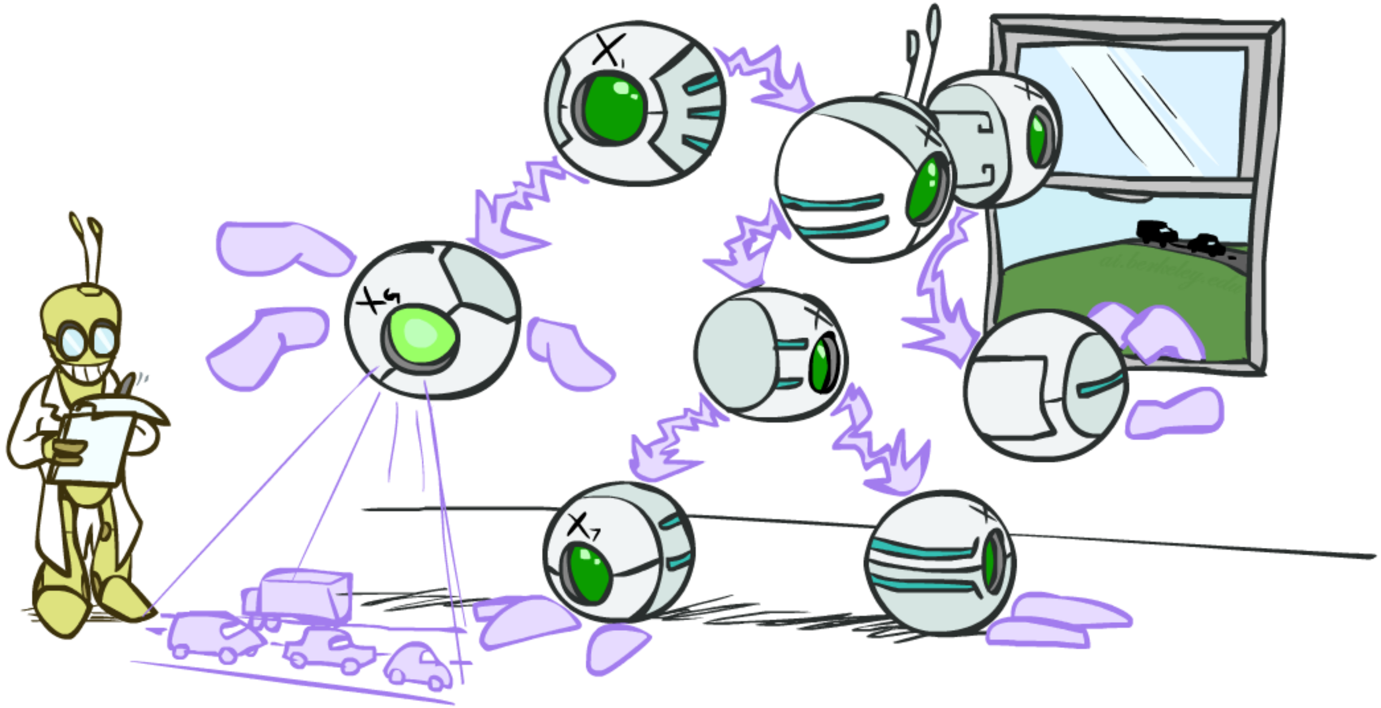
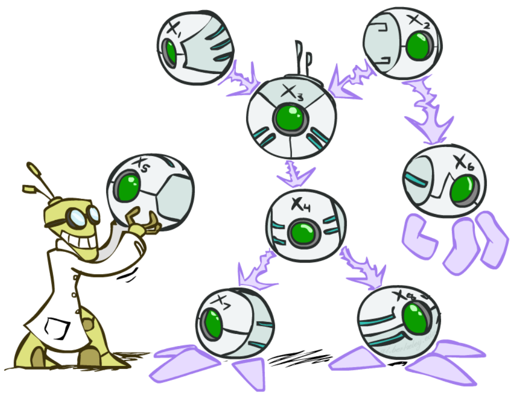
贝叶斯网络(BAYES NETS)：推理



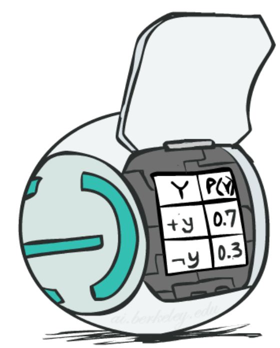
Bayes' Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values $P(X|a_1 \ldots a_n)$



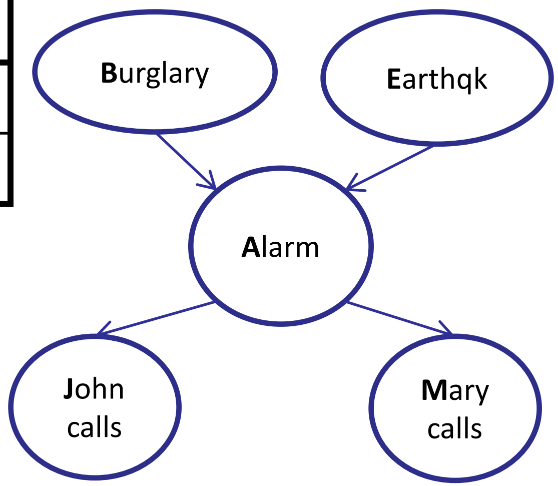
- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\textit{parents}(X_i))$$

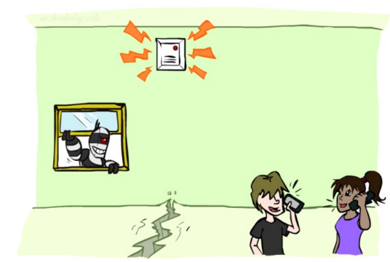


Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998



B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

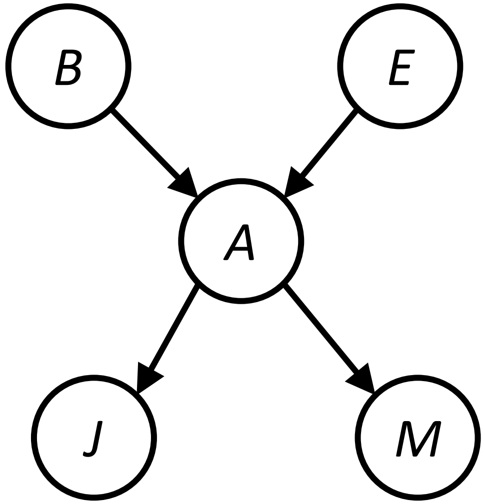
A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

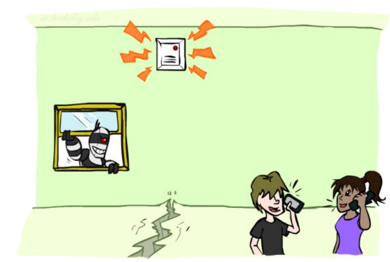
[Demo: BN Applet]

Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

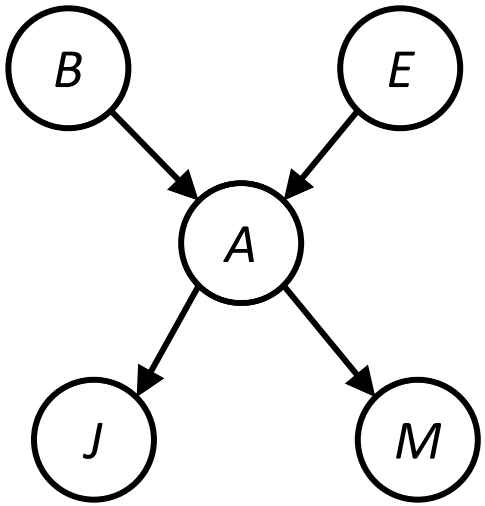
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(+b, -e, +a, -j, +m) =$$

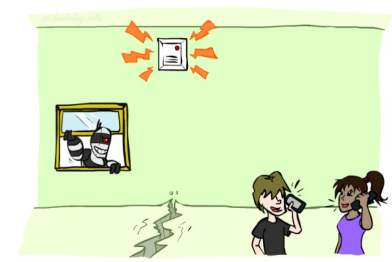
$$P(+b)P(-e)P(+a| + b, -e)P(-j| + a)P(+m| + a) =$$

Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(+b, -e, +a, -j, +m) =$$

$$P(+b)P(-e)P(+a| + b, -e)P(-j| + a)P(+m| + a) =$$

$$0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7$$

Bayes' Nets

- ✓ Representation
- ✓ Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

推理

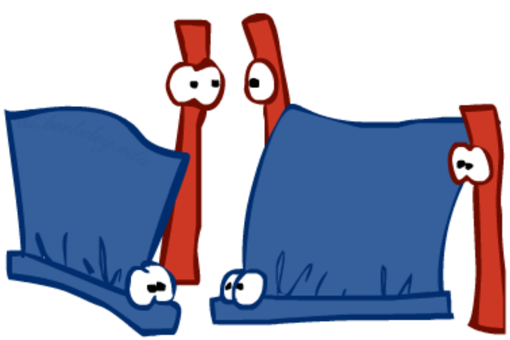
- Inference: calculating some useful quantity from a joint probability distribution

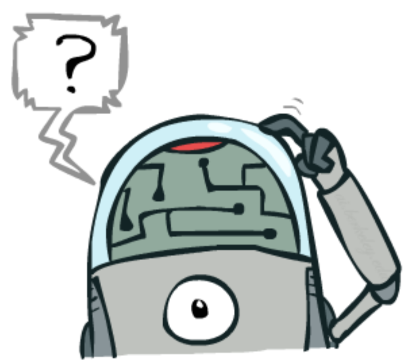
- Examples:
  - Posterior probability

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

  - Most likely explanation:

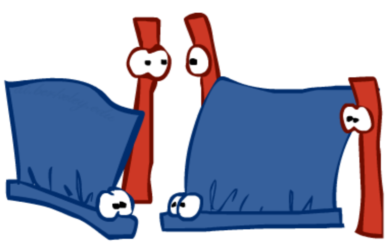
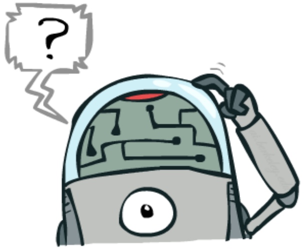
    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$





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)
    - $P(Q|e_1,..,e_k)$
    - 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:
    - $\text{argmax}_{q,r,s}\ P(Q=q,R=r,S=s|e_1,..,e_k)$

推理展望

随机变量 Q, H, E (询问, 隐藏, 证据)

列举法

我们知道如何在一个联合分布上做推理:

$$P(q|e) = \alpha\, P(q, e)$$

$$= \alpha \sum_{h \in \{h_1, h_2\}} P(q, h, e)$$

我们知道贝叶斯网络能够分解联合分布成 CPTs

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

$$= \alpha\, [P(h_1)\, P(q|h_1)\, P(e|q) + P(h_2)\, P(q|h_2)\, P(e|q)]$$

变量消除法

但是我们可以更有效率:

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

$$= \alpha\, P(e|q)\, [P(h_1) P(q|h_1) + P(h_2) P(q|h_2)]$$

$$= \alpha\, P(e|q)\, P(q)$$

现在可以扩展到更大的贝叶斯网络

Inference by Enumeration(列举法)

- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

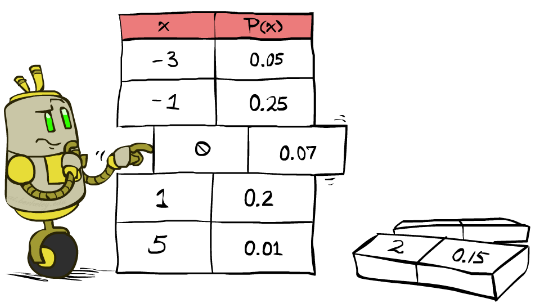
  } $X_1, X_2, \dots X_n$ *All variables*

- We want:

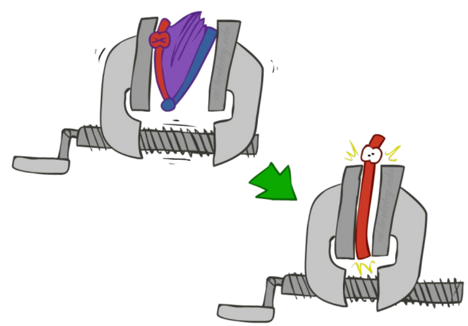
  ** Works fine with multiple query variables, too*

$$P(Q|e_1 \dots e_k)$$

- Step 1: Select the entries consistent with the evidence



- Step 2: Sum out H to get joint of Query and evidence



$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots X_n}$$

- Step 3: Normalize

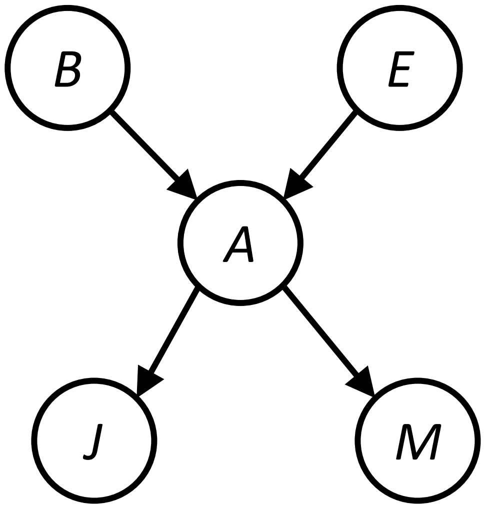
$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \cdots e_k)$$

$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k)$$

用列举法来进行推理

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:



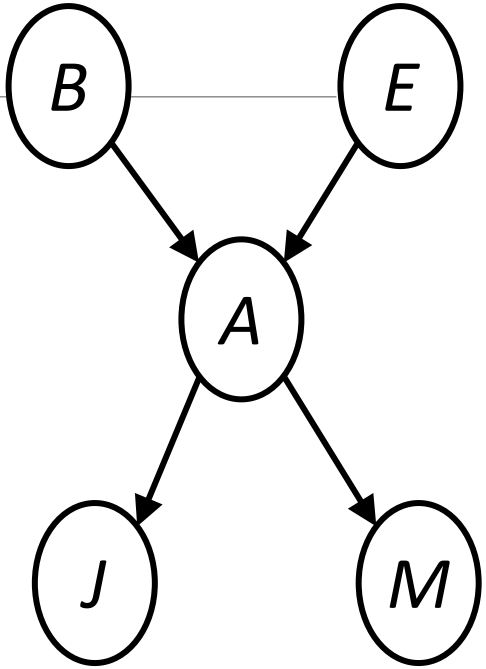
$$P(B \mid +j, +m) \quad \propto_B P(B, +j, +m)$$

$$= \sum_{e,a} P(B, e, a, +j, +m)$$

$$= \sum_{e,a} P(B)P(e)P(a|B, e)P(+j|a)P(+m|a)$$

$$= P(B)P(+e)P(+a|B, +e)P(+j| + a)P(+m| + a) + P(B)P(+e)P(-a|B, +e)P(+j| - a)P(+m| - a)$$
$$P(B)P(-e)P(+a|B, -e)P(+j| + a)P(+m| + a) + P(B)P(-e)P(-a|B, -e)P(+j| - a)P(+m| - a)$$

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

$$P(B \mid j, m) = \alpha P(B, j, m)$$

$$= \alpha \sum_{e,a} P(B, e, a, j, m)$$

$$= \alpha \sum_{e,a} P(B) P(e) P(a|B,e) P(j|a) P(m|a)$$

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?

