

上次的内容

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

今天的内容

- 贝叶斯网络(Bayes nets)
 - 语法和语义（继续）
 - 精确推理
 - 近似推理

贝叶斯网络语法和语义





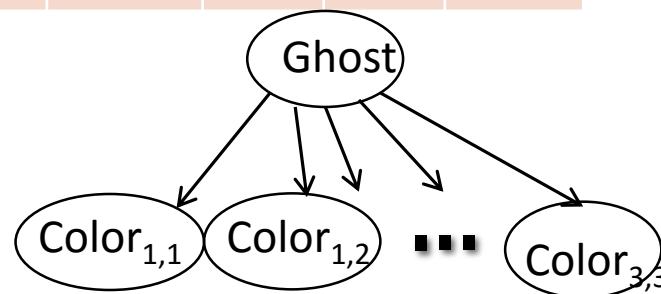
贝叶斯网络语法

- 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

盗窃B

地震E

报警
器响A

约翰打
电话J

玛丽打
电话M

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

2

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(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: $O(d^n)$

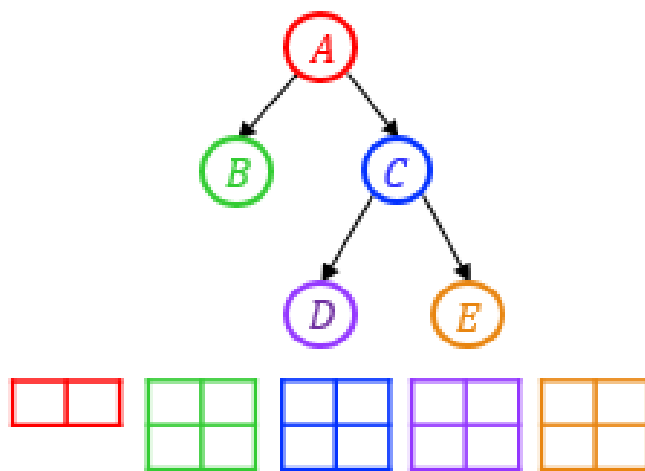
■ 贝叶斯网络的规模: $O(n \cdot d^k)$

- 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

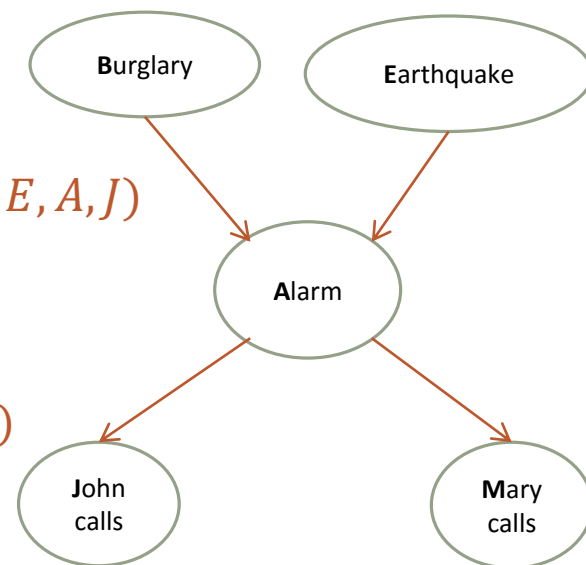
- $P(X_1 \dots X_2) = \prod_i P(X_i | X_1 \dots X_{i-1})$

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

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

贝叶斯网络

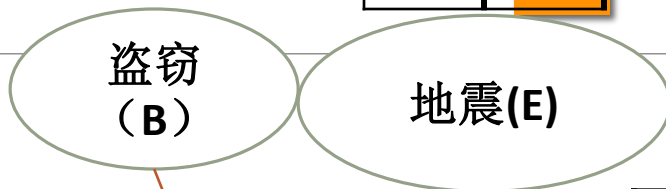
- $P(X_1 \dots X_2) = \prod_i P(X_i | \text{Parents}(X_i))$



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



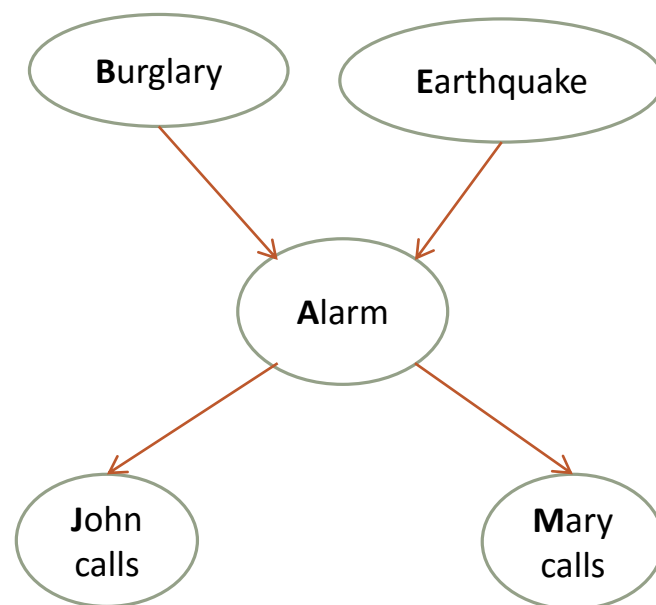
怎样采取这种捷径?

- 从链式法则

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

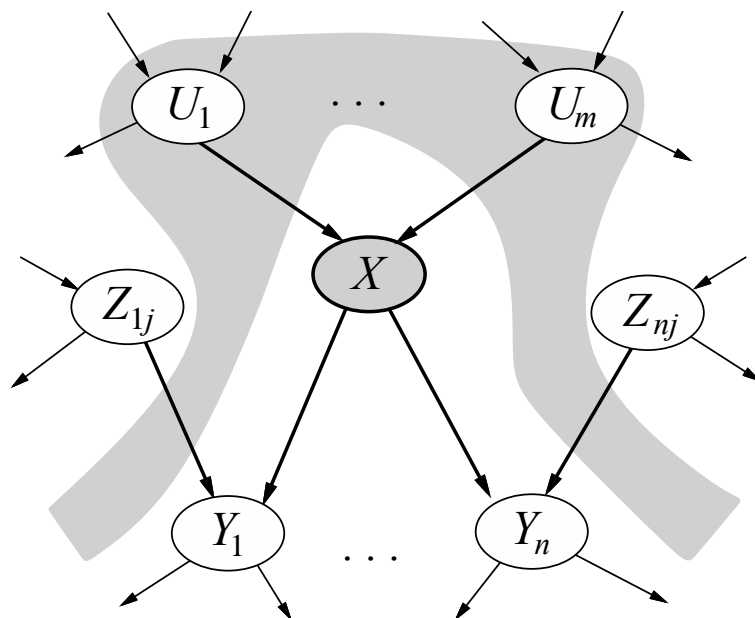
- 转为贝叶斯网络

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



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)



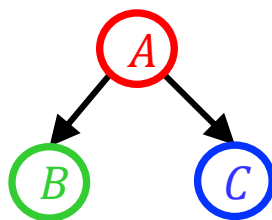
$$P(A) P(B|A) P(C|A, B)$$

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

前提:

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

C 独立于 A 当给定 B 后

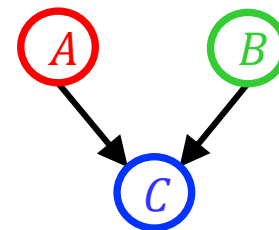


$$P(A) P(B|A) P(C|A, B)$$

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

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

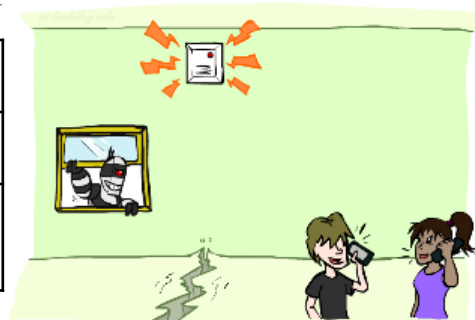
盗窃(B)

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

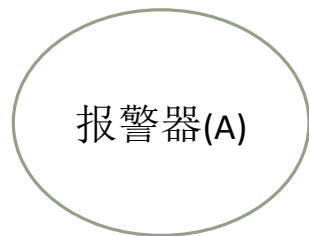


举例：入室偷盗报警

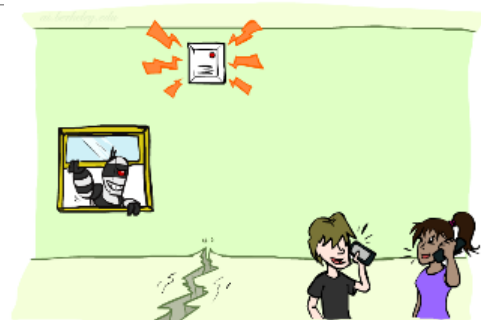
■ 报警器

■ 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)

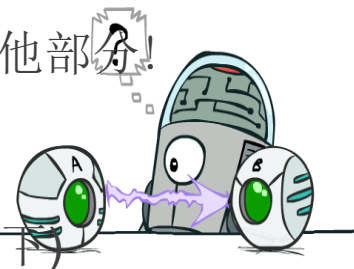
- 其结果是箭头关联反映的是相关性 (correlation), 而不是因果关系

