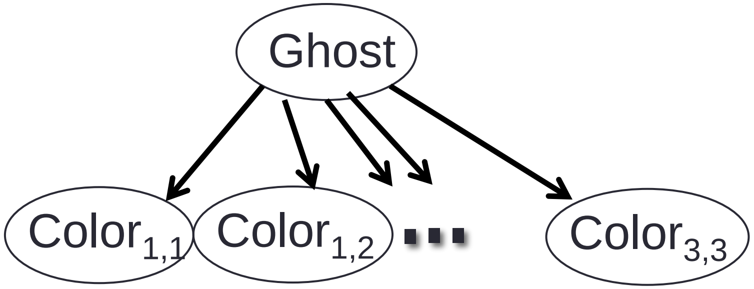
- 节点: 变量 (每个变量有个值域)
- 弧/边: 相互作用
  - 指示 变量间的“直接的影响”
  - 正式的表达: 编码条件独立性
- 现在可以简单地: 箭头意味着直接的因果关系 (通常情况下, 它们并不一定是这样!)







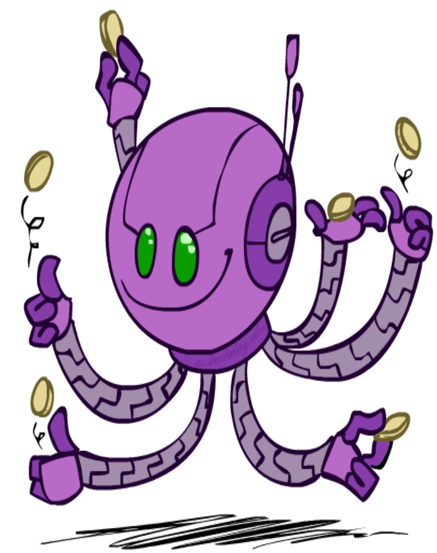
Catch，这里指的是牙医的探测器是否捕获到你的牙是否出问题；有可能会有遗漏。



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 $\cdots$ X_n



- 变量间没有相互作用: 绝对独立

举例: 交通流量预测

- 变量:
  - R: 下雨
  - T: 交通状况

- 模型 1: 独立的

R

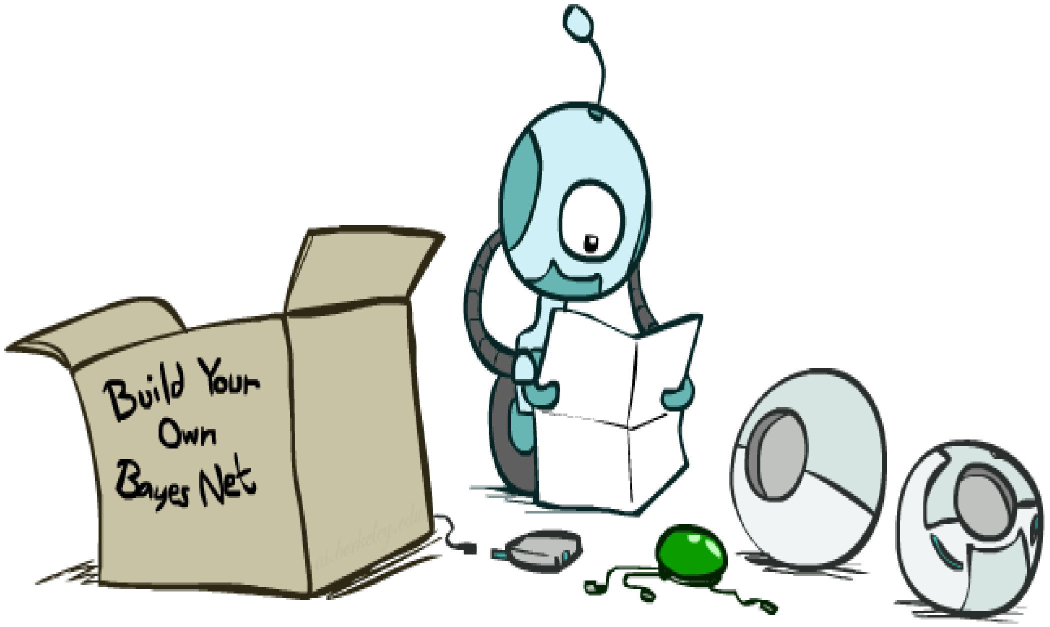
T

- 模型 2: 下雨导致交通状况变化

$R \rightarrow T$

- 为什么用模型 2 更好?

贝叶斯网络语法和语义

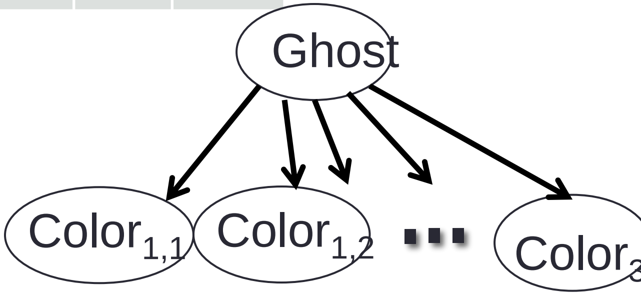


贝叶斯网络语法

P(Ghost)			
(1,1)	(1,2)	(1,3)	…
0.11	0.11	0.11	…

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

Ghost	$P(Color_{1,1} \mid Ghost)$			
	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
…				

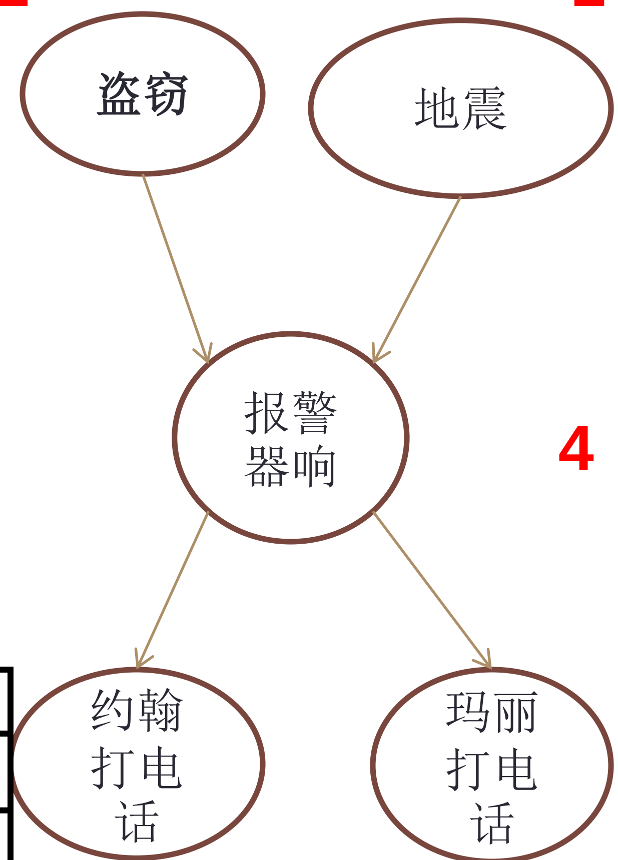


贝叶斯网络 = 拓扑结构(图形) + 局部条件概率

举例: 报警器网络

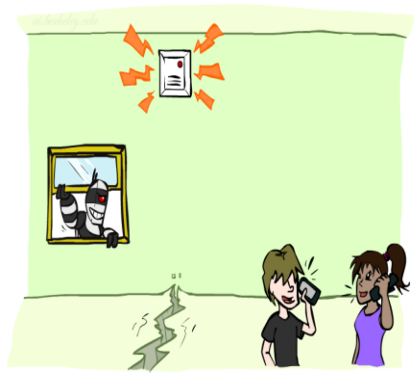
P(B)	
true	false
0.001	0.999

1



P(E)	
true	false
0.002	0.998

1



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

4

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: $d_1, \ldots, d_k$

子变量的值域为 d

表中每一行概率值之和为 1

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

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： $O(n \cdot d^{k+1})$
  - n 的线性比例，只要因果结构是局部的