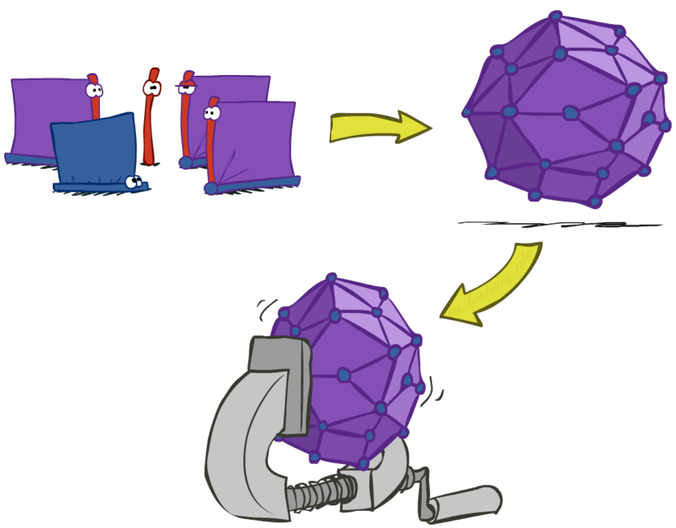
- 比如：
  - $x_1y_1z_1 + x_1y_1z_2 + x_1y_2z_1 + x_1y_2z_2 + x_2y_1z_1 + x_2y_1z_2 + x_2y_2z_1 + x_2y_2z_2$
  - 16 乘法, 7 个加法
  - 许多重复的子表达式!
- 重写成：
  - $(x_1 + x_2)(y_1 + y_2)(z_1 + z_2)$
  - 2 乘法, 3 加法
- $\sum_e \sum_a P(B)\,P(e)\,P(a|B,e)\,P(j|a)\,P(m|a)$

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

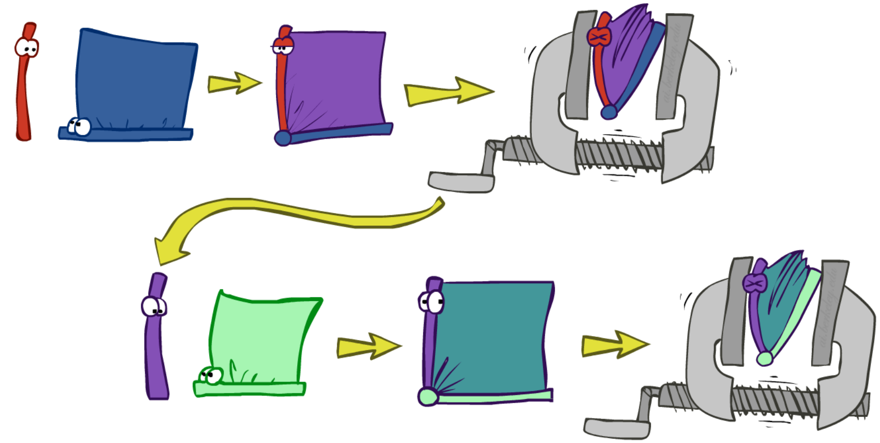
  - 许多重复的子表达式!

Inference by 列举法 vs. 变量消除法

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables



- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration



  - First we'll need some new notation: factors 因子式

变量消除法：基本思想

尽可能早的先做求和操作：

$$P(B|j,m) = \alpha \sum_{e} \sum_{a} P(B,e,a,j,m)$$

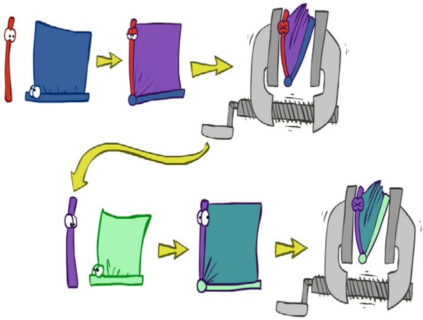
$$= \alpha \sum_{e} \sum_{a} P(j|a)\, P(e)\, P(m|a)\, P(a|B,e)\, P(B)$$

$$= \alpha\, P(B) \sum_{e} P(e) \sum_{a} P(j|a)\, P(m|a)\, P(a|B,e)$$

变量消除法: 基本思想

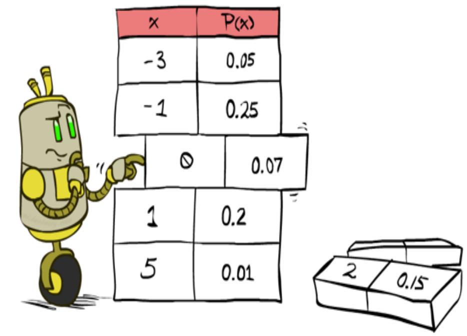
- 尽量把求和操作移到里面，先消掉一些变量

$$P(B \mid j, m) = \alpha \sum_e \sum_a P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$$
$$= \alpha\, P(B) \sum_e P(e) \sum_a P(a|B,e)\, P(j|a)\, P(m|a)$$

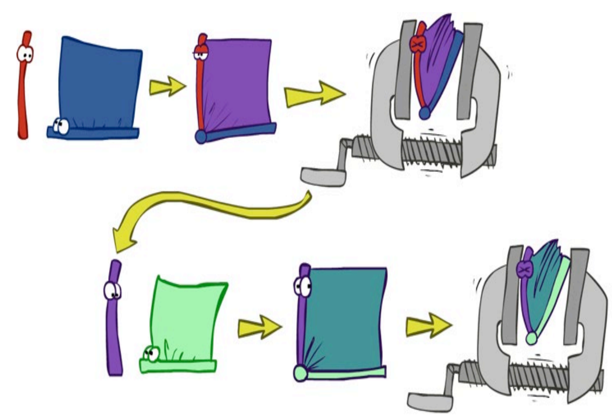


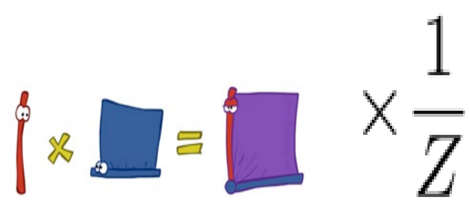
- 计算顺序由里向外
  - 即, 先在 a 上求和, 再在 e 上求和
  - 问题: $P(a|B,e)$ 不是单个数, 一组不同的数，依赖于B 和 e 的值
  - 解决办法：使用不同维度的数组，以及相应的操作；这些列表也叫作 *因子(factors)*

变量消除法

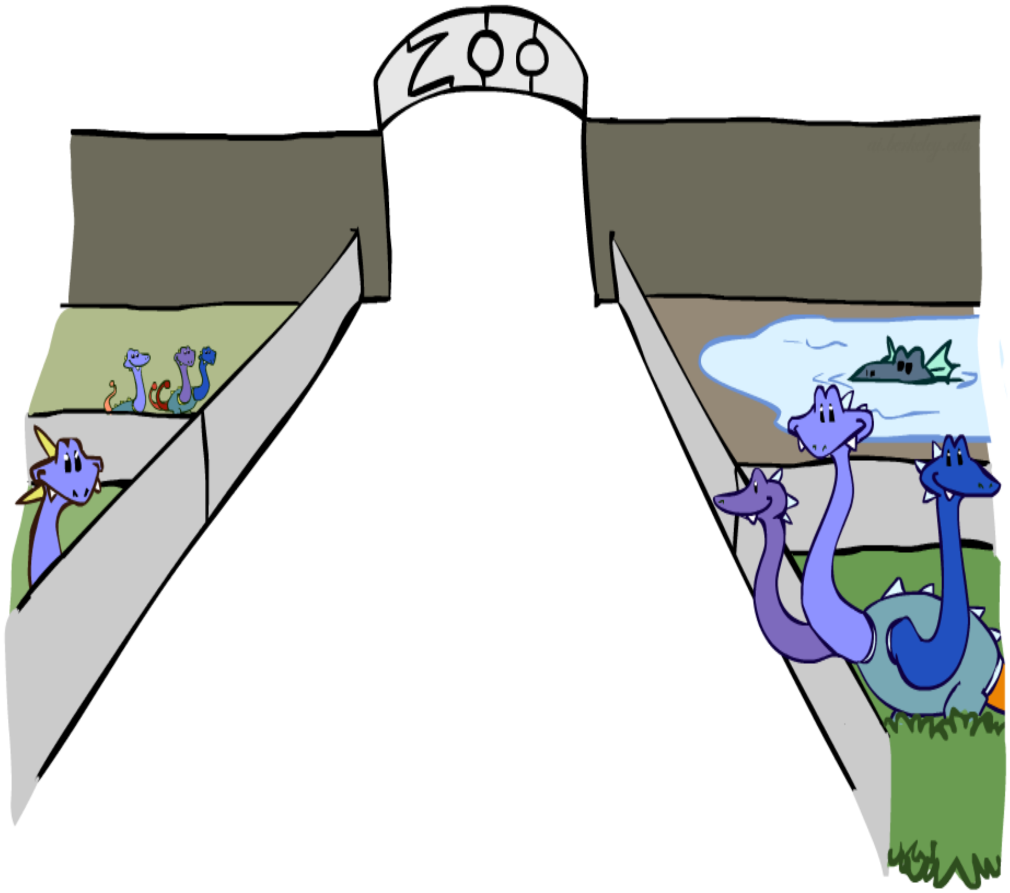


- 查询: $P(Q|E_1=e_1,.., E_k=e_k)$
- 开始于初始的因子表:
  - 局部的条件概率表 (CPTs) (但经过观察变量E的实例化之后)
- 当仍存在隐藏变量时 (既不是 Q 也不是E):
  - 选一个隐含变量 H
  - 合并所有包含 H 的因子表
  - 消除变量 (通过取和) H
- 合并所有剩余因子表，并对结果进行正规化