- 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

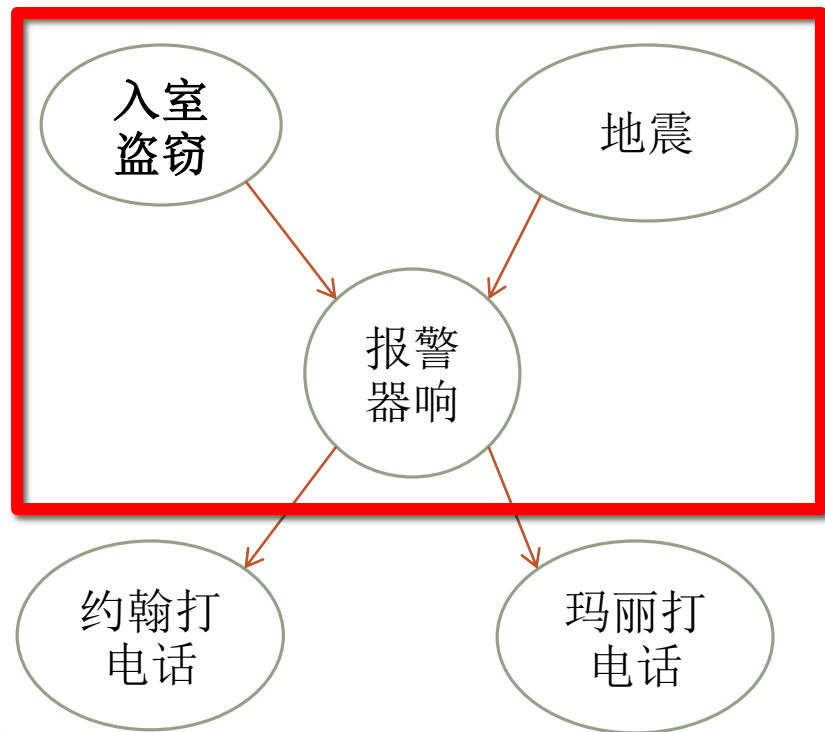
■ 当报警器响后, 盗窃 是否独立于 地震 ？

■ 不是！

■ 报警器已响，入室盗窃和地震都变得很有可能发生过

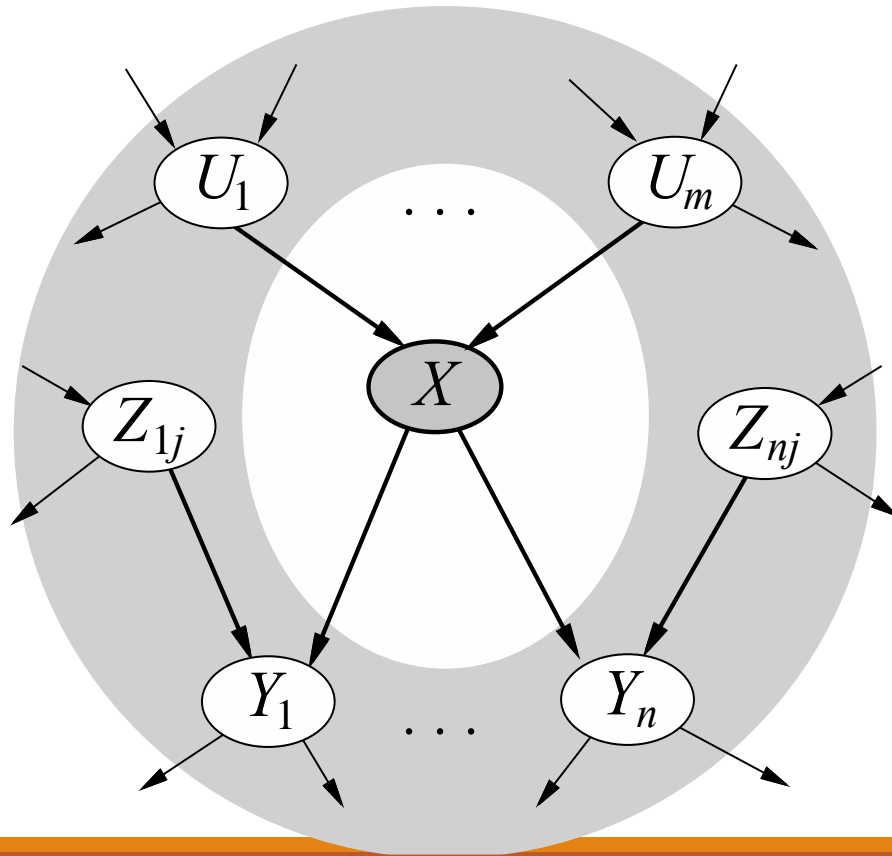
■ 但是，如果我们得知一个入室盗窃已经发生, 那么报警器响的原因被 **解释**，则地震发生的概率降低

V-结构



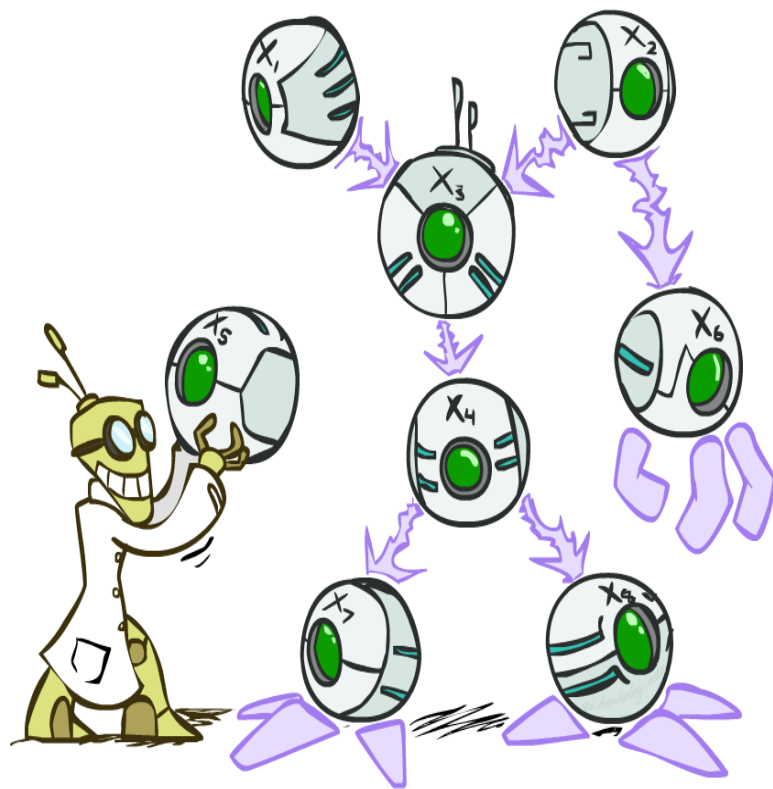
马尔科夫毯(Markov blanket)

- 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

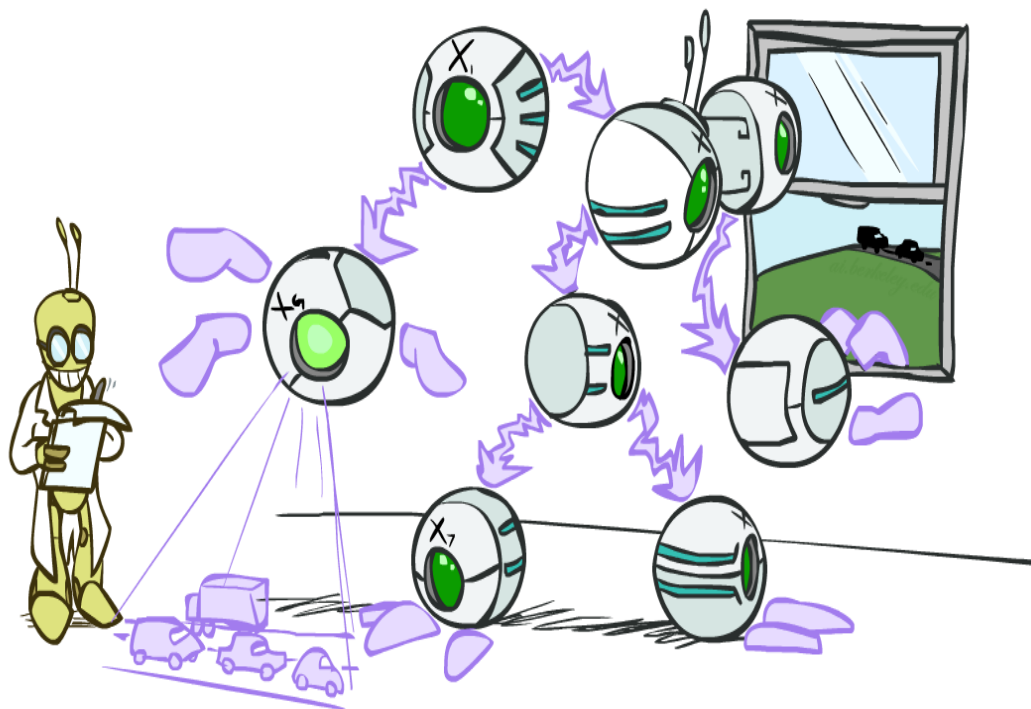


接下来的内容

- 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)

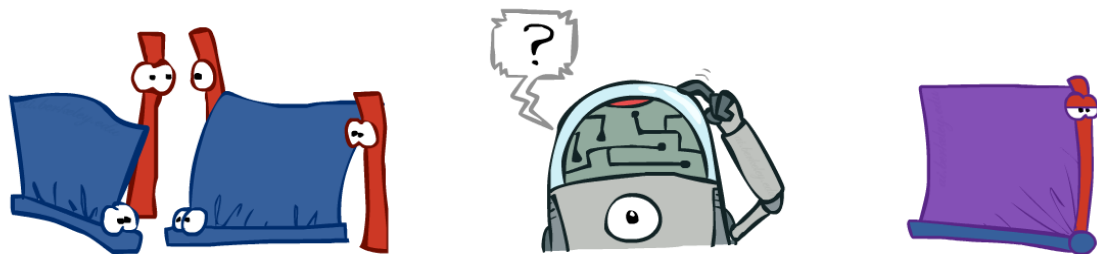
- 列举法 (总是指数复杂度)
- 变量消除法 (最差情况指数复杂度, 通常情况下更好)
- 推理问题是NP-hard

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

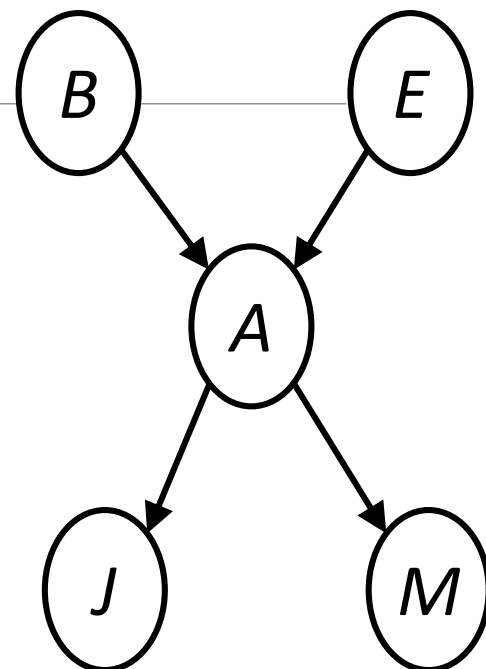
列举法推理回顾:

- 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

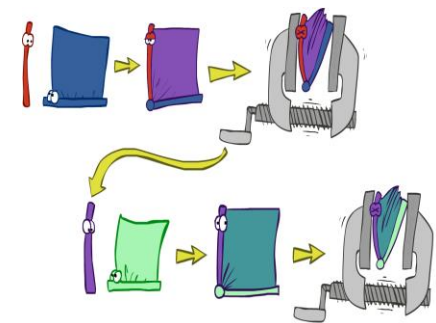
$$\begin{aligned} P(B \mid j, m) &= \alpha \sum_e \sum_a P(B) P(e) P(a \mid B, e) P(j \mid a) P(m \mid a) \\ &= \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) P(j \mid a) P(m \mid a) \end{aligned}$$