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：

$$P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$$

举例

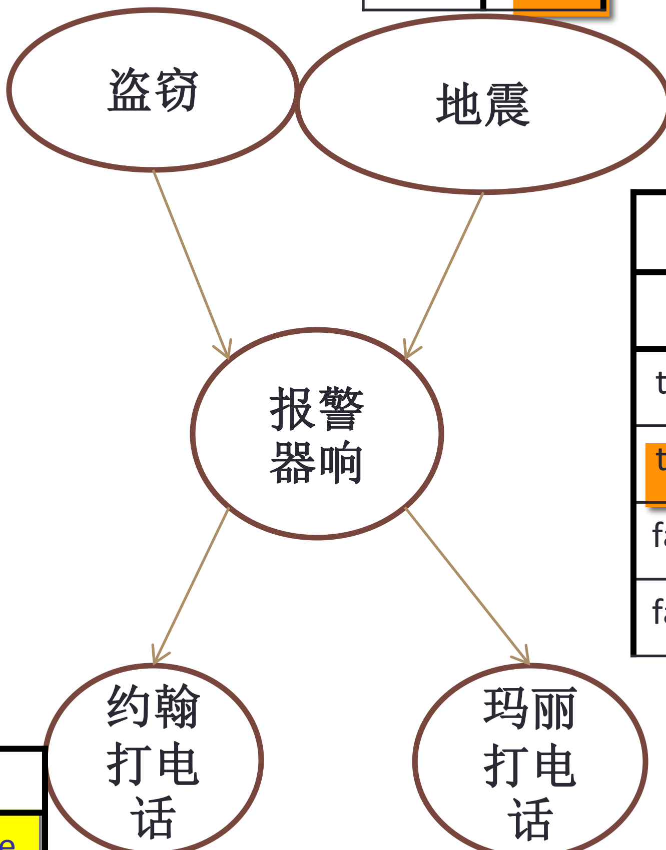
P(B)	
true	false
0.001	0.999

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

贝叶斯网络里的概率

- 为什么我们可以保证以下公式反映的是正确的联合分布

$$P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$$

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

- <u>假定</u> 条件独立性: $P(X_i \mid X_1,\ldots,X_{i-1}) = P(X_i \mid Parents(X_i))$
  - 当加入节点 X_i, 保证了其父节点 “屏蔽” 它与其他祖先节点的联系

→ 结果: $P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$

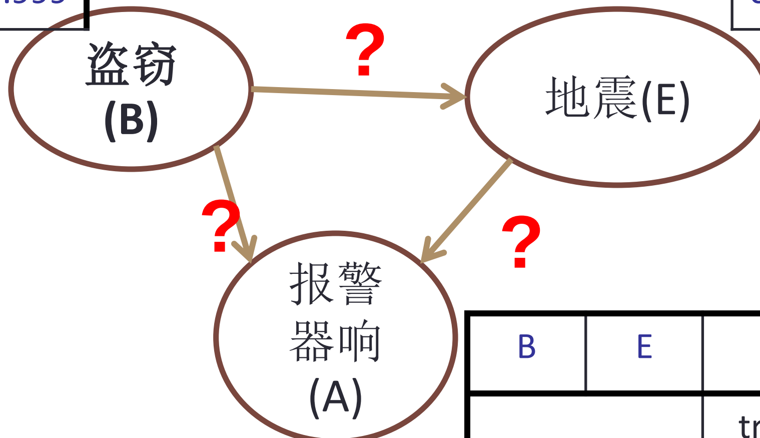
- 所以，网络的拓扑结构暗示着肯定的条件独立性的成立

举例: 入室偷盗报警

- 入室盗窃
- 地震
- 报警器

P(B)	
true	false
0.001	0.999

P(E)	
true	false
0.002	0.998

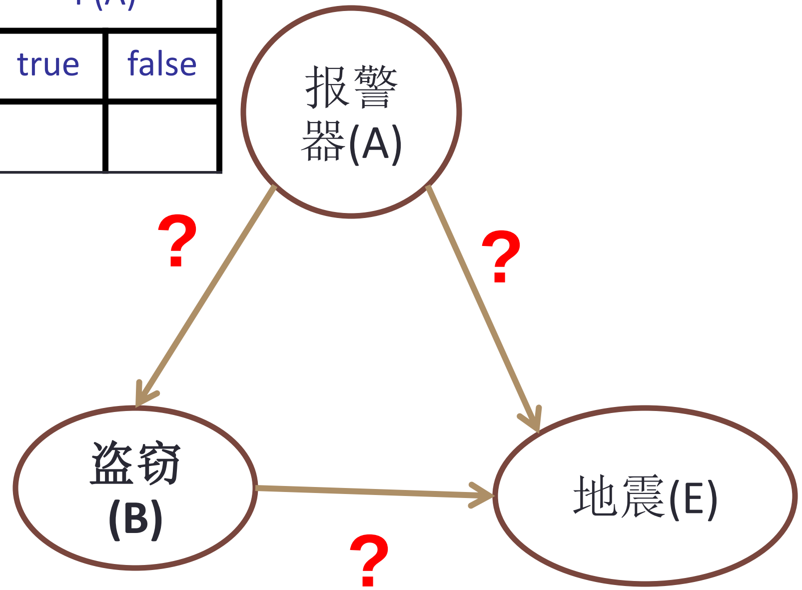


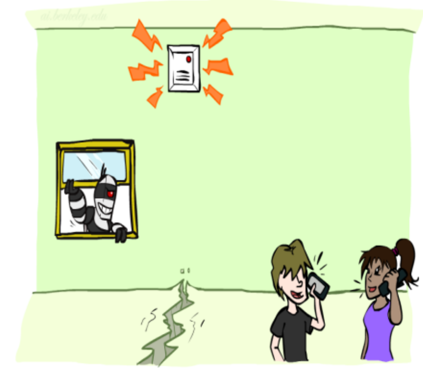
B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

举例: 入室偷盗报警

- 报警器
- 入室盗窃
- 地震

P(A)	
true	false



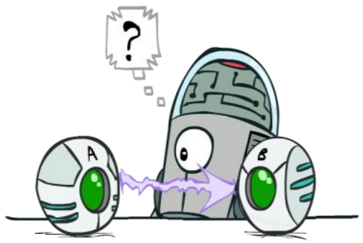


A	P(B|A)	
	true	false
true	?	
false		

A	B	P(E|A,B)	
		true	false
true	true	?	
true	false		
false	true		
false	false		

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(B)的频率后应该不影响模型里的其他部分!