因子式的相关概念

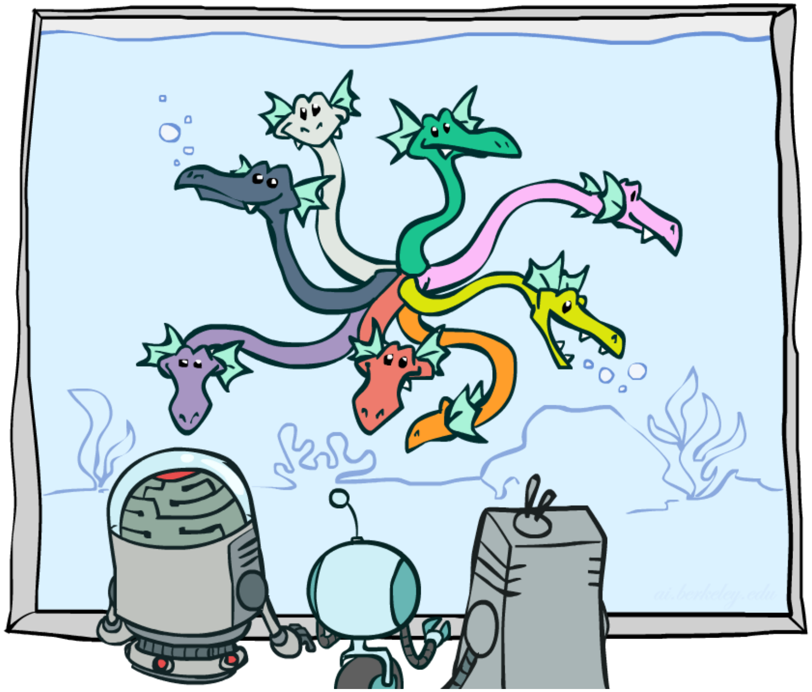


因子式概念I

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1

$P(T, W)$

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



- Selected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for fixed x, all y
  - Sums to P(x)

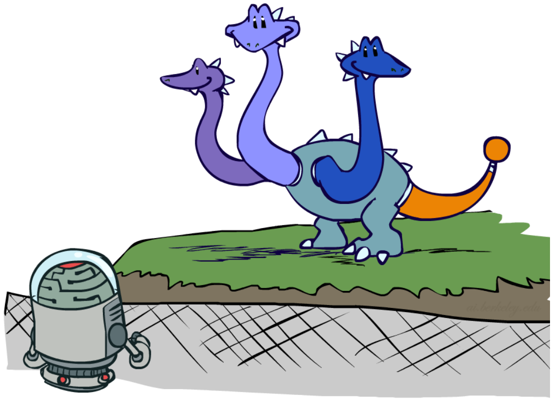
$P(cold, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

- Number of capitals = dimensionality of the table

因子式概念II

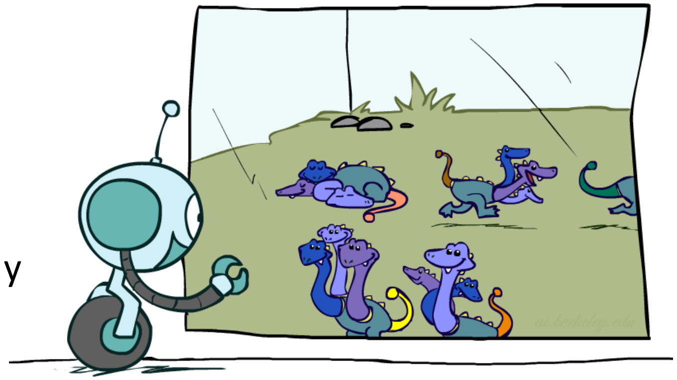
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all
  - Sums to 1



$P(W|cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

- Family of conditionals: P(Y | X)
  - Multiple conditionals
  - Entries P(y | x) for all x, y
  - Sums to |X|



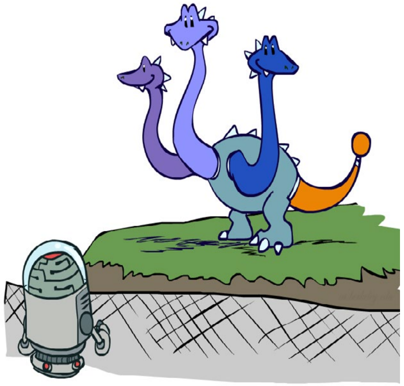
$P(W|T)$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

Rows hot: $P(W|hot)$; rows cold: $P(W|cold)$

因子概念II

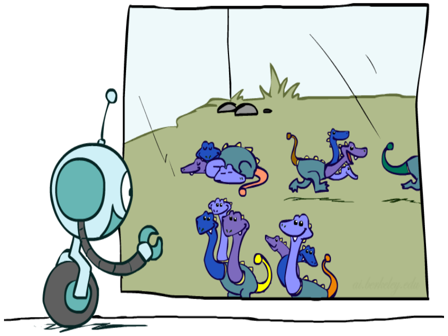
- 单条件概率: P(Y | x)
  - 表项 P(y | x)，对于固定的 x值, 所有的 y值
  - 表项之和为 1



$P(J|a)$

A \ 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 \ J	true	false	
true	0.9	0.1	} - $P(J\|a)$
false	0.05	0.95	} - $P(J\|\neg a)$

因子式概念III

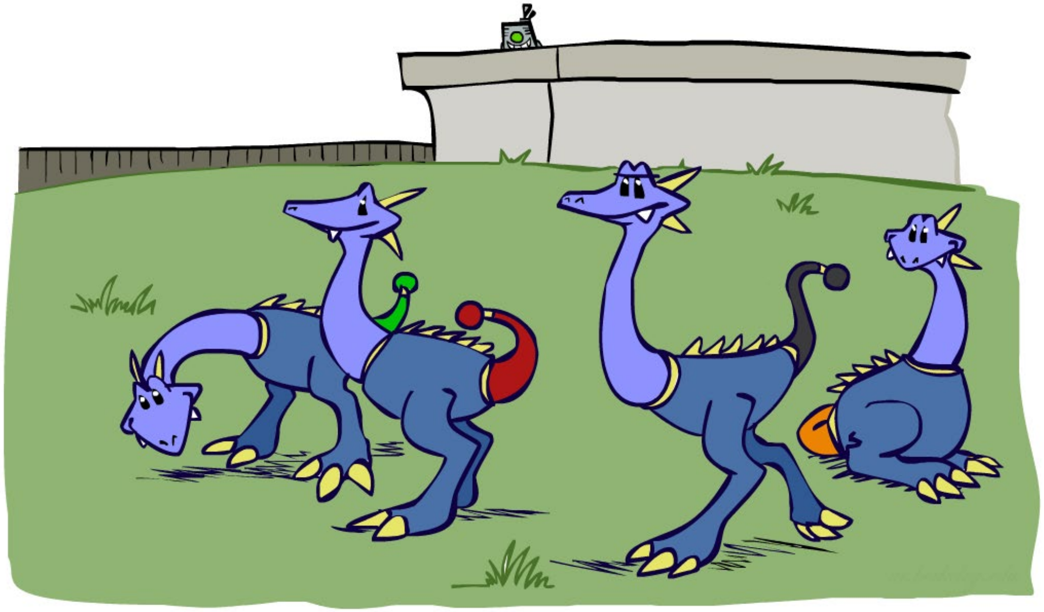
- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$P(rain|T)$