- 计算顺序由里向外

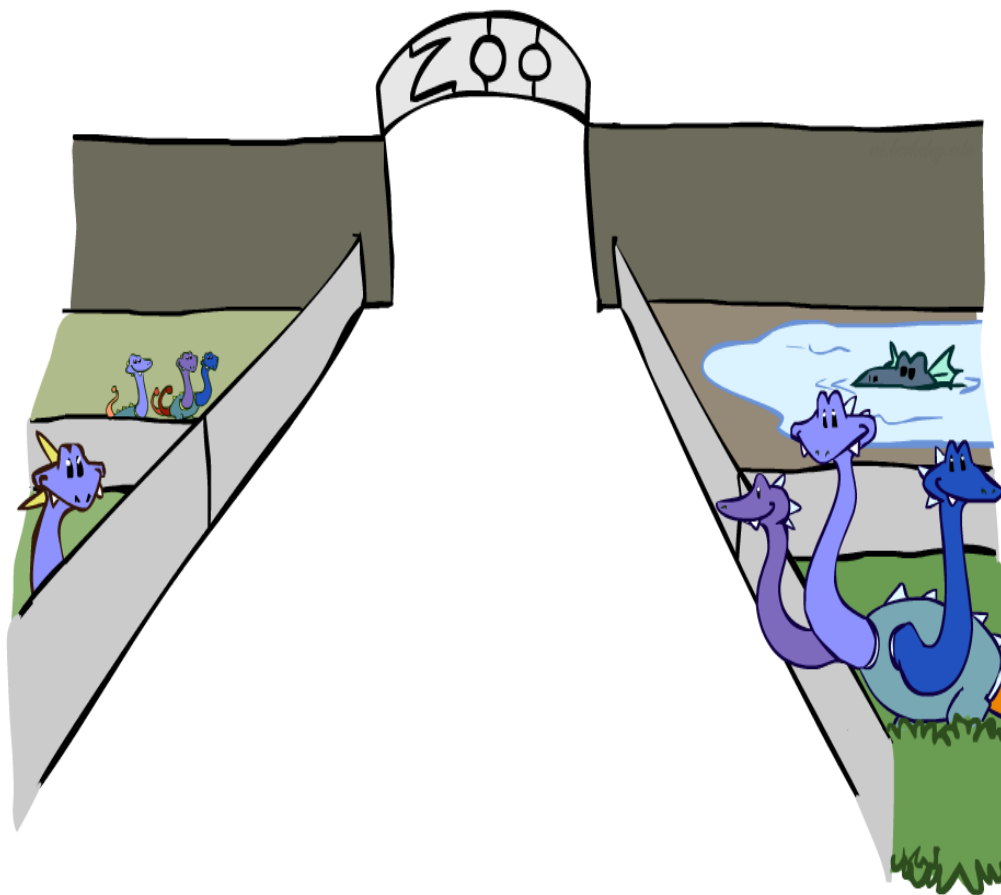
- 即, 先在 a 上求和, 再在 e 上求和

- 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

■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

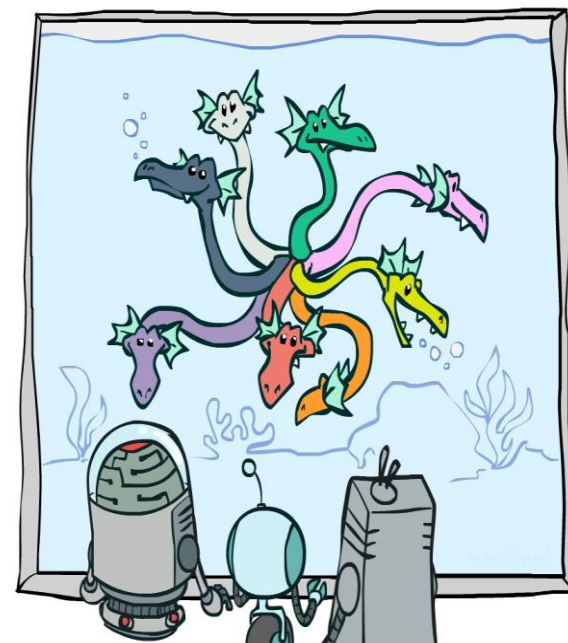
■数组之和为 $P(x)$

$P(A,J)$

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

$P(a,J)$

$A \setminus J$	true	false
true	0.09	0.01



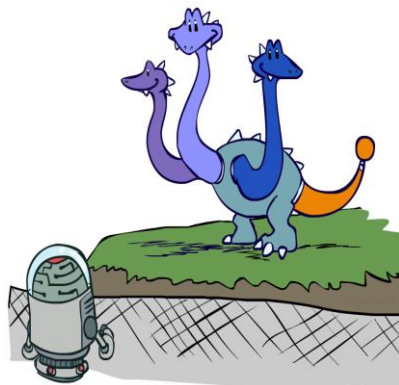
变量数之和 (大写的字符) = 表的维度

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



$P(J | a)$

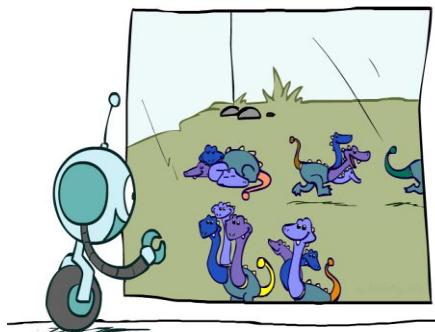
$A \setminus J$	true	false
true	0.9	0.1

■ 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 $|Y|$



$P(J | A)$

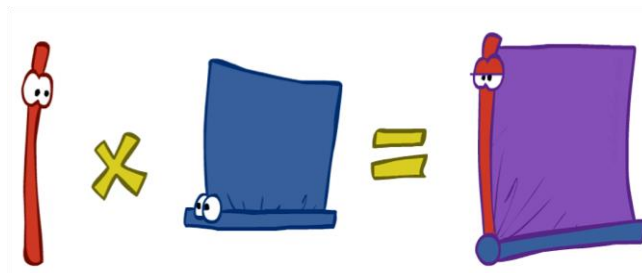
$A \setminus J$	true	false
true	0.9	0.1
false	0.05	0.95

} - $P(J | a)$

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

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: $P(J|A) \times P(A) = P(A,J)$

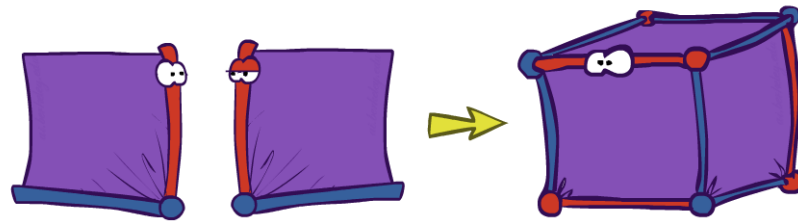
$P(A)$		\times	$P(J A)$			$=$	$P(A,J)$		
			A \ J	true	false		A \ J	true	false
true	0.1		true	0.9	0.1		true	0.09	0.01
false	0.9		false	0.05	0.95		false	0.045	0.855

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

$=$

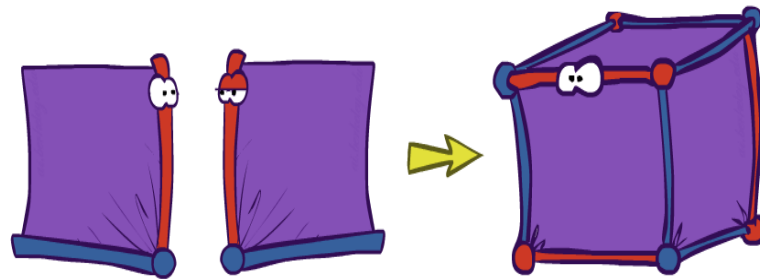
$P(J,M/A)$

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

$P(A, J)$

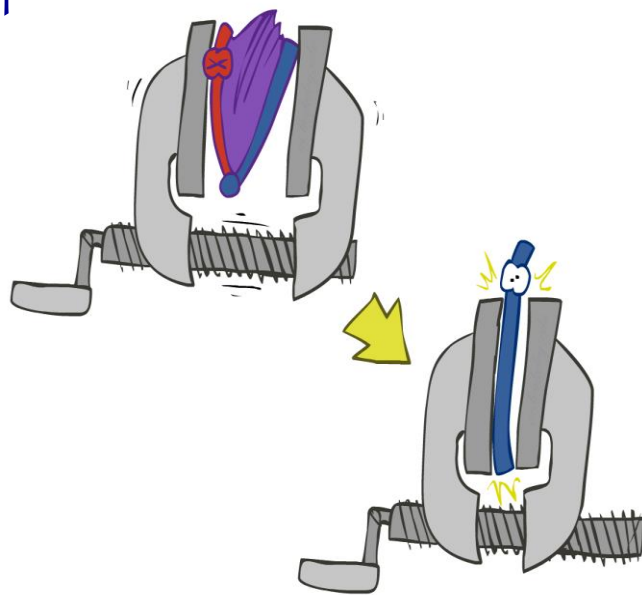
A \ J	true	false
true	0.09	0.01
false	0.045	0.855