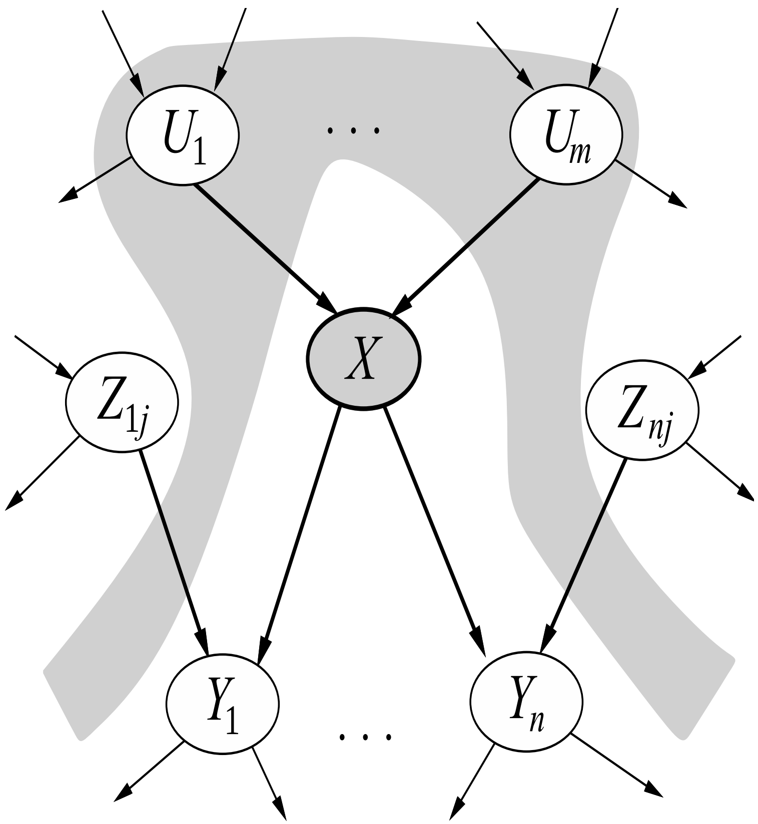


- BNs 不需要实际上表达因果关系
  - 有时没有因果网络存在于一个领域 (尤其是在一些变量丢失的情况下)
  - 例如，考虑变量 *交通状况* 和 *屋檐滴水*
  - 其结果是箭头关联反映的是相关性(correlation), 而不是因果关系
- 箭头实际表示的是什么?
  - 可能碰巧表达的是因果关系
  - 拓扑结构真正表达（编码）的是条件独立性:

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

条件独立性的语义

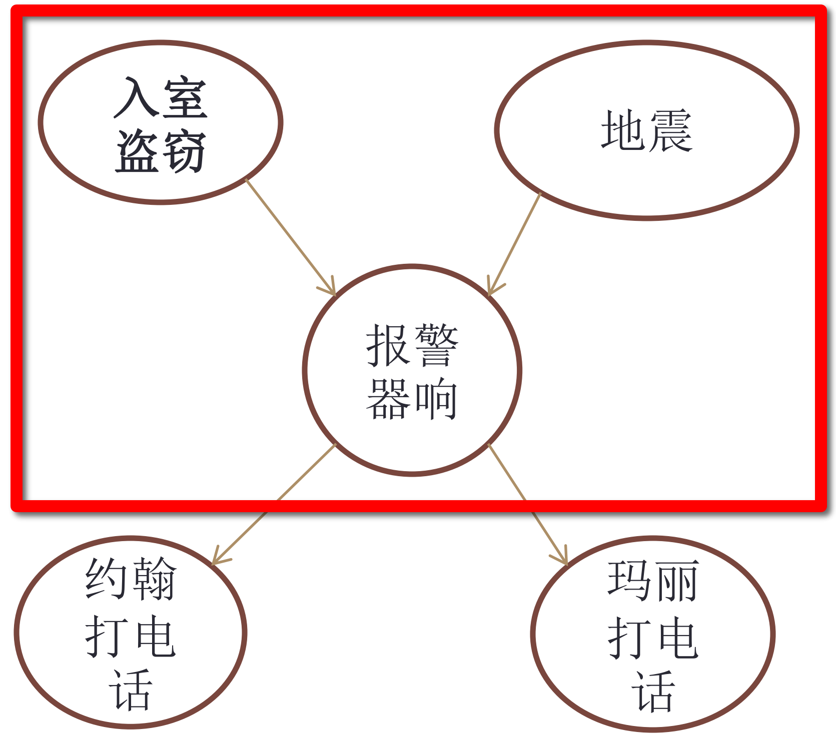
- *每个变量在给定它的父变量节点情况下，则是条件独立于它的非 后代变量*



举例

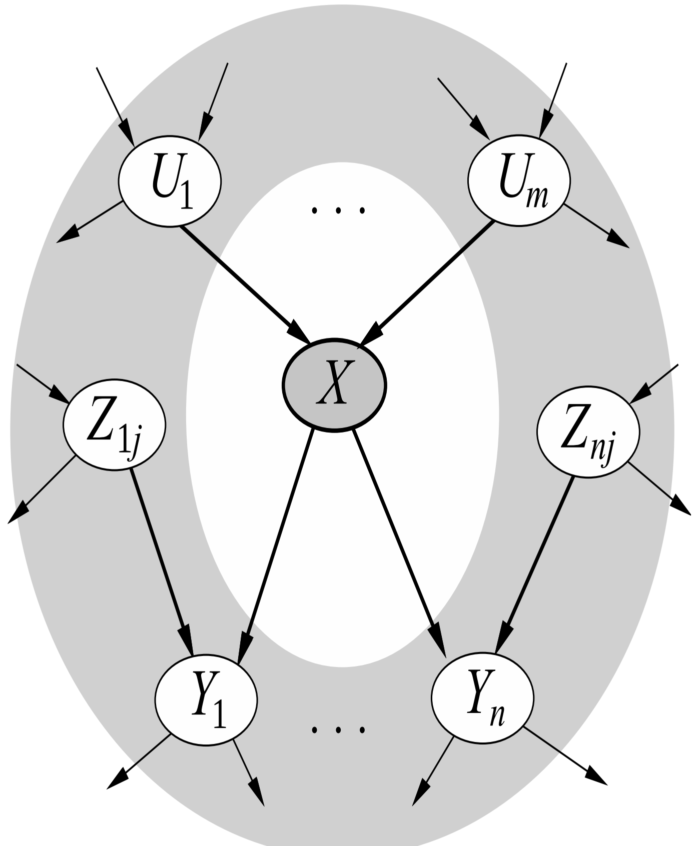
- 给定报警器响，约翰打电话 是否 独立于 入室盗窃的发生？
  - 是的
- 给定报警器响，约翰打电话 是否 独立于 玛丽打电话?
  - 是的
- 盗窃 是否独立于 地震?
  - 是的
- 盗窃 是否独立于 地震 当报警器响后？
  - 不是!
  - 报警器已响，入室盗窃和地震都变得很有可能发生过
  - 但是，如果我们得知一个入室盗窃已经发生, 那么报警器响的原因被 ***解释***， 则地震发生的概率降低

V-结构



马尔科夫毯(Markov blanket)

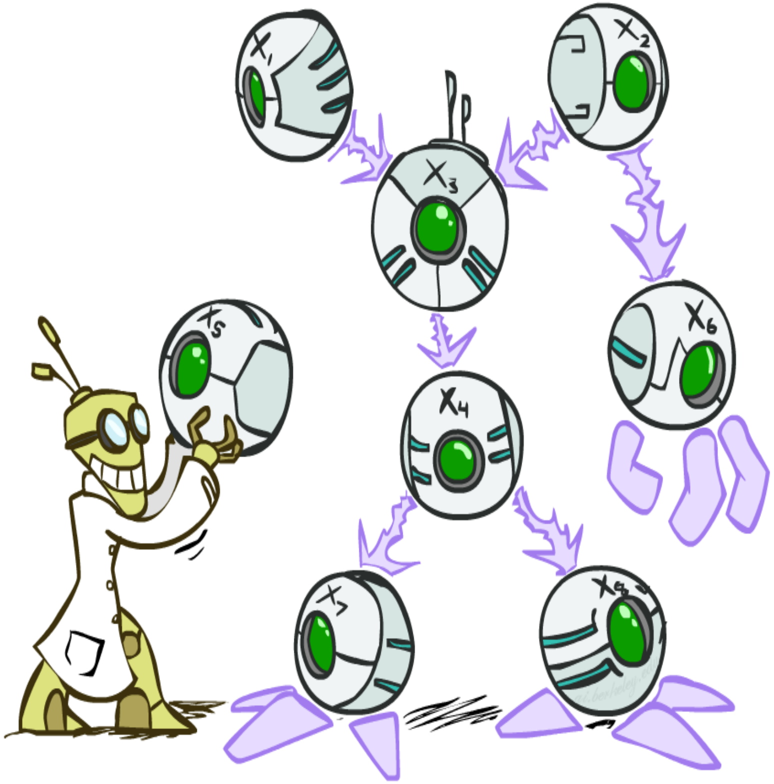
- 一个变量的马尔可夫毯包括父节点，子节点, 子节点的其他父节点
- ***每个变量给定它的马尔科夫毯，则是条件独立于所有其他变量***