T	W	P
hot	rain	0.2
cold	rain	0.6

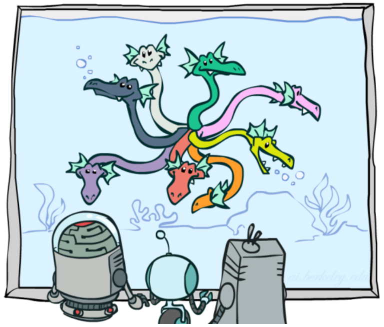
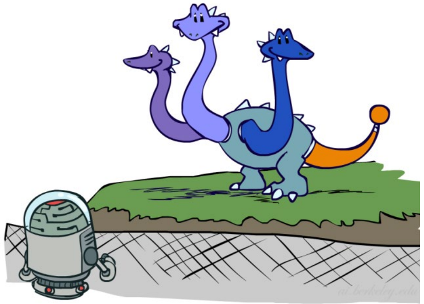
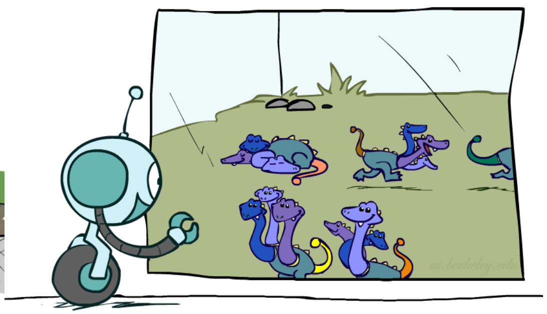
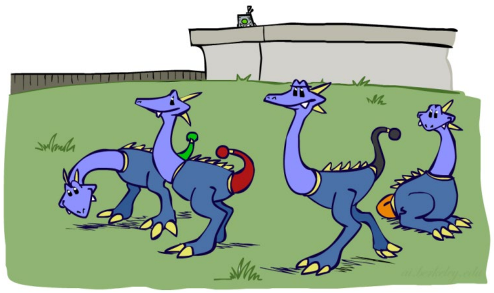
$P(rain|hot)$

$P(rain|cold)$



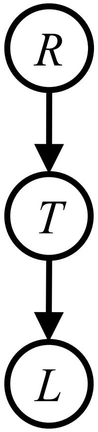
因子式小结

- In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$
  - It is a “因子,” a 多维数组
  - Its values are $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
  - Any assigned (=lower-case) X or Y is a dimension missing (selected) from the array

Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$$P(L) = ?$$

$$= \sum_{r,t} P(r, t, L)$$

$$= \sum_{r,t} P(r)P(t|r)P(L|t)$$

Inference by Enumeration: Procedural Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know $L = +\ell$, the initial factors are

$P(R)$

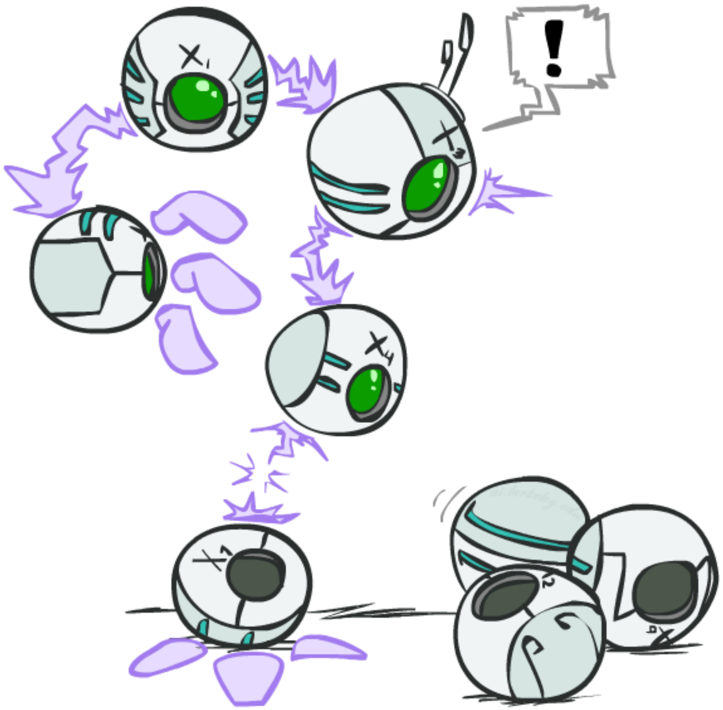
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell|T)$

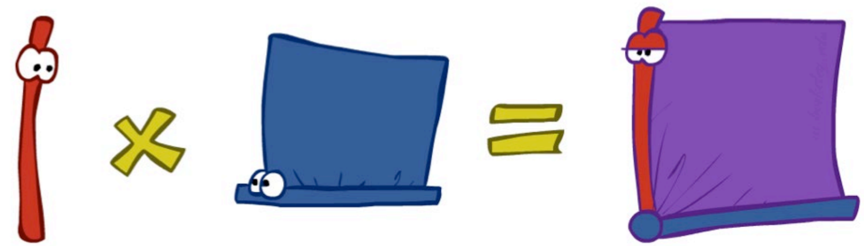
+t	+l	0.3
-t	+l	0.1



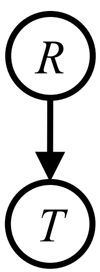
- Procedure: Join all factors, eliminate all hidden variables, normalize

操作 1: 联合因子式

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved



- Example: Join on R



$P(R)$

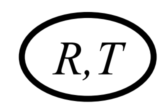
+r	0.1
-r	0.9

$\times$ $P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$\rightarrow$ $P(R,T)$

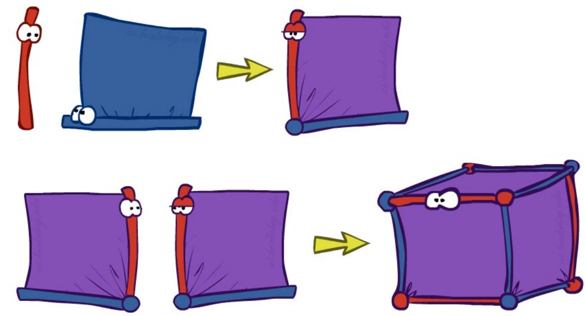
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

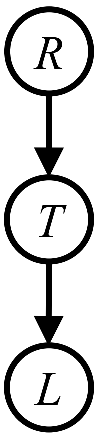


  - Computation for each entry: 逐点乘积

$$\forall r, t : \quad P(r,t) = P(r) \cdot P(t|r)$$

Example: 多次联合操作





$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

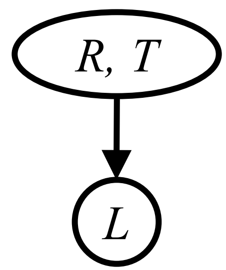
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join T

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

操作 2: 消除变量

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

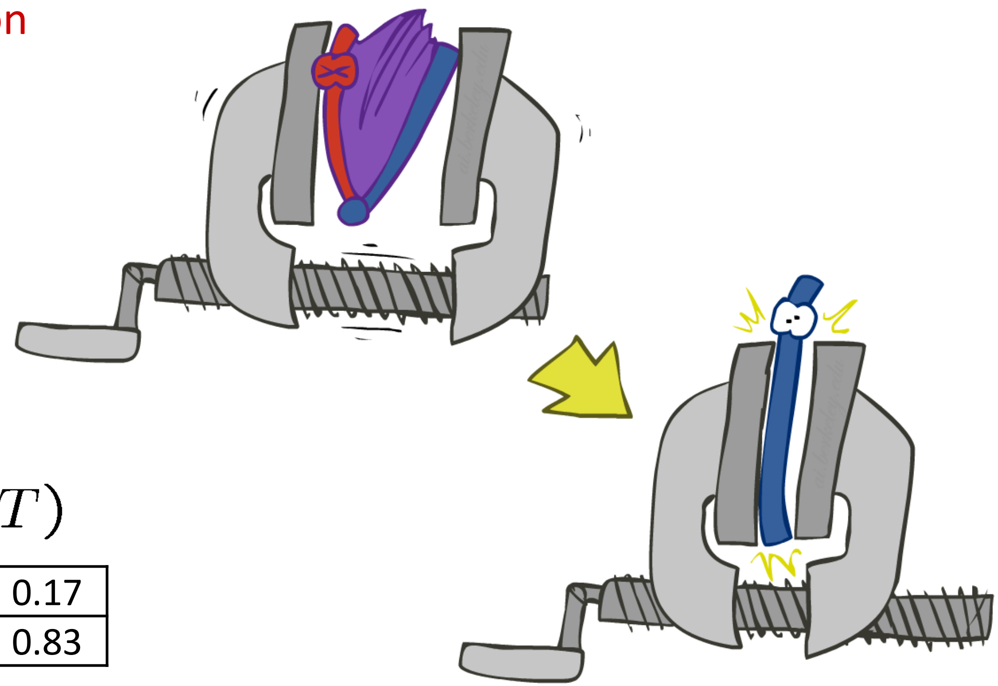
$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum R →