加和消掉 J



$P(A)$

true	0.1
false	0.9

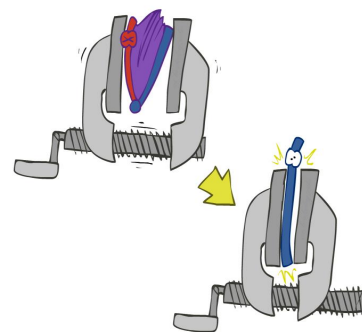


加和消除作用于因子乘积

■ 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：

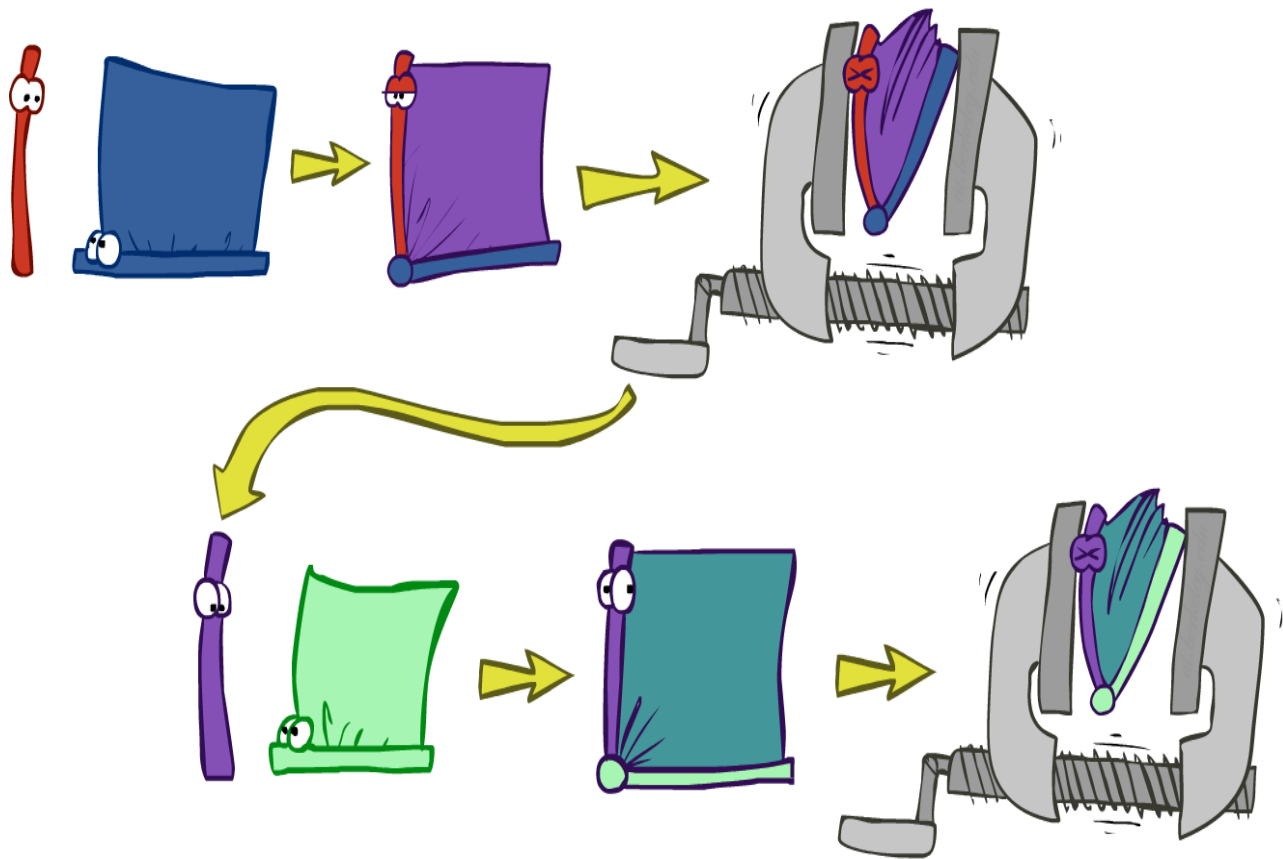
$$\begin{aligned} \circ \quad & \sum_a P(a|B,e) P(j|a) P(m|a) \\ &= P(a|B,e) P(j|a) P(m|a) + P(\neg a|B,e) P(j|\neg a) P(m|\neg a) \\ &= P(a,j,m|B,e) + P(\neg a,j,m|B,e) \\ &= P(j,m|B,e) \end{aligned}$$



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)

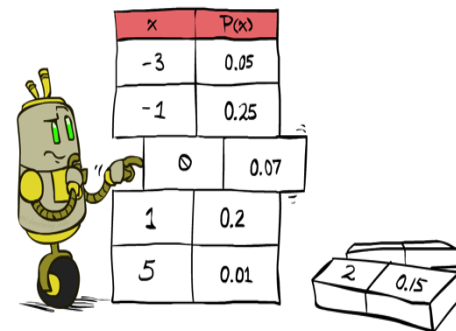
■ 当仍存在隐藏变量时 (既不是 Q 也不是E):

■ 选一个隐含变量 H

■ 合并所有包含 H 的因子表

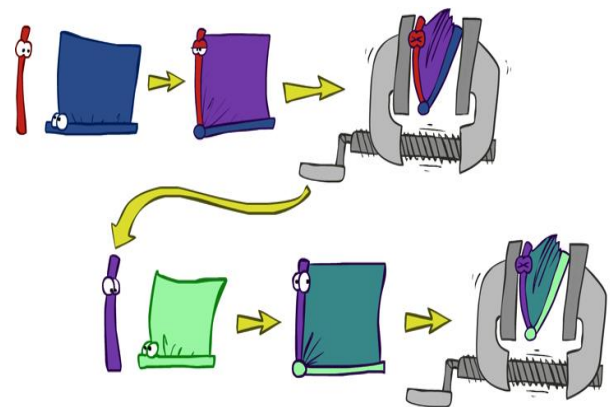
■ 消除变量 (通过取和) H

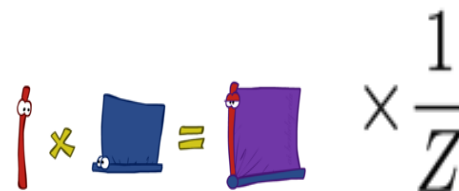
■ 合并所有剩余因子表, 并对结果进行正规化



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**

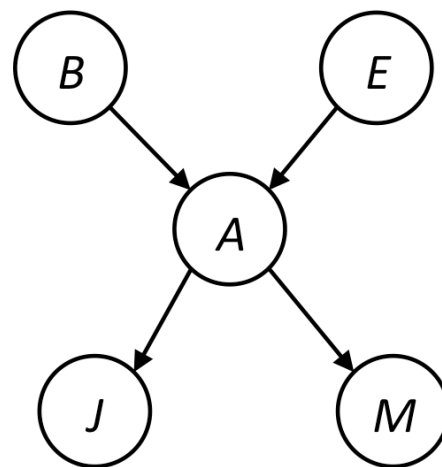
$factors \leftarrow SUM-OUT(var, factors)$

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 $P(B \mid j, m)$

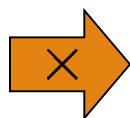
$P(B)$	$P(E)$	$P(A \mid B, E)$	$P(j \mid A)$	$P(m \mid A)$
--------	--------	------------------	---------------	---------------

选择 A

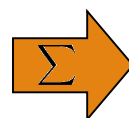
$P(A \mid B, E)$

$P(j \mid A)$

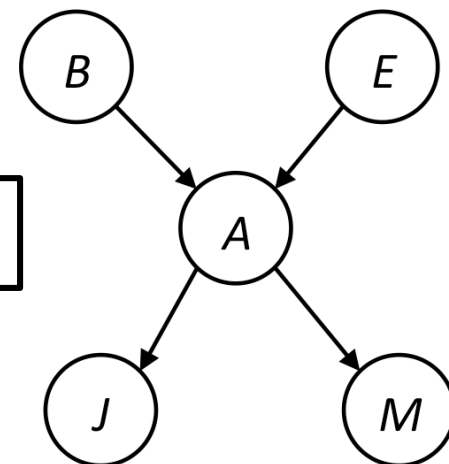
$P(m \mid A)$



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



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



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

举例

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

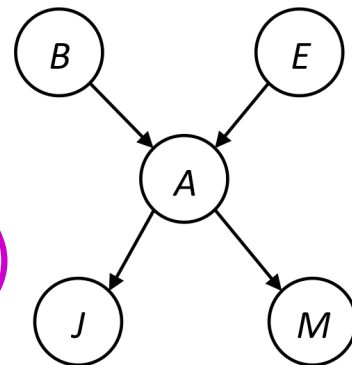
选择 E

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

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

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

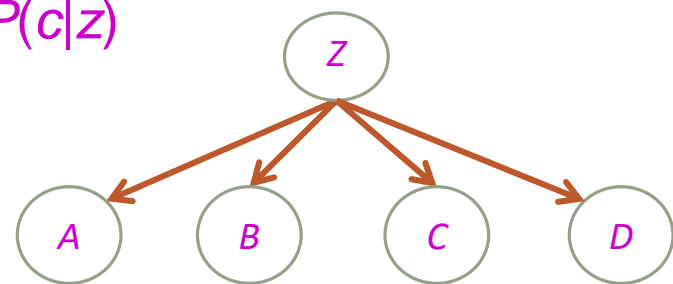
- 如果排序为 **D, Z, A, B C**

- $P(D) = \alpha \sum_{z,a,b,c} P(D|z) P(z) P(a|z) P(b|z) P(c|z)$
- $= \alpha \sum_z P(D|z) P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z)$
- 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)

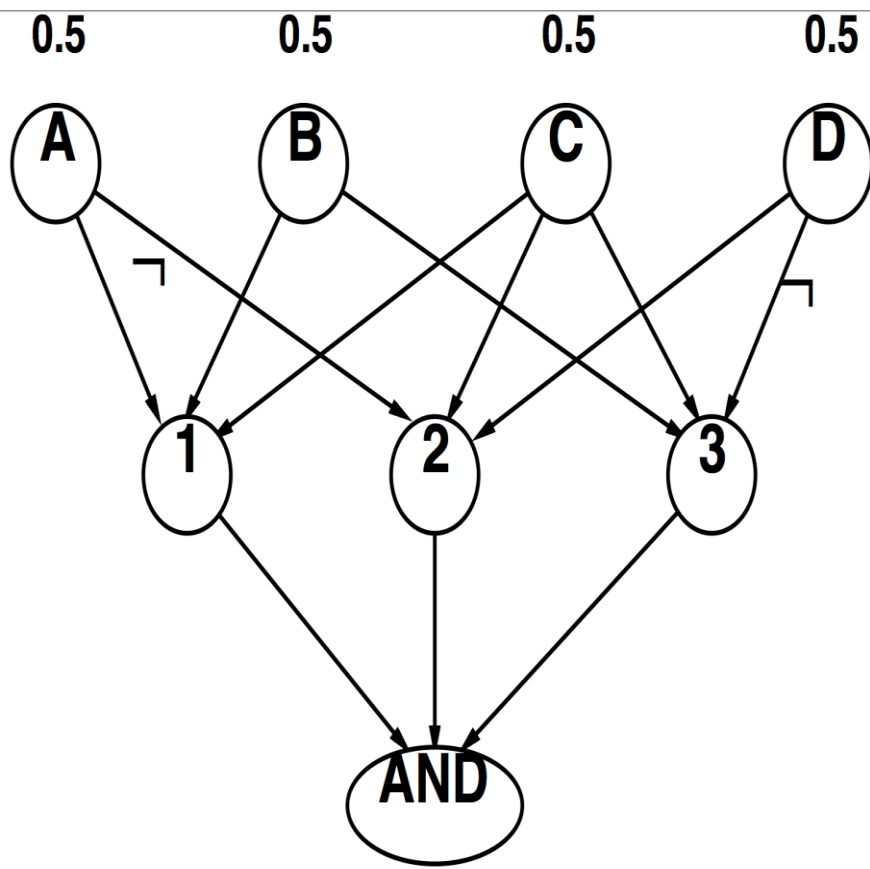
- 通常, 如果有 n 个叶节点, 因子表的大小是 2^n



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

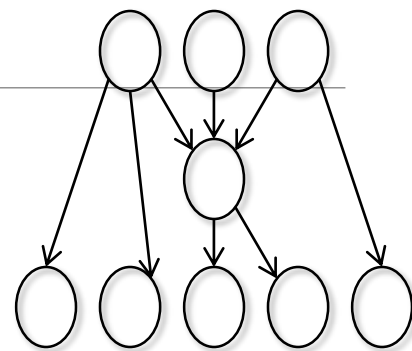
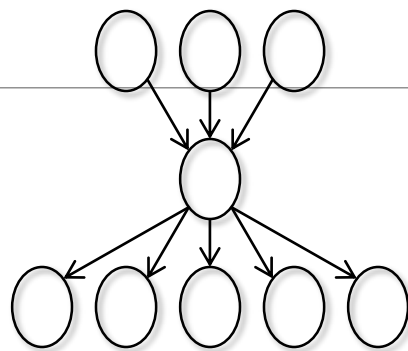
■ \Rightarrow #P-难度 (至少和其对应的NP-难度问题一样难, 或更难的)