贝叶斯网络(Bayes Nets)

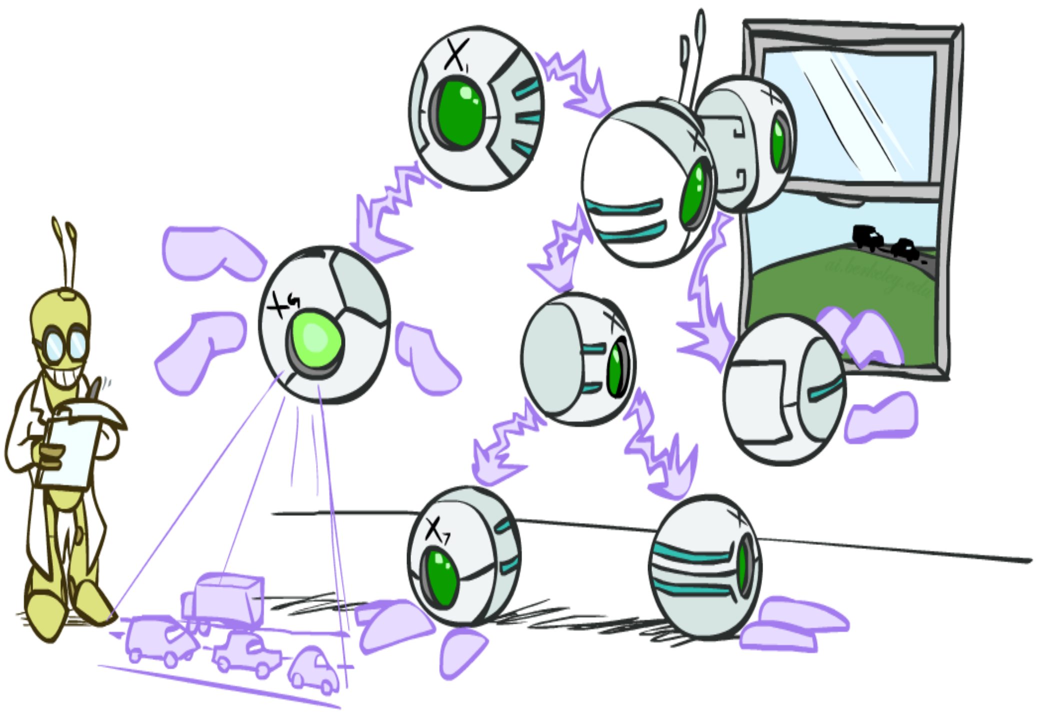
- 已经介绍：贝叶斯网络如何实现了对联合分布的表达

- 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



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

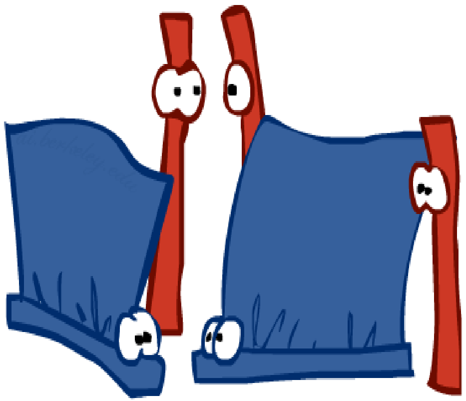
Part III: 近似推理(Approximate Inference)

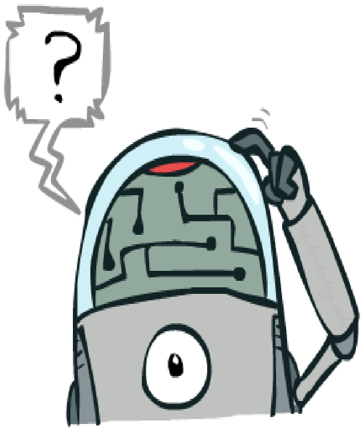
后面: 从数据学习构建网络结构

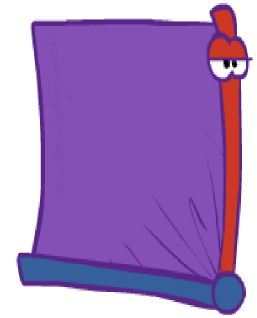
推理

- 计算某些有用的数量，从一个概率模型里（联合概率分布）

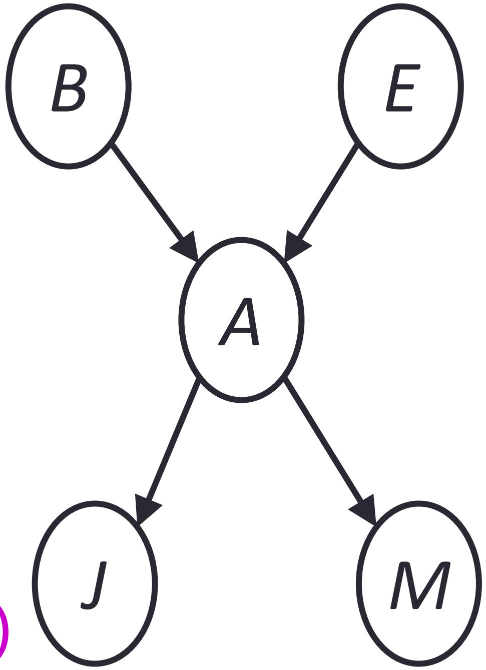
- 例如:
  - 后验边缘概率(Posterior marginal probability)
    - $P(Q|e_1,..,e_k)$
    - 举例：给定一些症状，推理可能的疾病原因
  - 推理最有可能的解释是什么:
    - $\mathrm{argmax}_{q,r,s}\ P(Q=q,R=r,S=s|e_1,..,e_k)$







通过列举法(Enumeration)在贝叶斯网络(BN)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
- $P(B \mid j, m) = \alpha P(B, j, m)$
- $= \alpha \sum_{e,a} P(B, e, a, j, m)$
- $= \alpha \sum_{e,a} P(B) P(e) P(a|B,e) P(j|a) P(m|a)$
- 所以BN推理意味着对数量乘积进行求和计算: 似乎很容易!!
- 问题: 要计算 指数增长的乘积项之和!

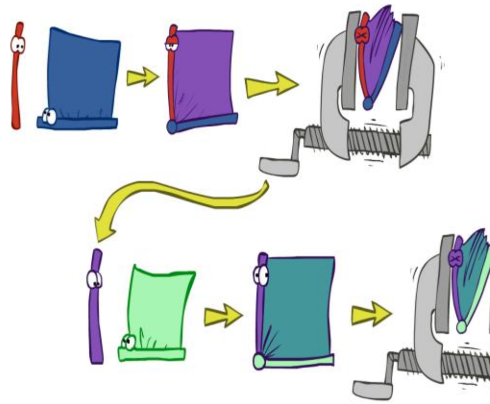
是否能做的更好?