$P(T)$

+t	0.17
-t	0.83



多次消除操作

R, T, L

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Sum out R

T, L

$P(T,L)$

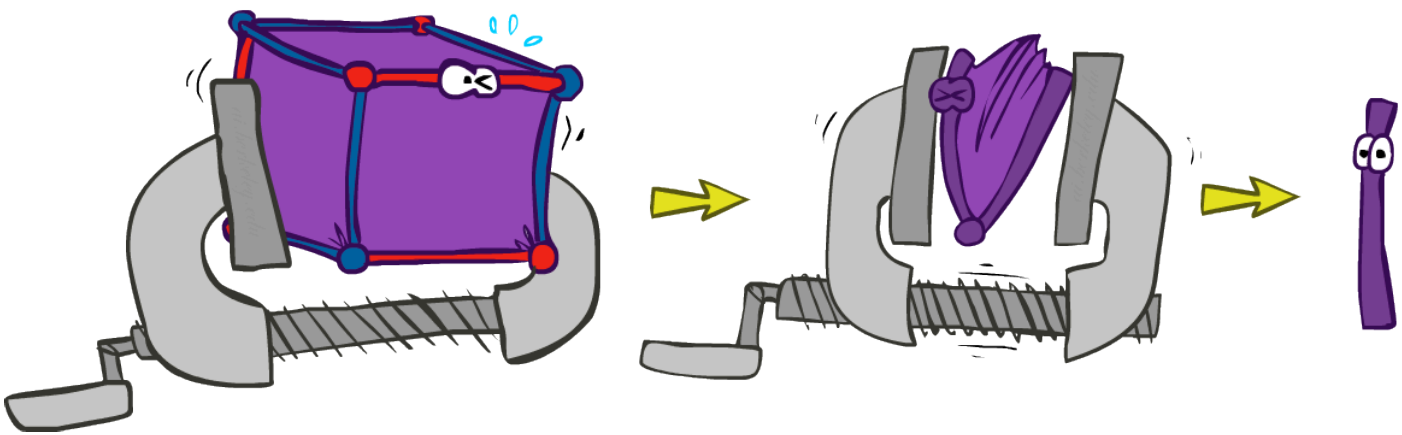
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

L

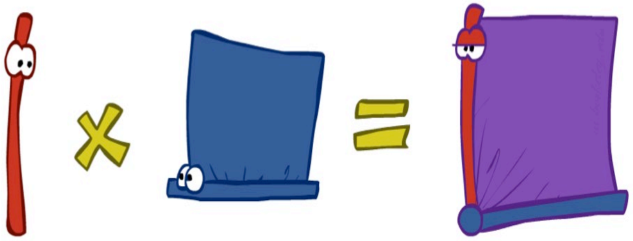
$P(L)$

+l	0.134
-l	0.886



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

$P(A)$

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

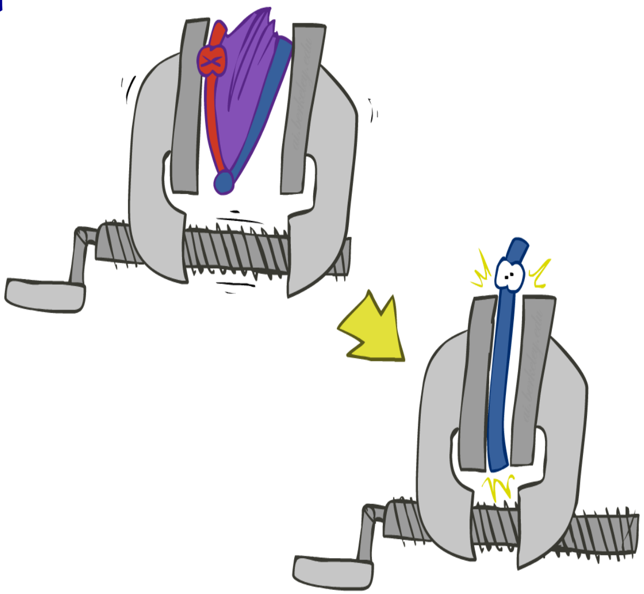
逐点乘积举例

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: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

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



$P(A,J)$

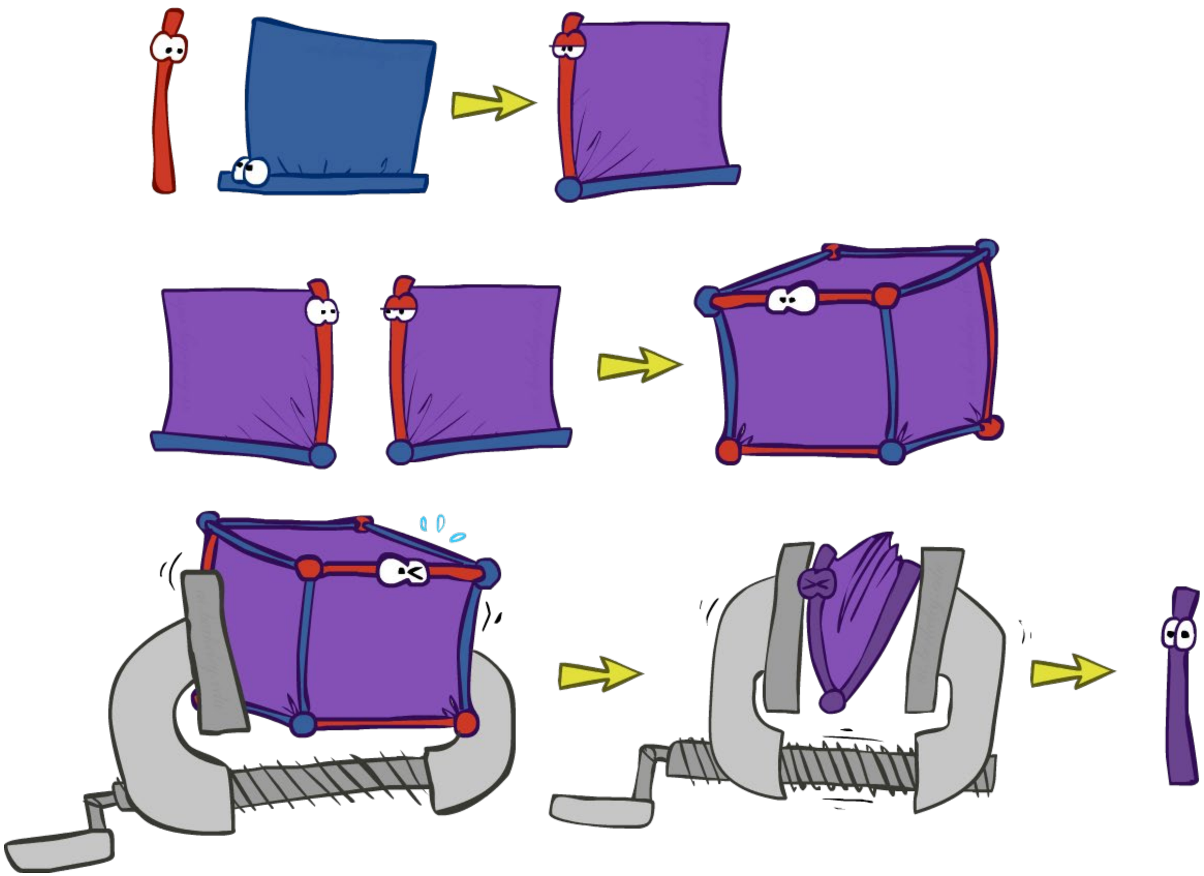
A \ J	true	false
true	0.09	0.01
false	0.045	0.855