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。

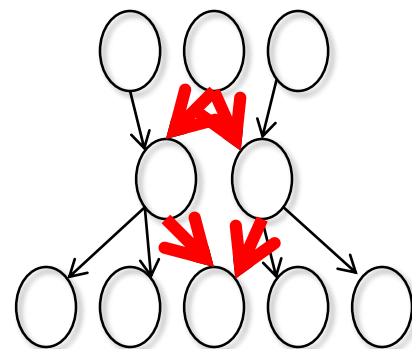
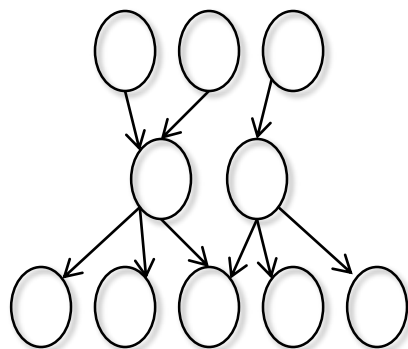
多树 (Polytrees)

■ 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)

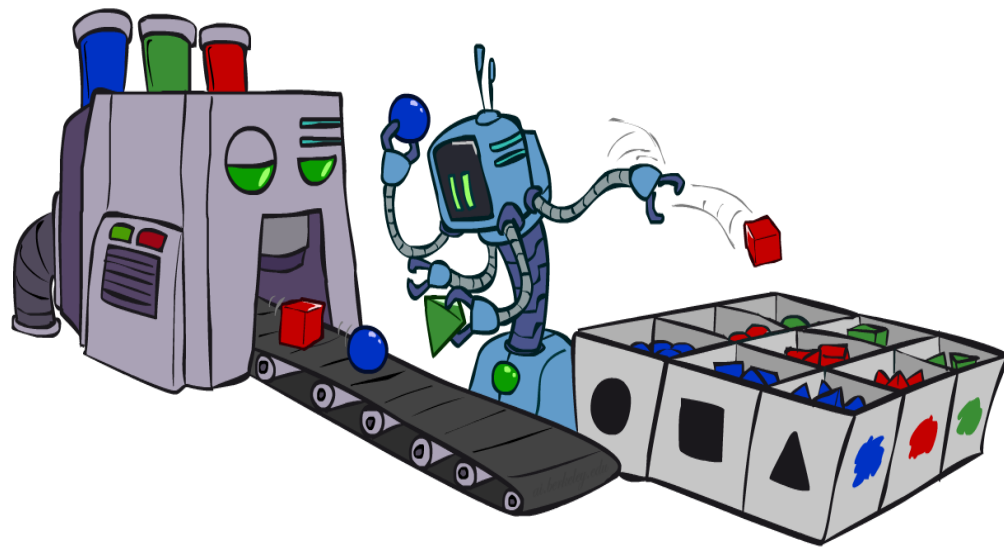
Part III: 近似推理

接下来的内容

- 贝叶斯网络上的近似推理

人工智能导论

贝叶斯网络：近似推理 APPROXIMATE INFERENCE



采样 (Sampling)

采样很像重复的模拟

■ 基本思想

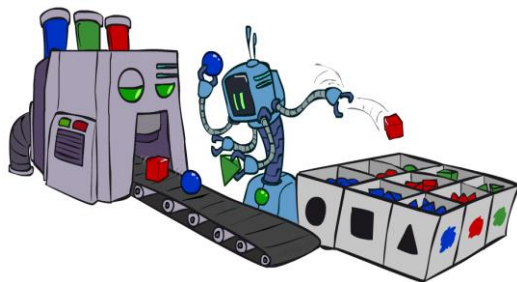
■ 抽取 N 样本，形成一个采样分布 S

■ 计算一个近似后验概率

■ 证明可以收敛到真实的概率 P

■ 为什么采样？

- 通常很快得到一个好的近似解
- 算法简单而且通用 (很容易应用在不同的概率模型上)
- 算法只需很少的存储空间 ($O(n)$)
- 可以应用于大的模型上；对比准确算法（比如变量消除法）



举例

- 假设你有两个大富翁游戏的智能体程序 **A** 和 **B**
- **A** 获胜的概率是多少?
 - 方法 1:
 - 让 **s** 是一序列的骰子数，机会和公益金牌
 - 给定 **s**, 结果 $V(s)$ 可能是 1（赢）, 0（输）
 - **A** 赢的概率是 $\sum_s P(s) V(s)$
 - 问题: 无限多这样的序列 **s**!
 - 方法 2:
 - 采样 **N** (也许 100) 组序列从概率分布 $P(s)$, 即玩 **N** 次游戏
 - **A** 获胜的概率大概是 $(1/N) \sum_i V(s_i)$ 即在采样里获胜的比例

从一个离散分布中采样

■ 举例

■ 步骤 1: 获取一个采样 u 从均匀分布 $[0, 1)$

■ 例如 `random()`

■ 步骤 2: 把这个采样值 u 转化成一个给定分布的输出结果。（通过关联每个输出结果 x 和一个 $P(x)$ -大小的在 $[0,1)$ 上的一个子区间）

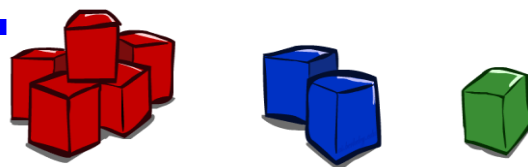
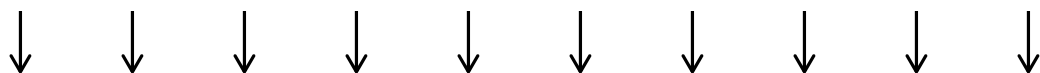
C	P(C)
red	0.6
green	0.1
blue	0.3

$0.0 \leq u < 0.6, \rightarrow C=\text{red}$

$0.6 \leq u < 0.7, \rightarrow C=\text{green}$

$0.7 \leq u < 1.0, \rightarrow C=\text{blue}$

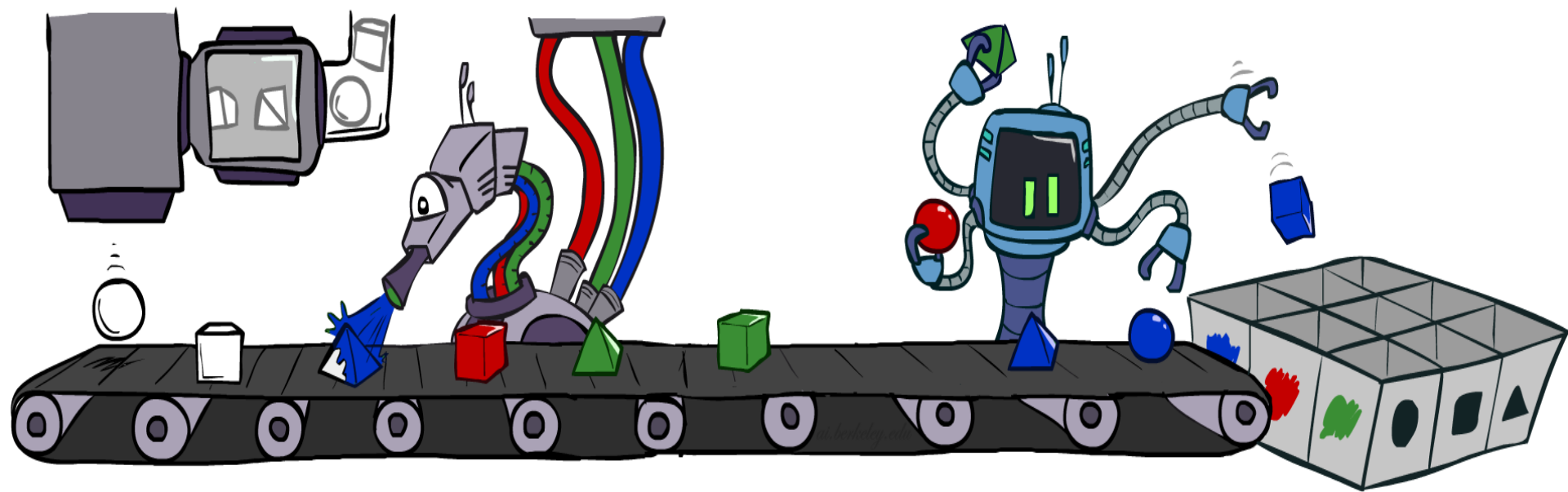
- 如果 `random()` 返回 $u = 0.83$, 那么采样为 $C = \text{blue}$
- 再例如, 在8次采样以后有:



贝叶斯网络里的采样

- 先验采样 (Prior Sampling)
- 拒绝抽样 (Rejection Sampling)
- 似然性/可能性加权 (Likelihood Weighting)
- 吉布斯采样 (Gibbs Sampling)

先验采样(Prior Sampling)



先验采样

$$P(C)$$

c	0.5
$\neg c$	0.5

$$P(S|C)$$

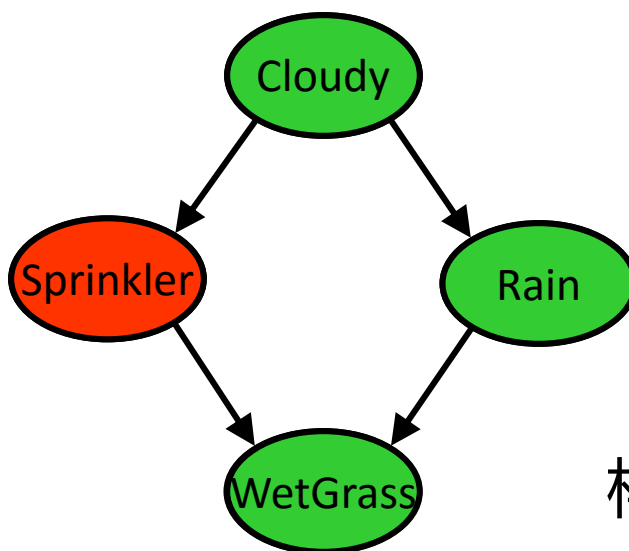
c	s	0.1
	$\neg s$	0.9
$\neg c$	s	0.5
	$\neg s$	0.5

$$P(R|C)$$

c	r	0.8
	$\neg r$	0.2
$\neg c$	r	0.2
	$\neg r$	0.8

$$P(W|S, R)$$

s	r	w	0.99
		$\neg w$	0.01
	$\neg r$	w	0.90
		$\neg w$	0.10
$\neg s$	r	w	0.90
		$\neg w$	0.10
	$\neg r$	w	0.01
		$\neg w$	0.99



样本:

c, $\neg s$, r, w(这个例子里)