- 思考： **uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz**
  - 16 乘法, 7 个加法
  - 许多重复的子表达式!
- 重写为： **(u+v)(w+x)(y+z)**
  - 2 乘法, 3 个加法
- $\sum_{e,a} P(B)\ P(e)\ P(a|B,e)\ P(j|a)\ P(m|a)$
- $= P(B)P(e)P(a|B,e)P(j|a)P(m|a) + P(B)P(\neg e)P(a|B,\neg e)P(j|a)P(m|a)$
  $+ P(B)P(e)P(\neg a|B,e)P(j|\neg a)P(m|\neg a) + P(B)P(\neg e)P(\neg a|B,\neg e)P(j|\neg a)P(m|\neg a)$

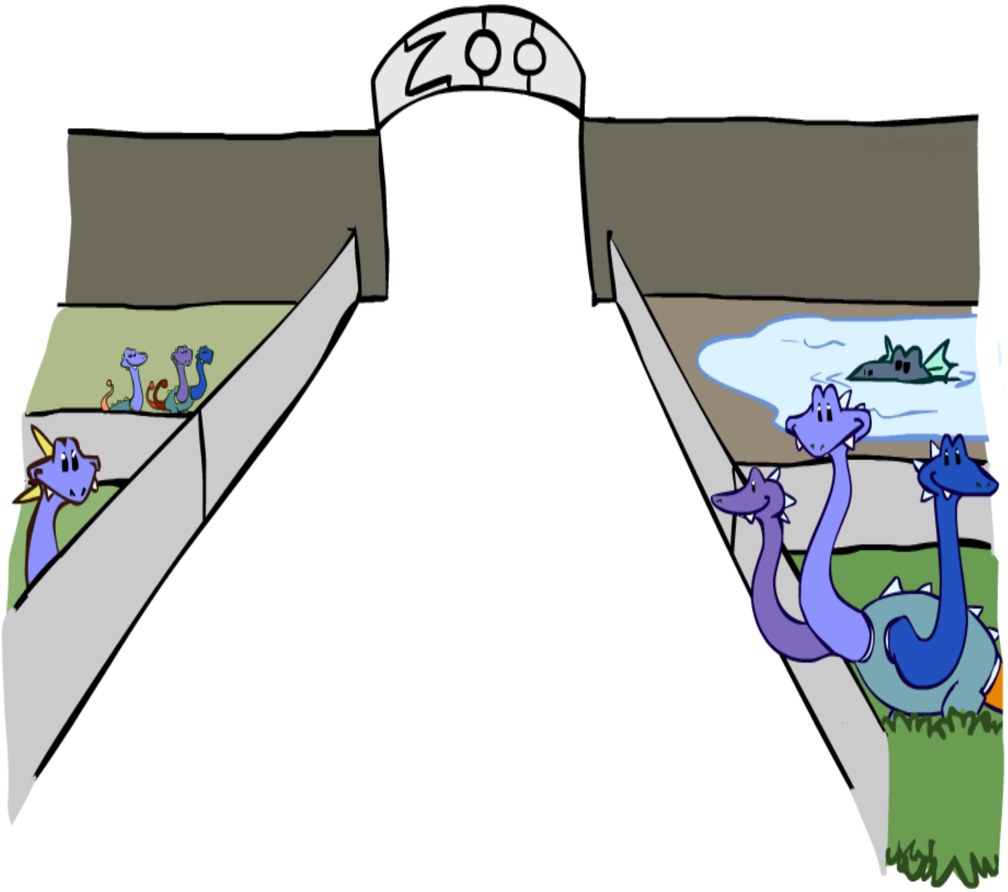
重复的子式!

变量消除法: 基本概念

- 尽量把求和操作移到里面，先消掉一些变量
  - $P(B \mid j, m) = \alpha \sum_{e,a} P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$
  - $= \alpha\, P(B) \sum_{e} P(e) \sum_{a} P(a|B,e)\, P(j|a)\, P(m|a)$
- 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|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：使用数组，以及相应的操作；这些列表也叫作 ***因子(factors)***；其每行数字之和不必须为1



因子表达形式(Factor Zoo)



因子动物园 I

- 联合分布: P(X,Y)
  - 表项 P(x,y) 对任一x, y
  - |X| x |Y| 矩阵
  - 表项之和 为 1

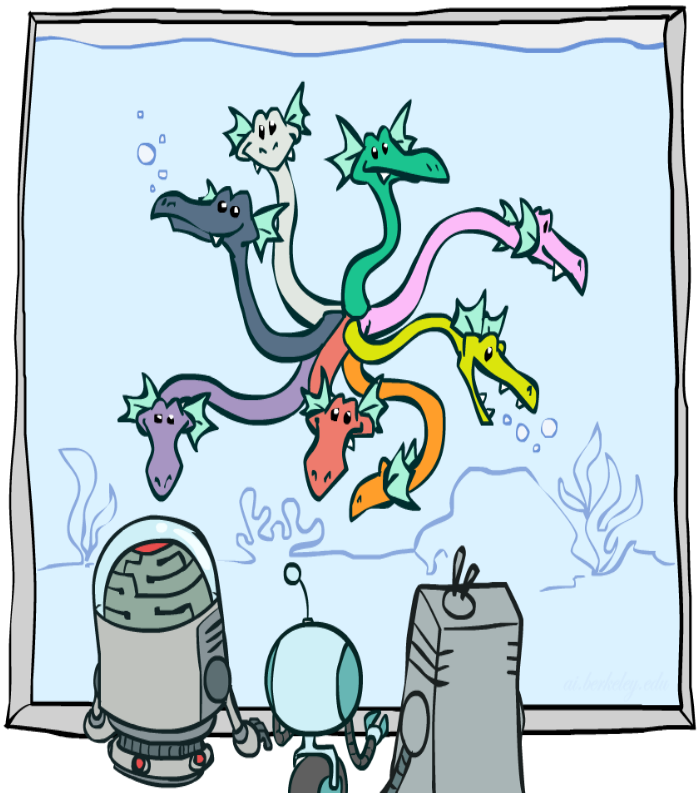
P(A,J)

A \ 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, 所有的 y
  - |Y|-个元素的向量数组
  - 之和为 P(x)

P(a,J)

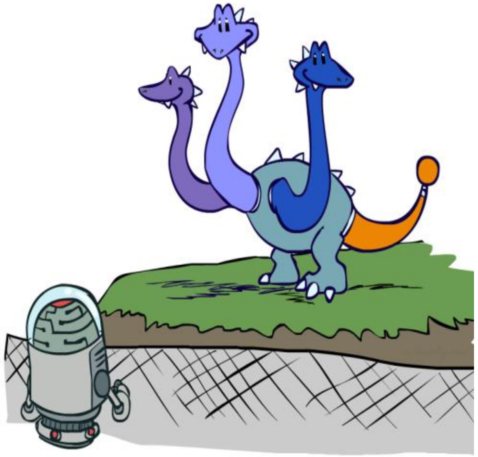
A \ J	true	false
true	0.09	0.01



变量数之和 (大写的变量) = 表的维度

因子动物园II

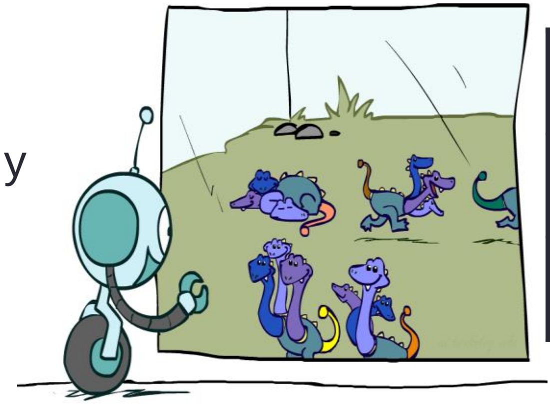
- 单独条件概率: P(Y | x)
  - 表项 P(y | x)，对于固定的 x, 所有的 y
  - 表项之和为 1



P(J|a)

A \ J	true	false
true	0.9	0.1

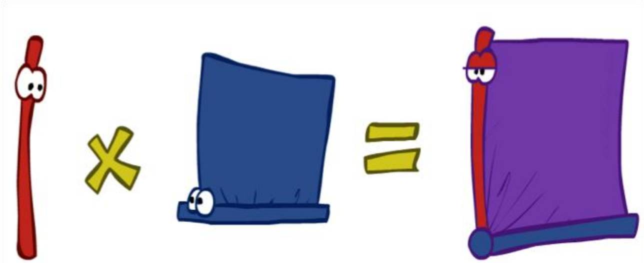
- 条件概率的分布: P(X |Y)
  - 多个条件概率
  - 表项 P(x | y)，对于所有 x, y
  - 表项之和为 |Y|



P(J|A)

A \ J	true	false	
true	0.9	0.1	} - *P(J|a)*
false	0.05	0.95	} - *P(J|¬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!)
  - 结果的新因子里包含两个原始因子里变量的合集
  - 每个表项是原始因子相应项的乘积
- 例如: $P(J|A)$ x $P(A)$ = $P(A,J)$

$P(A)$

true	0.1
false	0.9

x

$P(J|A)$

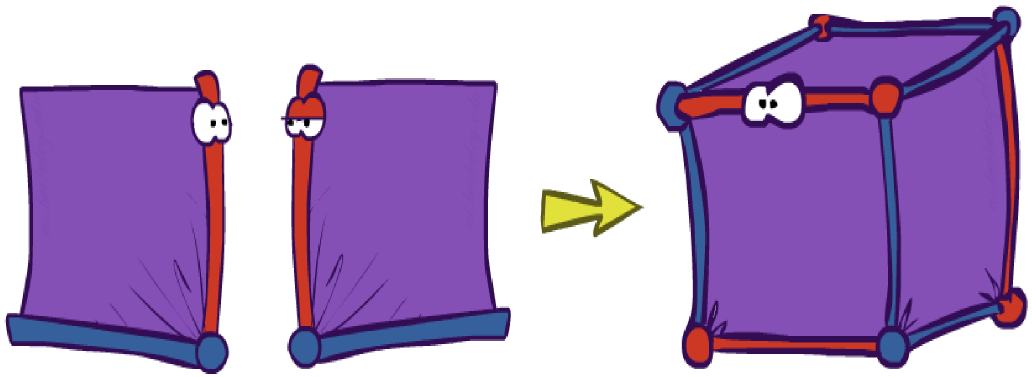
A \ J	true	false
true	0.9	0.1
false	0.05	0.95

=

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

举例: 生成更大的因子



- 举例: $P(A,J)$ x $P(A,M)$ = $P(A,J,M)$

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

x

$P(A,M)$

A \ M	true	false
true	0.07	0.03
false	0.009	0.891

=

$P(A,J,M)$

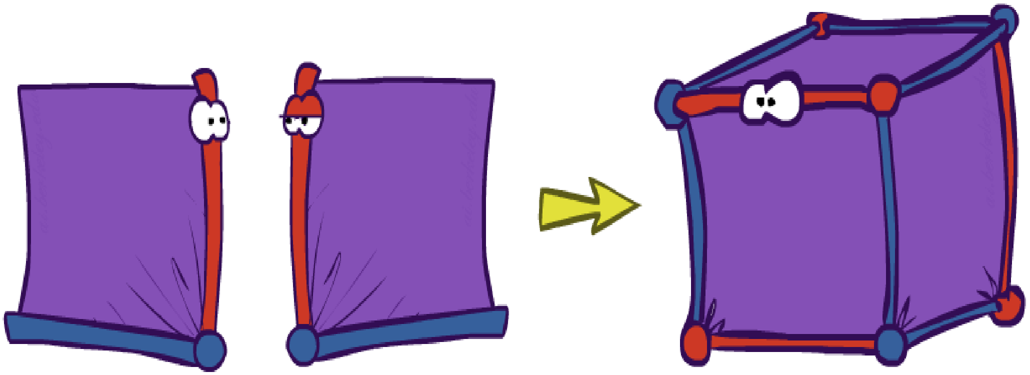
A=false

J \ M	true	false
		18

A=true

J \ M	true	false
true		
false		.0003

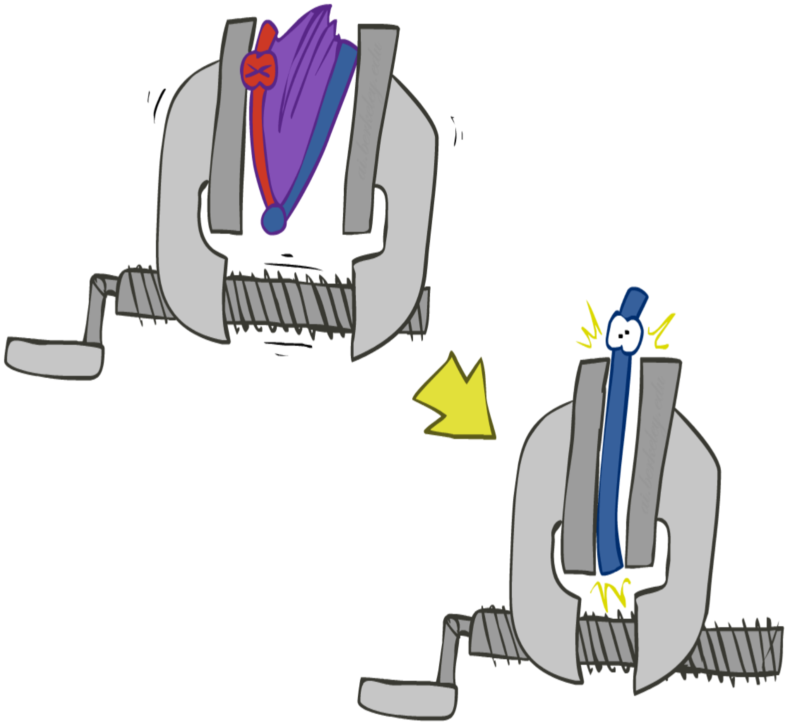
举例：生成更大的因子



- 例如: *P*(*U*,*V*) x *P*(*V*,*W*) x *P*(*W*,*X*) = *P*(*U*,*V*,*W*,*X*)
- 因子表尺寸: [10,10] x [10,10] x [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