$\neg c$, s, $\neg r$, w

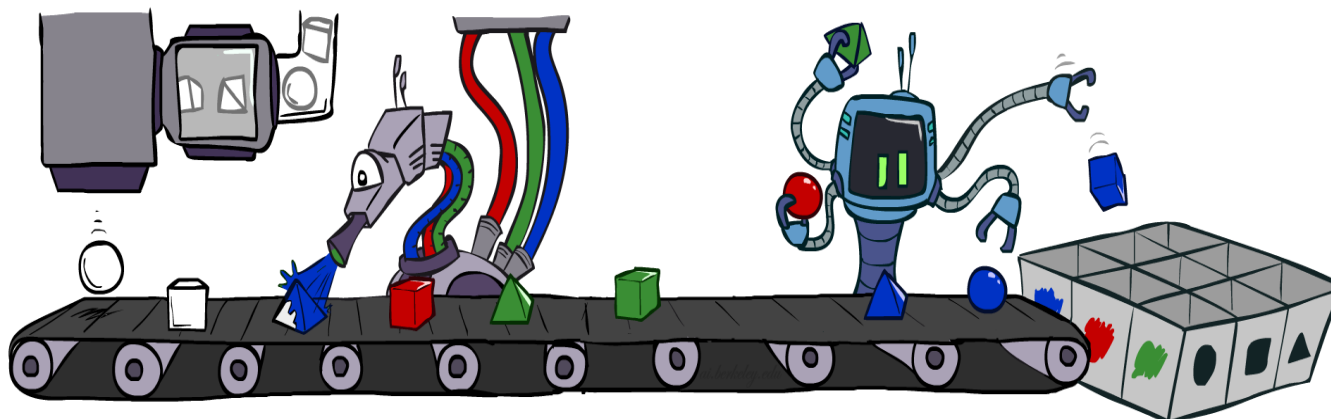
...

先验采样

For $i=1, 2, \dots, n$ (按拓扑顺序)

- 采样 x_i 从 $P(x_i | \text{parents}(x_i))$

Return (x_1, x_2, \dots, x_n)



先验采样

- 这个过程产生这样的样本的概率是:
-

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

...即 是贝叶斯网络的联合概率

- 让 一个事件的样本数为 $N_{PS}(x_1 \dots x_n)$

- 那么
$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n) / N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$

- 即, 这个采样过程是 一致的/连续的 (**consistent**)

例如

我们从这个贝叶斯网络里获得一系列的样本：

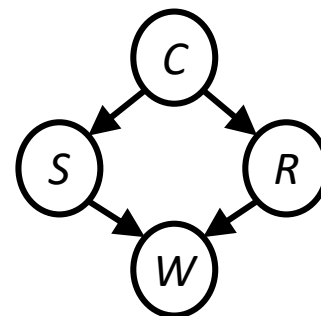
$C, \neg S, r, W$

C, S, r, W

$\neg C, S, r, W$

$C, \neg S, r, W$

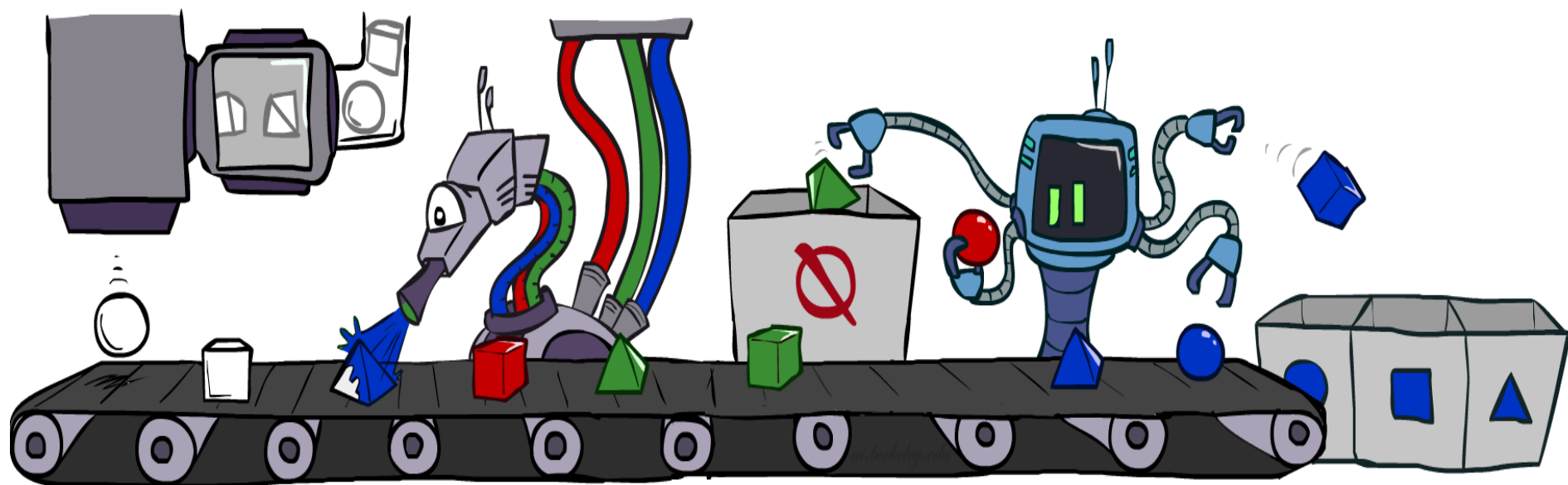
$\neg C, \neg S, \neg r, W$



如果我们想知道： $P(W)$

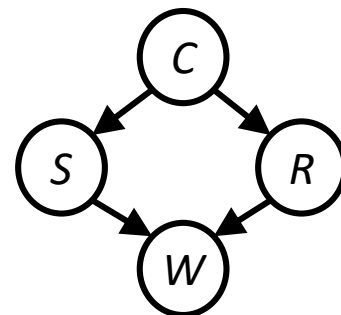
- 我们可以数出 $\langle w:4, \neg w:1 \rangle$
- 正规化后得到 $P(W) = \langle w:0.8, \neg w:0.2 \rangle$
- 样本越多，越接近真实的分布
- 还可以估计其他的概率量
- 比如，想查询概率 $P(C | r, w)$ ， 使用 $P(C | r, w) = \alpha P(C, r, w)$

拒绝采样(Rejection Sampling)



拒绝采样

- 为了计算条件概率，对先验采样进行简单修改
- 假如我们想计算 $P(C | r, w)$
- 计算采样中 C 的结果，但是忽略（拒绝）那些不含有 $R=\text{true}$, $W=\text{true}$ 的样本
- 这就叫做拒绝采样
- 对于条件概率的估计，也是满足一致性的（即， N 趋于无限大时，等于理论真值）



$C, \neg S, r, w$

~~$C, S, \neg r$~~

~~$\neg C, S, r, \neg w$~~

~~$C, \neg S, \neg r$~~

$\neg C, \neg S, r, w$

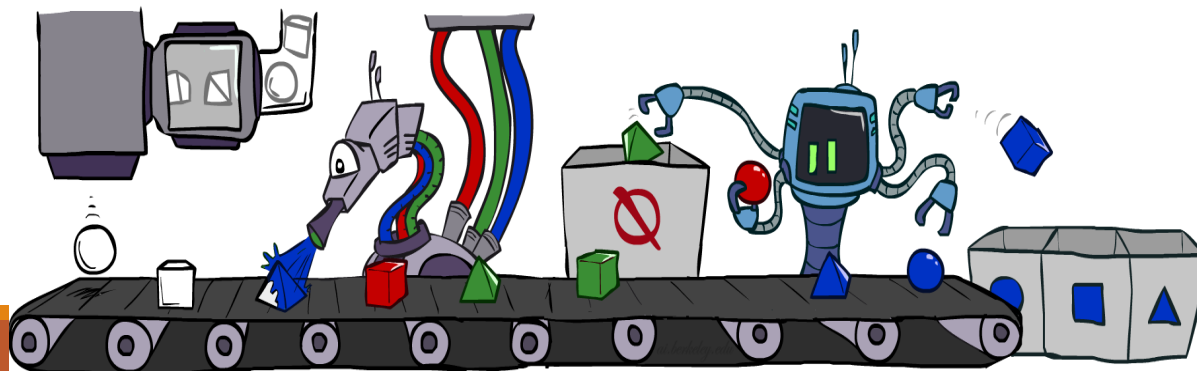
拒绝采样 Rejection Sampling

输入: 观察值 e_1, \dots, e_k

For $i=1, 2, \dots, n$

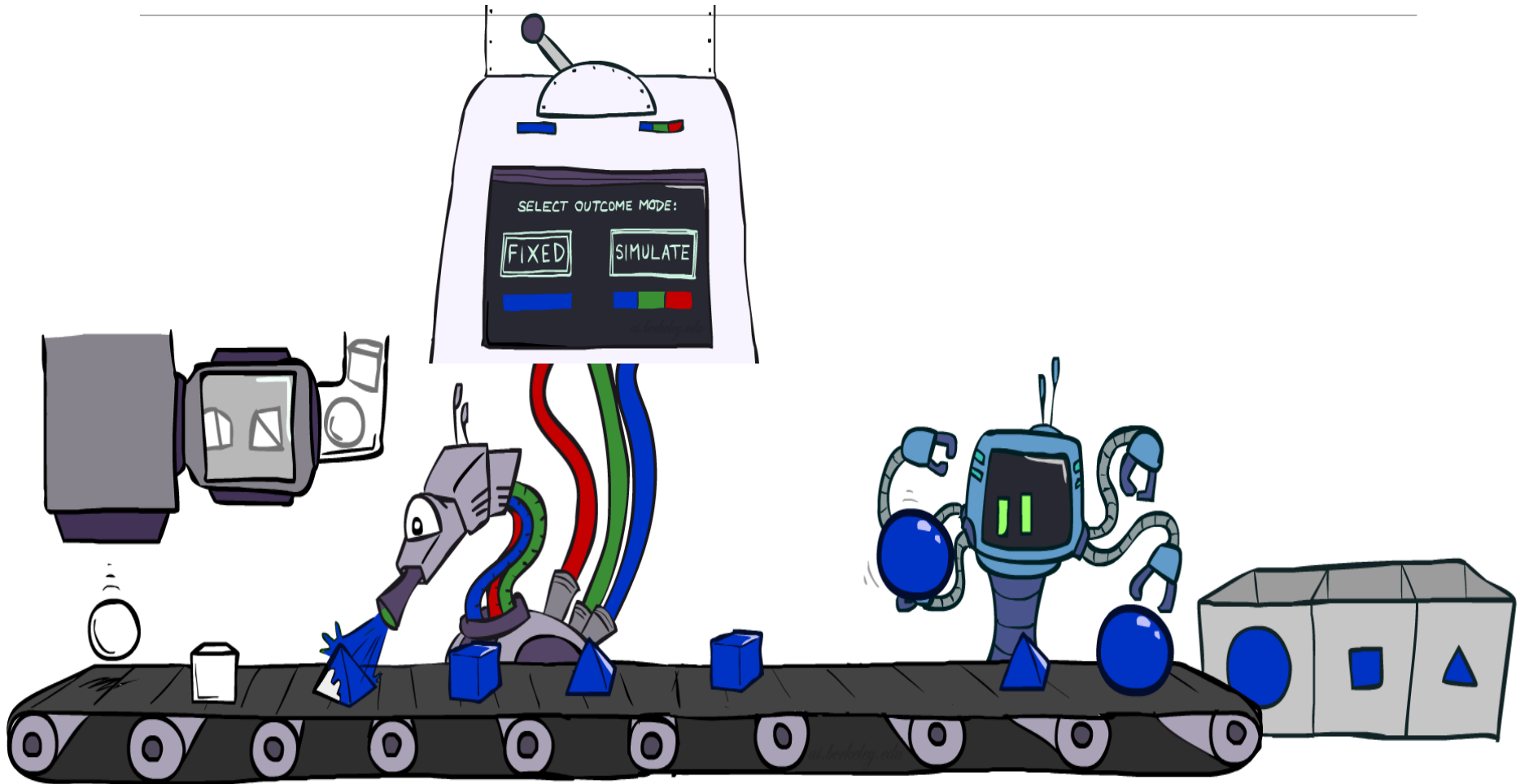
- 采样 X_i 从 $P(X_i | \text{parents}(X_i))$
- 如果 x_i 和观察值不一致
 - 拒绝这个样本: **Return**, 则在这个循环里没有样本产生