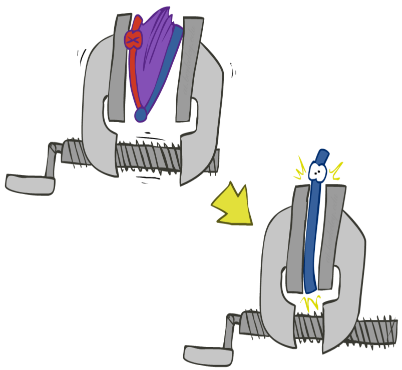
加和消掉 J

$P(A)$

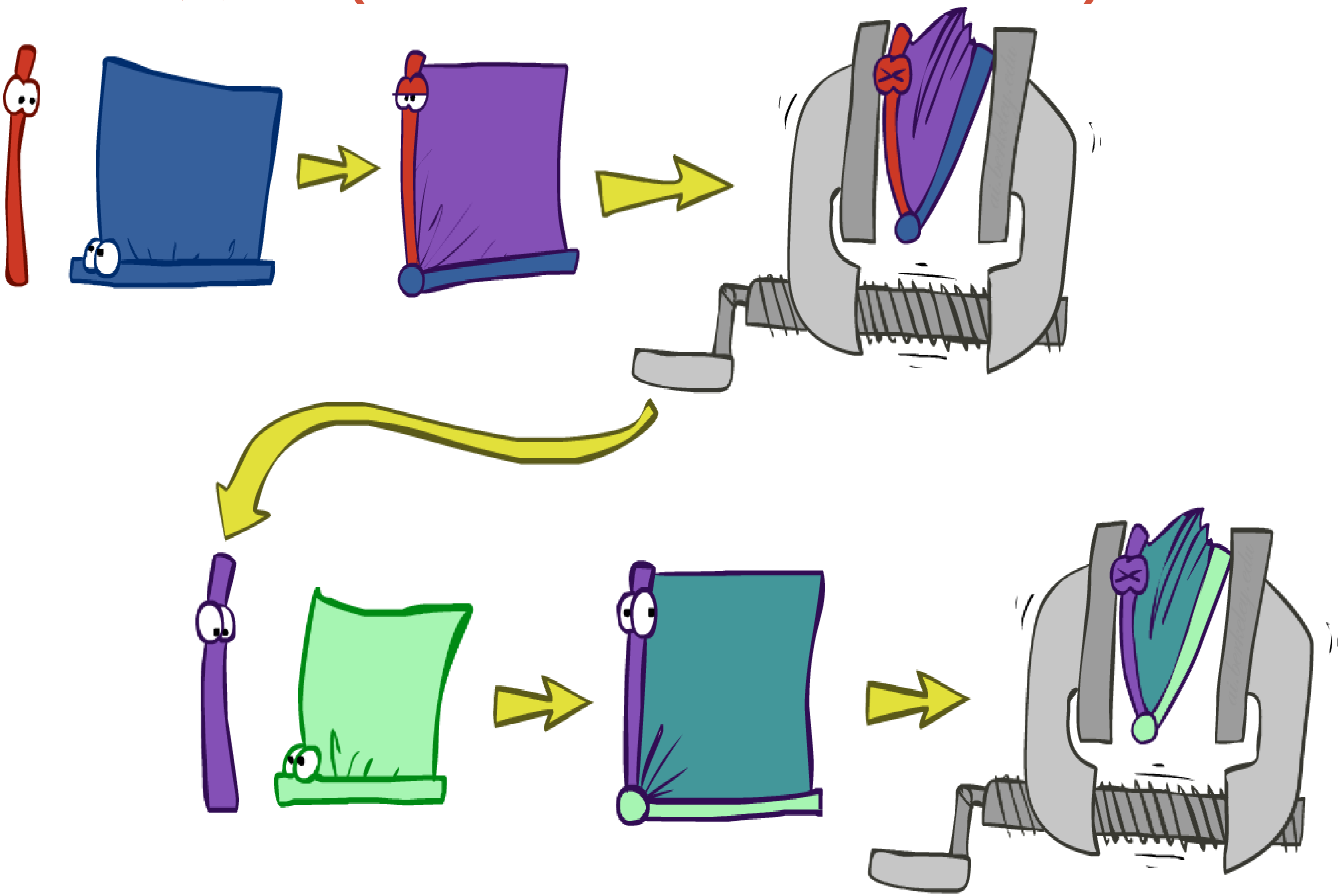
true	0.1
false	0.9

对因子乘积进行加和消除变量操作

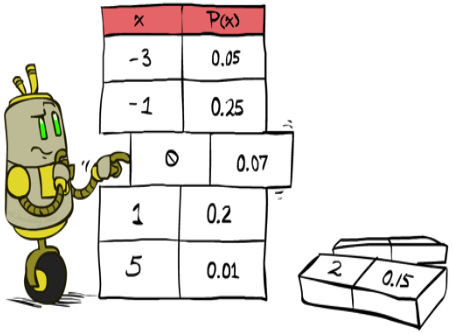
- 对每个因子表先根据实例化的变量值进行筛选（投射），然后再把乘积表达式取和
- 例如:
- $\sum_a P(a|B,e) \times P(j|a) \times P(m|a)$
- $= P(a|B,e) \times P(j|a) \times P(m|a) \quad + \quad P(\neg a|B,e) \times P(j|\neg a) \times P(m|\neg a)$



变量消除法(Variable Elimination)

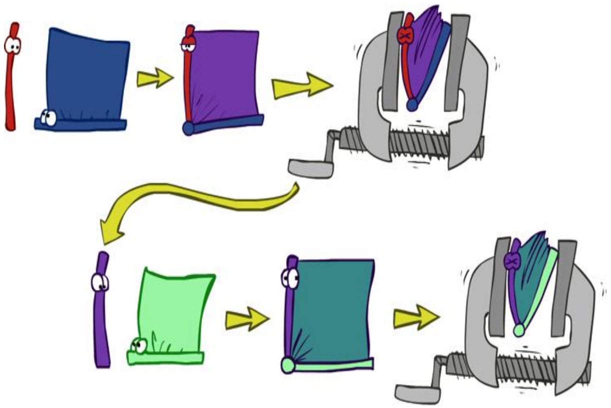


变量消除法



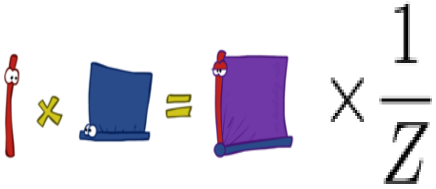
- 查询: $P(Q|E_1 = e_1, \ldots E_k = e_k)$

- 开始于初始的因子表:
  - 局部的条件概率分布表 (CPTs) (但经过观察变量的实例化之后)



- 当仍存在隐藏变量时 (既不是 Q 也不是已观察变量):
  - 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



变量消除法

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

factors ← [ ]

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

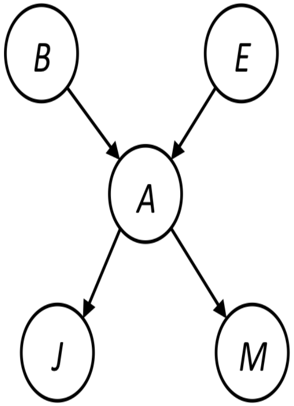
    factors ← [MAKE-FACTOR(*var*, **e**)|*factors*]

    if *var* 是一个隐含变量 **then**

        factors ← SUM-OUT(*var*,*factors*)

return NORMALIZE(POINTWISE-PRODUCT(factors))

举例



$$P(B|j,m) \propto P(B,j,m)$$

$$P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)$$

选择变量 A

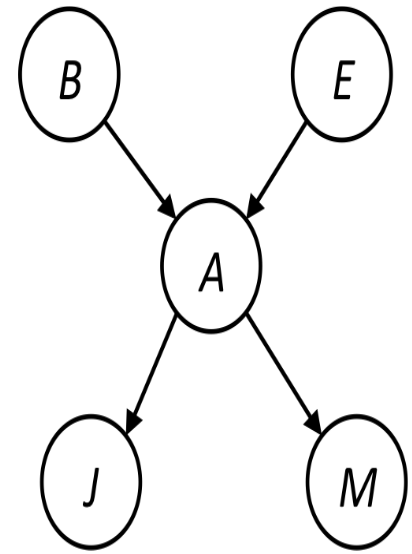
$P(A|B,E)$

$P(j|A)$ $\quad \times \quad \sum \quad$ $P(j,m|B,E)$

$P(m|A)$

$$P(B) \qquad P(E) \qquad P(j,m|B,E)$$

举例



$$P(B) \qquad P(E) \qquad P(j,m|B,E)$$

选择变量 E

$$\left.\begin{matrix} P(E) \\ P(j,m|B,E) \end{matrix}\right. \xrightarrow{\times} \xrightarrow{\Sigma} P(j,m|B)$$