Return (x_1, x_2, \dots, x_n)



似然性加权（采样）

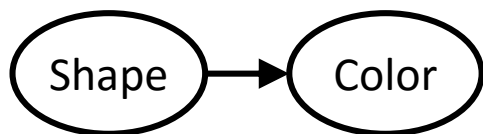
Likelihood Weighting



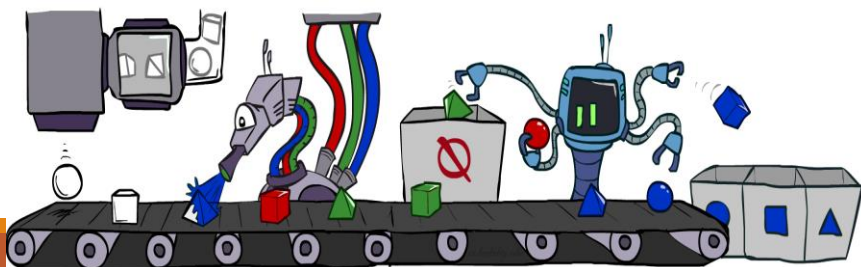
似然性加权（采样）

拒绝采样法的问题:

- 有可能拒绝许多样本，尤其当观察变量很多时
- 采样时没有利用已被观察变量的值
- 比如考虑 $P(\text{Shape}|\text{Color}=\text{blue})$

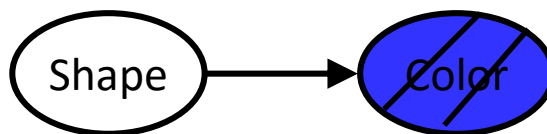


~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~

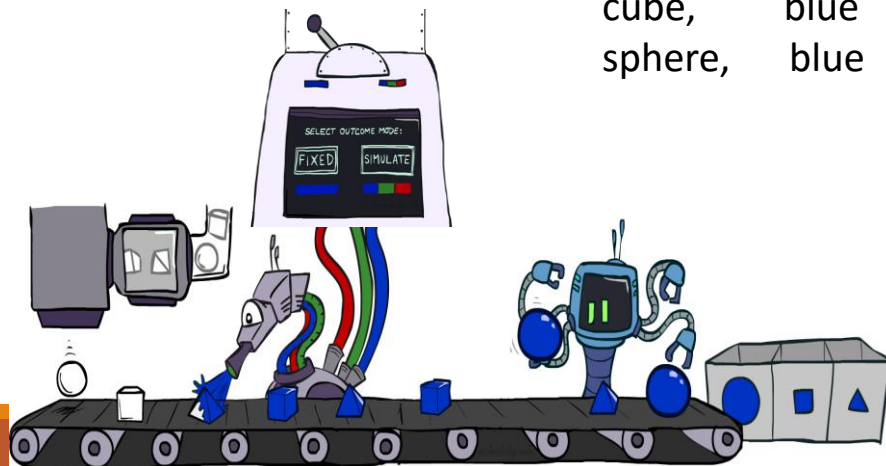


■ 想法: 固定观察变量的值，对其他变量值进行采样

- 问题: 样本分布与理论分布不一致!
- 解决办法: **权重** 每个样本，通过使用观察变量给定父变量的概率



pyramid, blue
pyramid, blue
sphere, blue
cube, blue
sphere, blue



似然性加权（采样）

w 初始化为1.0;
拓扑排序: C, S, R, W
S, W 值固定为真

$$P(C)$$

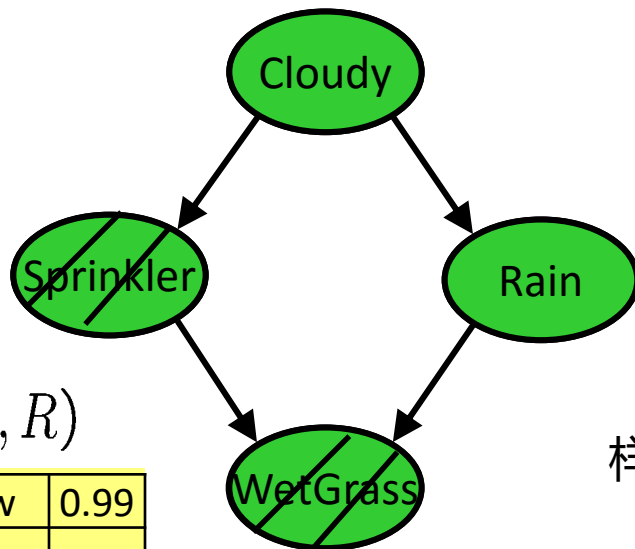
c	0.5
$\neg c$	0.5

$$P(S|C)$$

c	s	0.1
	$\neg s$	0.9
$\neg c$	s	0.5
	$\neg s$	0.5

$$P(R|C)$$

c	r	0.8
	$\neg r$	0.2
$\neg c$	r	0.2
	$\neg r$	0.8


$$P(W|S, R)$$

s	r	w	0.99
		$\neg w$	0.01
	$\neg r$	w	0.90
		$\neg w$	0.10
$\neg s$	r	w	0.90
		$\neg w$	0.10
	$\neg r$	w	0.01
		$\neg w$	0.99

样本事件:

c, s, r, w

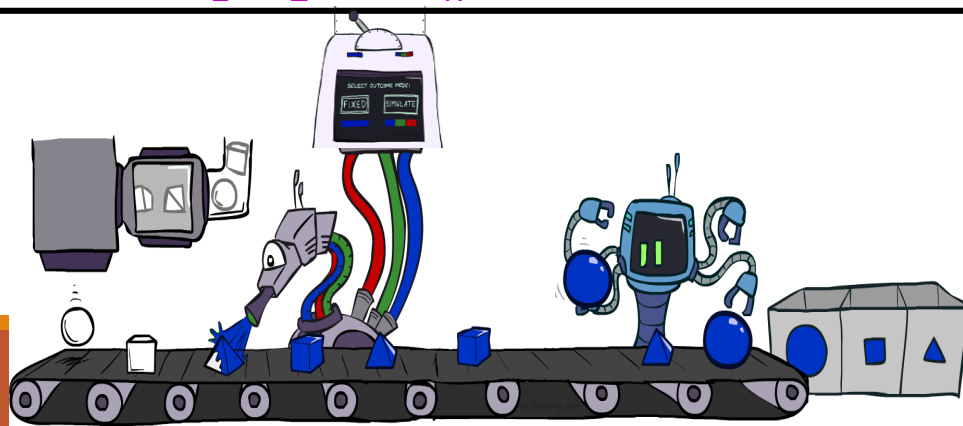
...

样本事件的权值:

$$w = 1.0 \times 0.1 \times 0.99$$

似然性加权采样

- 输入: 观察值 e_1, \dots, e_k
- $w = 1.0$
- for $i=1, 2, \dots, n$
 - 如果 x_i 是已观察的变量 (evidence variables)
 - x_i = 观察到的 value _{i} for X_i
 - 让 $w = w * P(x_i \mid \text{Parents}(X_i))$
 - 否则
 - 抽样 x_i 从 $P(X_i \mid \text{Parents}(X_i))$
- return $(x_1, x_2, \dots, x_n), w$



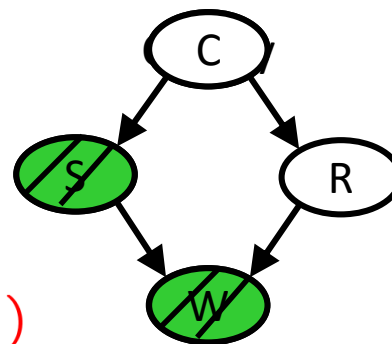
似然性加权采样

- 采样分布为（ \mathbf{z} 为非观察变量的采样值 \mathbf{e} 为固定的观察值）

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^l P(z_i | \text{Parents}(Z_i))$$

- 现在, 每个样本都有权重

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^m P(e_i | \text{Parents}(E_i))$$



- 合起来, 加权的样本分布是具有一致性的, 即:

$$\begin{aligned} S_{WS}(\mathbf{z}, \mathbf{e}) \cdot w(\mathbf{z}, \mathbf{e}) &= \prod_{i=1}^l P(z_i | \text{Parents}(z_i)) \prod_{i=1}^m P(e_i | \text{Parents}(e_i)) \\ &= P(\mathbf{z}, \mathbf{e}) \end{aligned}$$

似然性加权

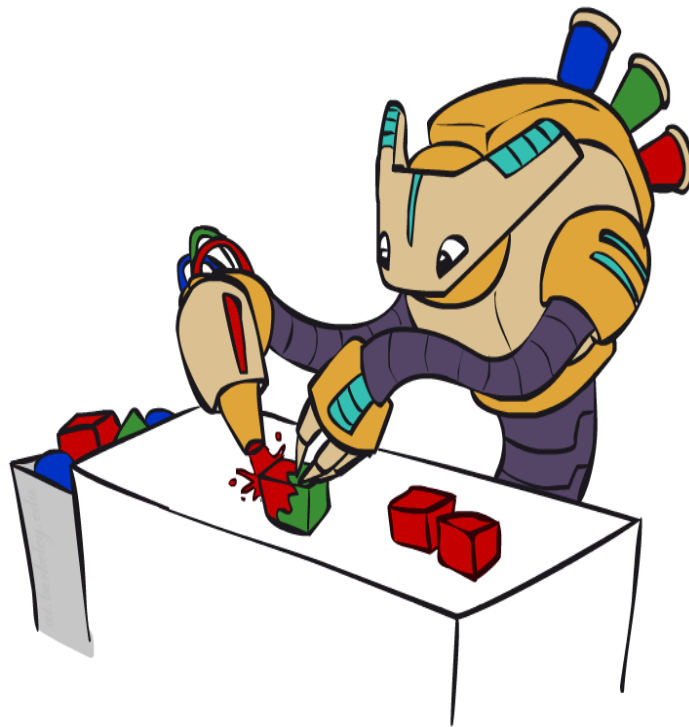
■ 优点:

- 可利用所有样本（加权的）
- **下游** 变量的采样值会被 **上游** 已观察变量的值所影响

■ 也有弱点:

- **上游** 变量的采样值 不受 **下游** 观察变量值的影响
- 假设观察到的值在 k 个叶节点上, 那么样本的权重可能为 $O(2^{-k})$
- 随着观察变量的增多, 而且如果这些变量出现在拓扑顺序的后面, 那么许多样本的权值会很小, 只有极少的幸运样本将有相对很大的权值, 从而主导估计概率的结果
- 我们希望的是, 每个变量都可以“看见” **所有** 已观察到的值!

吉布斯采样(Gibbs Sampling)



马尔科夫蒙特卡洛 (Markov Chain Monte Carlo)

- MCMC (Markov chain Monte Carlo) 是随机算法家族一员，用来估计某个感兴趣的数值在一个很大的状态空间里
 - Markov chain = 一序列随机选择的状态 (“随机漫步 random walk”), 其中每个状态的选择是基于它前一个状态
 - Monte Carlo = 摩纳哥的旅游城市，有一个著名的赌场



马尔科夫蒙特卡洛理论

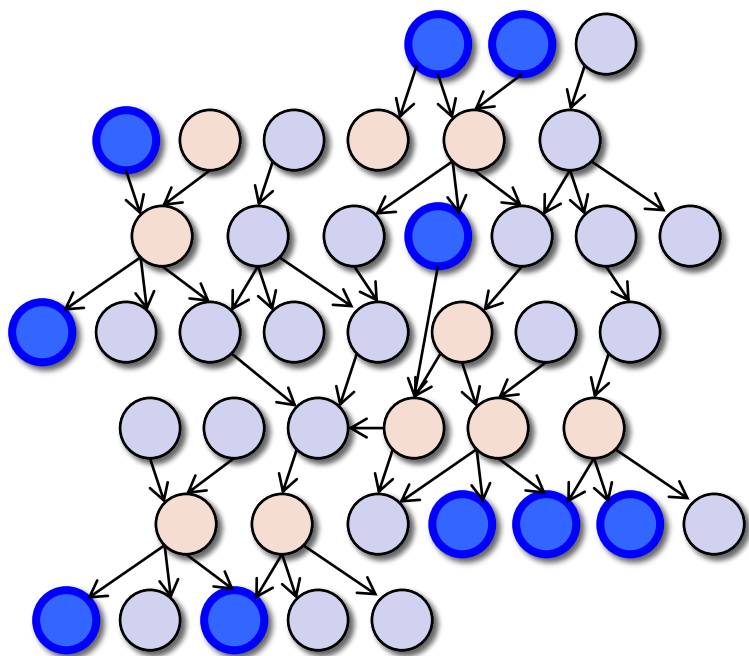
Markov Chain Monte Carlo

- MCMC 属于随机算法家族，用于在一个很大的状态空间里，近似估计某些感兴趣的量
 - 马尔科夫链 = 一序列随机选择的状态 (“随机漫步random walk”), 每个状态的选择是条件取决于前一个状态
 - 蒙特卡洛理论 Monte Carlo = 一种算法 (通常基于随机采样), 存在产生一个不正确解答的可能性 (概率)
- MCMC = 随机漫步一会，平均化你所观察到的情况

吉布斯采样 (Gibbs sampling)

- 属于 MCMC 家族一类
 - 状态是对所有变量的完整的赋值
 - (对比局部搜索里的 模拟退火算法, 属于同一算法家族!)
 - 观察 (证据) 变量的值固定, 改变其他变量的值
 - 当产生下一个状态时, 选出一个变量, 并对其采样一个值, 采样的分布是条件于所有其他变量
 - $X_i' \sim P(X_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$
 - 趋向于朝高概率发生的状态移动, 但也可能移动到一个低概率的状态
 - 在贝叶斯网络里, $P(X_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(X_i | \text{马可夫毯}(X_i))$
- 定理: 吉布斯采样是具有一致性的
 - 给定吉布斯分布概率是远离0和1, 并且变量选择是公平的

为什么这样做？



采样很快开始反应网络里所有的观察值（已观察节点的值对其他变量值的采样施加影响）

最终样本将从真实的后验概率分布上抽取！

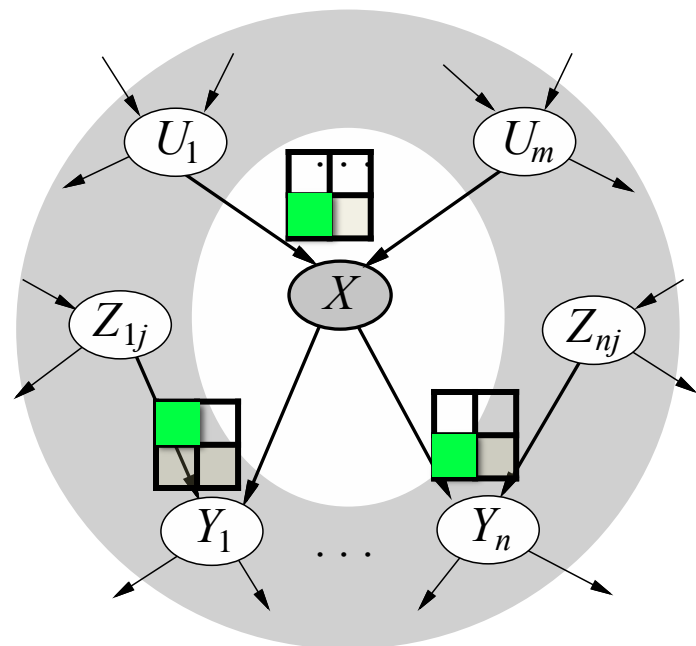
如何进行采样？

■ 重复许多次：

■ 对一个非观察到的变量 X_i 进行采样，从概率分布：

■ $P(X_i \mid x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(X_i \mid \text{马尔科夫毯}(X_i))$

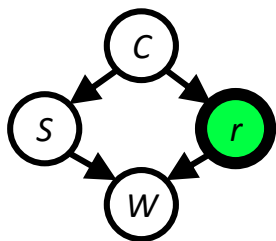
■ $= \alpha P(X_i \mid u_1, \dots, u_m) \prod_j P(y_j \mid \text{parents}(Y_j))$



吉布斯采样举例: $P(S | r)$

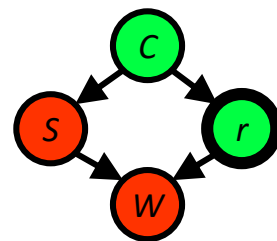
Step 1: 固定观察值

- $R = \text{true}$



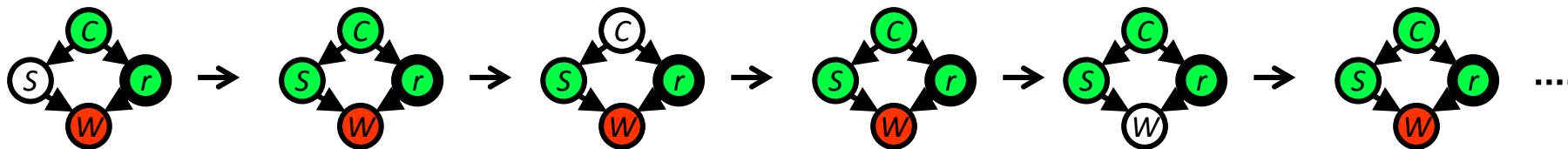
Step 2: 初始化其他变量

- 随机地



Step 3: 重复以下

- 选择一个非证据变量 X
- 重采样 X 从 $P(X | \text{马可夫毯}(X))$



采样 $S \sim P(S | c, r, \neg w)$

采样 $C \sim P(C | s, r)$

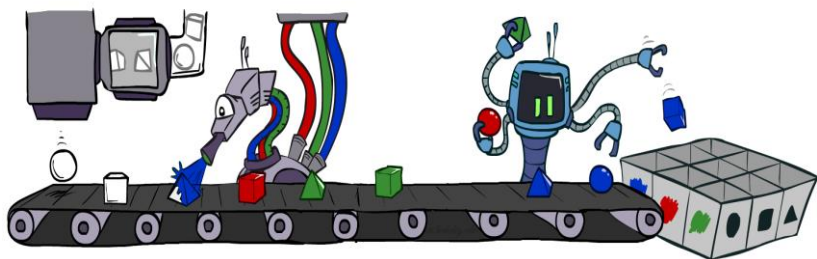
采样 $W \sim P(W | s, r)$

为什么这种方法有效?

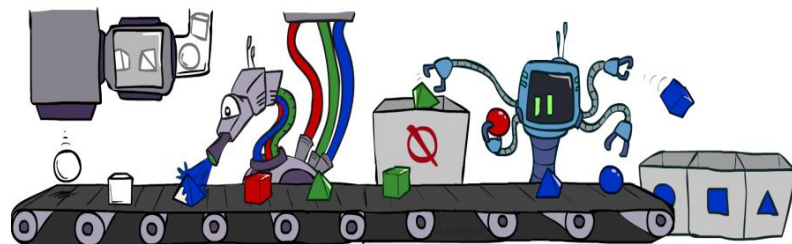
- 假定运行这种方法很长一段时间, 并预测在时刻 t 到达任何一个状态的概率为: $\pi_t(x_1, \dots, x_n)$ or $\pi_t(\underline{x})$
- 对每个吉布斯采样步骤 (挑一个变量, 重采样它的值) 当它应用到一个状态 \underline{x} 时, 有一个概率 $q(\underline{x}' | \underline{x})$ 移动到下个状态 \underline{x}'
- 所以 $\pi_{t+1}(\underline{x}') = \sum_{\underline{x}} q(\underline{x}' | \underline{x}) \pi_t(\underline{x})$ 或, 用矩阵或向量形式表示:
$$\pi_{t+1} = Q\pi_t$$
- 当这一动态过程处于平衡, 即 $\pi_{t+1} = \pi_t$, 所以 $Q\pi_t = \pi_t$
- 这种情况下有一个唯一解, 即 $\pi_t = P(x_1, \dots, x_n | e_1, \dots, e_k)$
- 所以当时刻 t 足够大时, 下一个样本将会从真实的后验条件概率分布上被采集

贝叶斯网络采样技术小结

■ 先验采样 P



■ 拒绝采样法 $P(Q | e)$



■ 吉布斯采样 $P(Q | e)$

■ 似然加权采样法 $P(Q | e)$