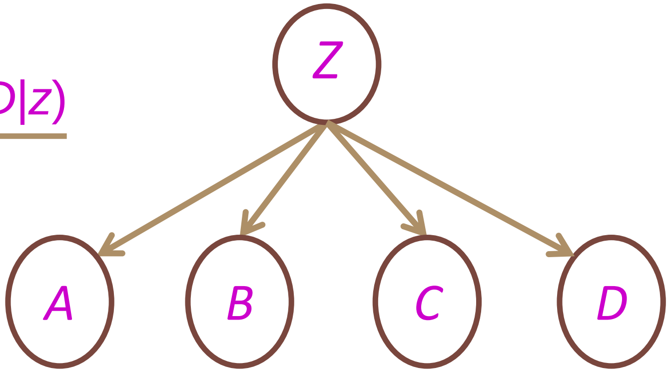
$$P(B) \qquad P(j,m|B)$$

最后是查询变量 B

$$\left.\begin{matrix} P(B) \\ P(j,m|B) \end{matrix}\right. \xrightarrow{\times} P(j,m,B) \xrightarrow{\text{正规化}} P(B|j,m)$$

顺序关系

- 如果排序为 Z, A, B C, D
  - $P(D) = \alpha \sum_{z,a,b,c} P(z)\, P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)$
  - $= \alpha \sum_z P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c \underline{P(c|z)\, P(D|z)}$
  - 最大的因子有2个变量 (D,Z)



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

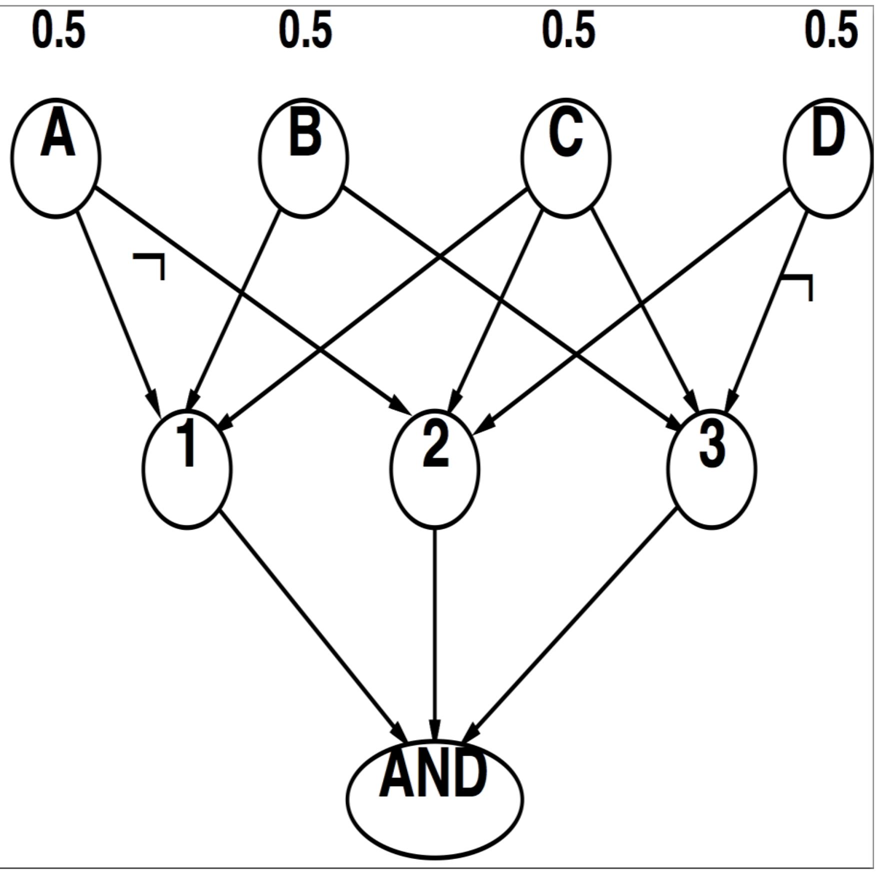
变量消除法: 计算时间和空间复杂度

- 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. 4（2x2）
  - 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!
  - 为什么？

最差情况复杂度? 从 SAT 问题约简过来



- 合取范式(CNF)的子句:
  1. A ∨ B ∨ C
  2. C ∨ D ∨ ¬A
  3. B ∨ C ∨ ¬D
- *P(AND) > 0* 当且仅当 所有子句是可满足的
  - => NP-难度
- P(AND) = S/(2^n), S 是使该合取范式满足的变量配值的组数
  - => #P-难度 （至少和其对应的NP-难度问题一样难）

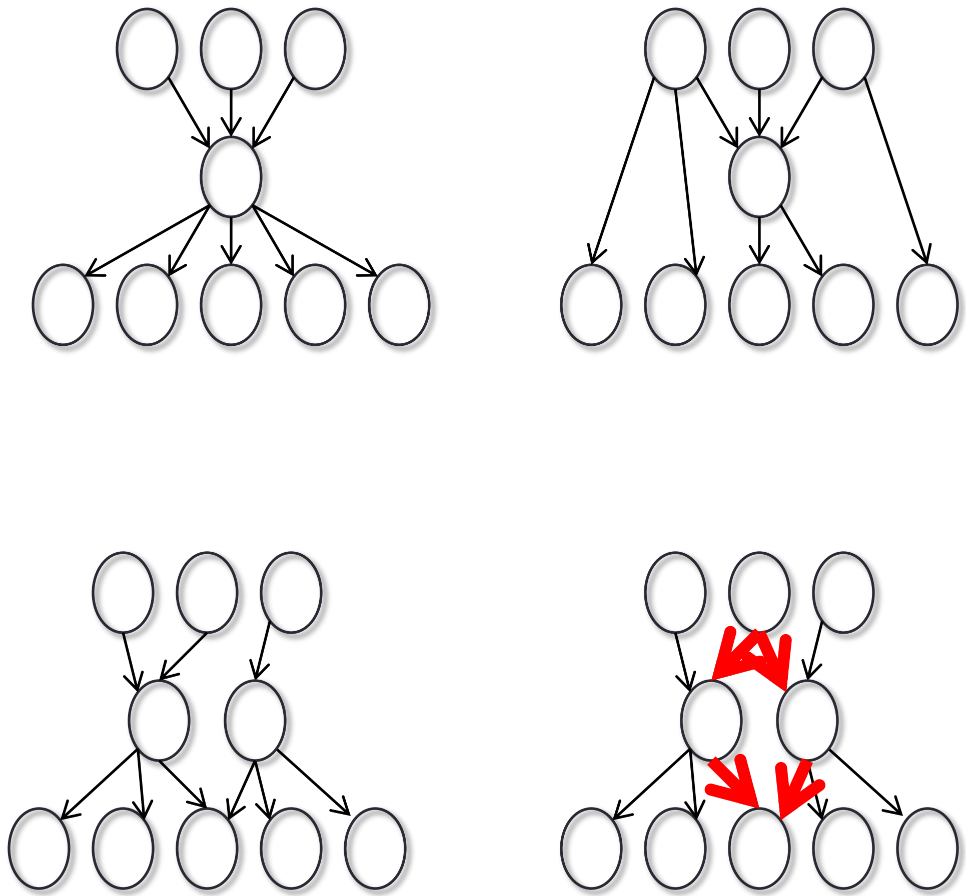
最差情况复杂度? 从 SAT 问题约简过来

- 如果我们能够回答P(AND)=0 或大于0的话，那么我们就已经回答了这个SAT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，变量消除法的复杂度是和网络的大小成线性关系的，当变量消除的顺序是从叶到根的话
  - 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

之后: 从数据学习构建网络结构