加和消掉 J →

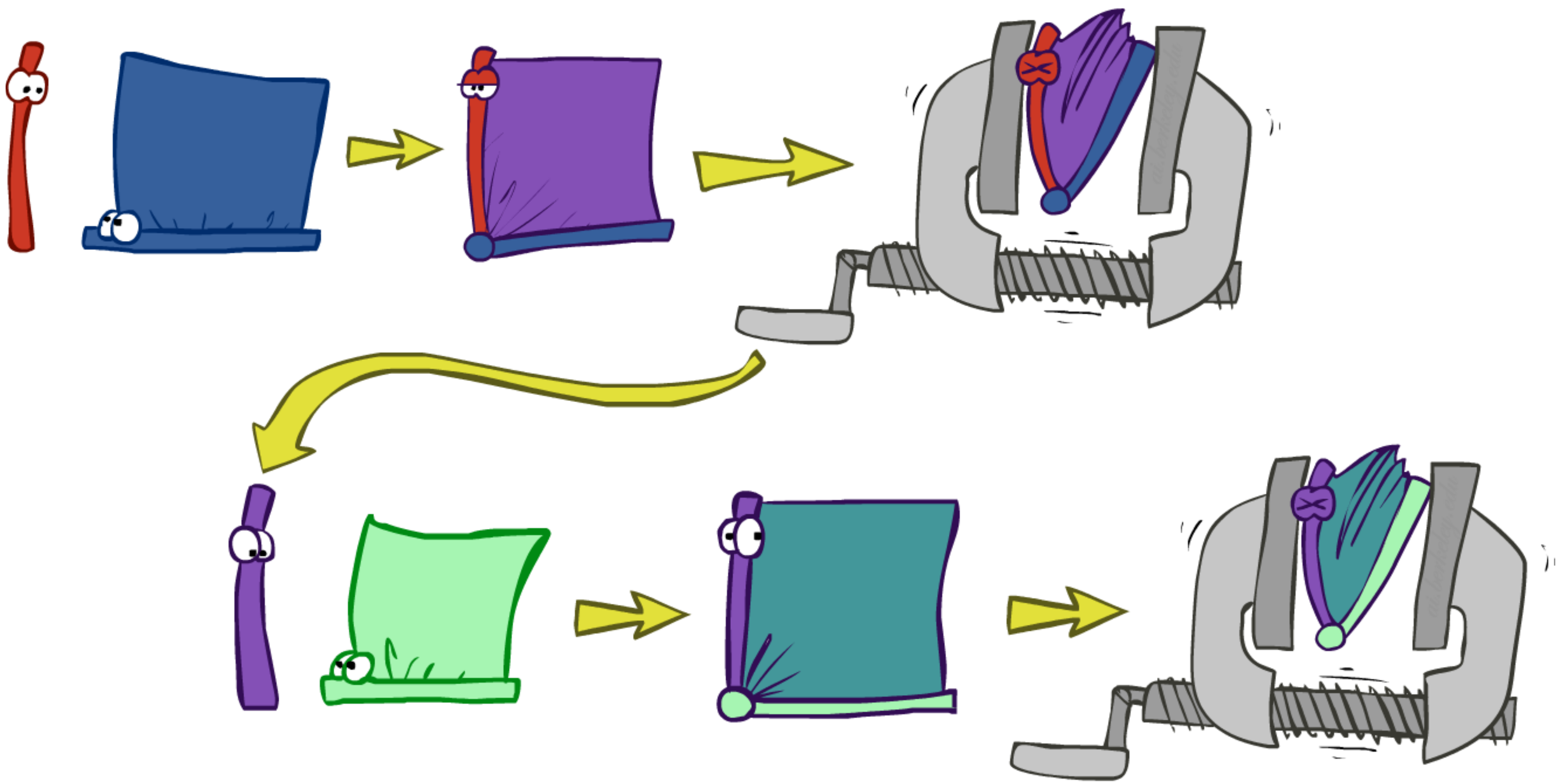
$P(A)$

true	0.1
false	0.9

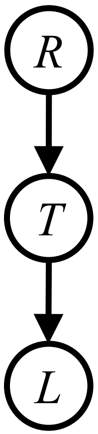
Thus Far: Multiple Join, Multiple Eliminate (= Inference by 列举法)



Marginalizing Early (= 变量消除法)



Traffic Domain



$$P(L) = ?$$

- Inference by Enumeration

$$= \sum_t \sum_r P(L|t)P(r)P(t|r)$$

Join on r

Join on t

Eliminate r

Eliminate t

- Variable Elimination

$$= \sum_t P(L|t) \sum_r P(r)P(t|r)$$

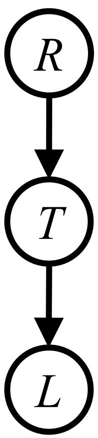
Join on r

Eliminate r

Join on t

Eliminate t

Marginalizing Early! (即变量消除法)



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

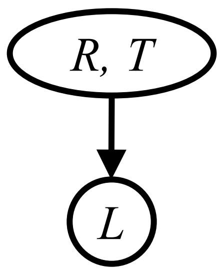
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



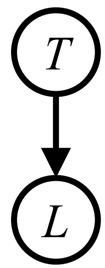
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Sum out R

$P(T)$

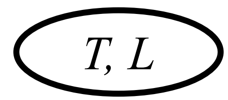
+t	0.17
-t	0.83



$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

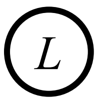
Join T



$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T



$P(L)$

+l	0.134
-l	0.866

举例：取和消除变量A

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

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

Evidence （观察值）

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

  - Computing $P(L|+r)$ the initial factors become:

$P(+r)$

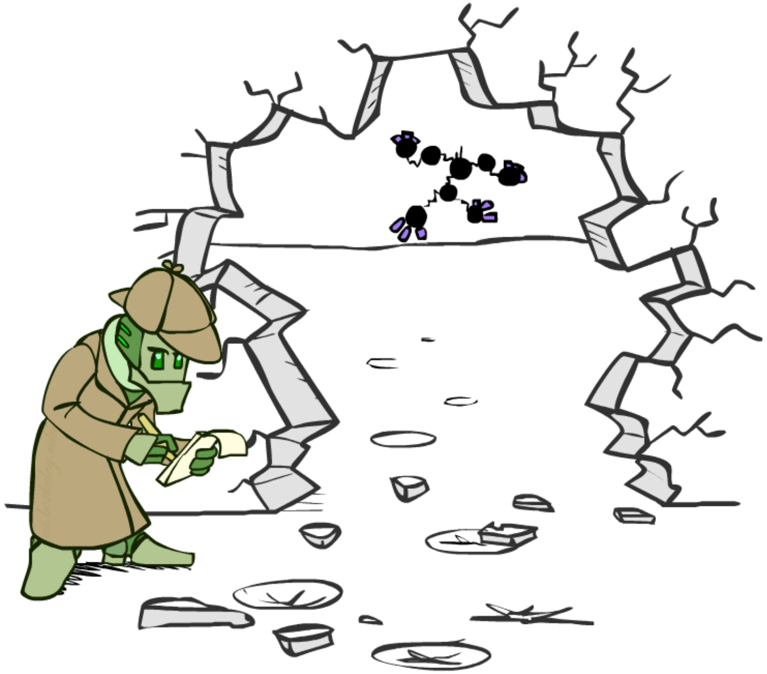
+r	0.1

$P(T|+r)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



- We eliminate all vars other than query + evidence

Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for P(L | +r), we would end up with:

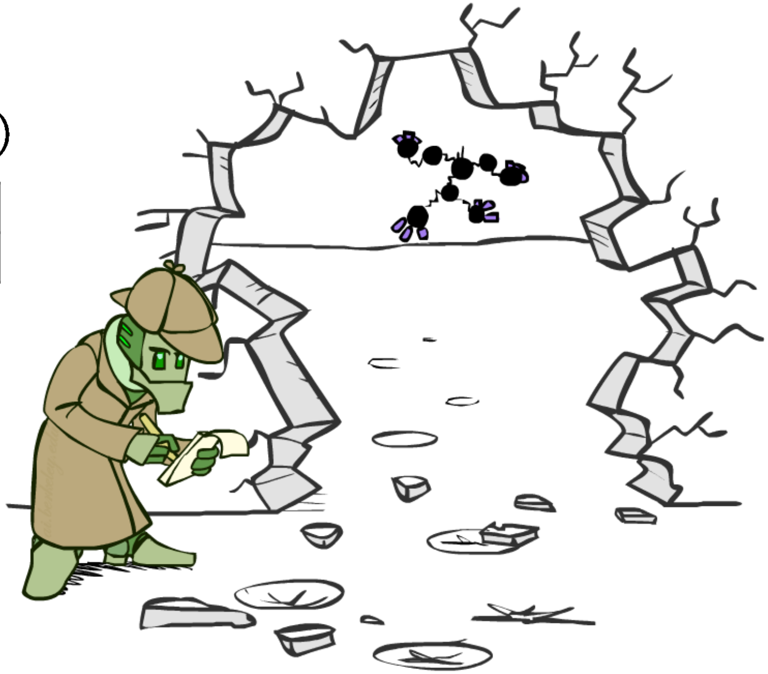
$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize →

$P(L | +r)$

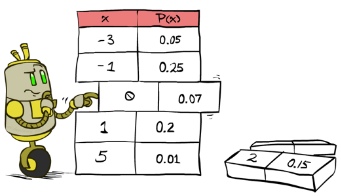
+l	0.26
-l	0.74



- To get our answer, just normalize this!
- That's it!

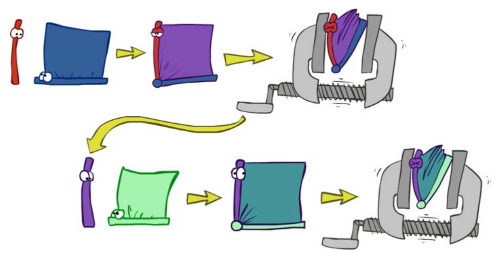
General Variable Elimination （变量消除法一般步骤）

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

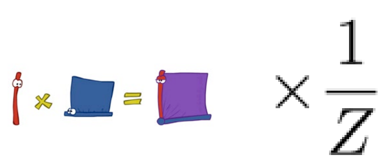


- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

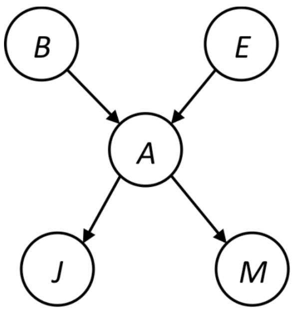


- Join all remaining factors and normalize



举例:之前的防盗报警器网络

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



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

Choose A

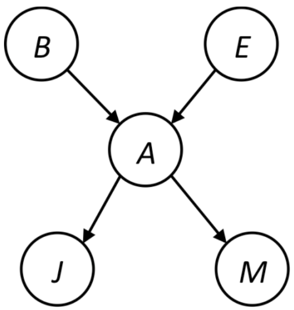
$P(A|B,E)$
$P(j|A)$
$P(m|A)$

$\xrightarrow{\times}$ $P(j,m,A|B,E)$ $\xrightarrow{\Sigma}$ $P(j,m|B,E)$

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

Example

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



Choose E

$P(E)$

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

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

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

Finish with B

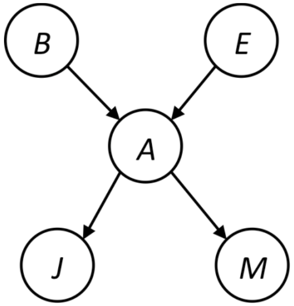
$P(B)$

$P(j, m|B)$

$\times$ → $P(j, m, B)$ Normalize → $P(B|j, m)$

同样的例子用数学表达式表示这个过程

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



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

$$\begin{aligned} P(B|j,m) &\propto P(B,j,m) \\ &= \sum_{e,a} P(B,j,m,e,a) \\ &= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)f_1(B,e,j,m) \\ &= P(B)\sum_{e} P(e)f_1(B,e,j,m) \\ &= P(B)f_2(B,j,m) \end{aligned}$$

marginal obtained from joint by summing out

use Bayes' net joint distribution expression

use x*(y+z) = xy + xz

joining on a, and then summing out gives f_1

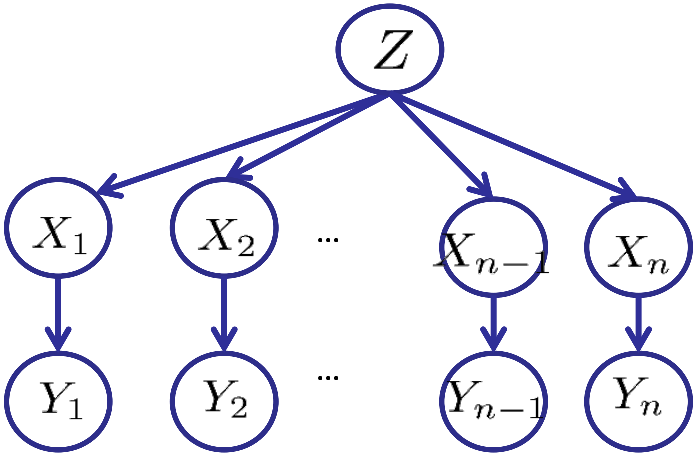
use x*(y+z) = xy + xz

joining on e, and then summing out gives f_2

All we are doing is exploiting uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz = (u+v)(w+x)(y+z) to improve computational efficiency!

选择变量进行消除的顺序

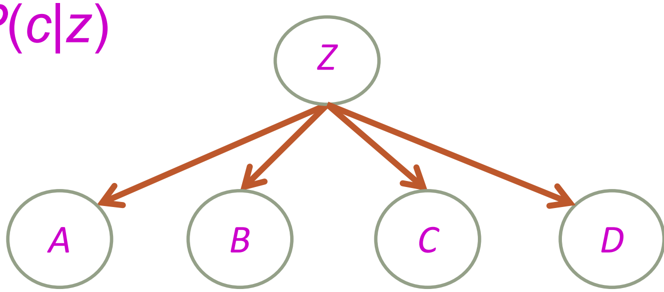
- For the query $P(X_n | y_1,\ldots,y_n)$ work through the following two different orderings as done in previous slide: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?



- Answer: 2^{n+1} versus 2^2 (assuming binary)
- In general: the ordering can greatly affect efficiency.

选择变量的顺序有关系

- 如果排序为 D, Z, A, B C
  - $P(D) = \alpha \sum_{z,a,b,c} P(D|z)\, P(z)\, P(a|z)\, P(b|z)\, P(c|z)$
  - $\quad\quad = \alpha \sum_{z} P(D|z)\, P(z) \sum_{a} P(a|z) \sum_{b} P(b|z) \sum_{c} P(c|z)$
  - 最大的因子（乘积后）有 2 个变量 (D,Z)



- 如果排序为 A, B C, D, Z
  - $P(D) = \alpha \sum_{a,b,c,z} P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - $\quad\quad = \alpha \sum_{a} \sum_{b} \sum_{c} \sum_{z} P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - 最大的因子（乘积后）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

- 时间和空间复杂度决定于 the largest factor(最大的因子表)

- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example 2^{n+1} vs. 2^2

- Does there always exist an ordering that only results in small factors?
  - No!

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!

Worst Case Complexity?

- CSP:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

$$P(X_i = 0) = P(X_i = 1) = 0.5$$

$$Y_1 = X_1 \vee X_2 \vee \neg X_3$$

$$\ldots$$

$$Y_8 = \neg X_5 \vee X_6 \vee X_7$$

$$Y_{1,2} = Y_1 \wedge Y_2$$

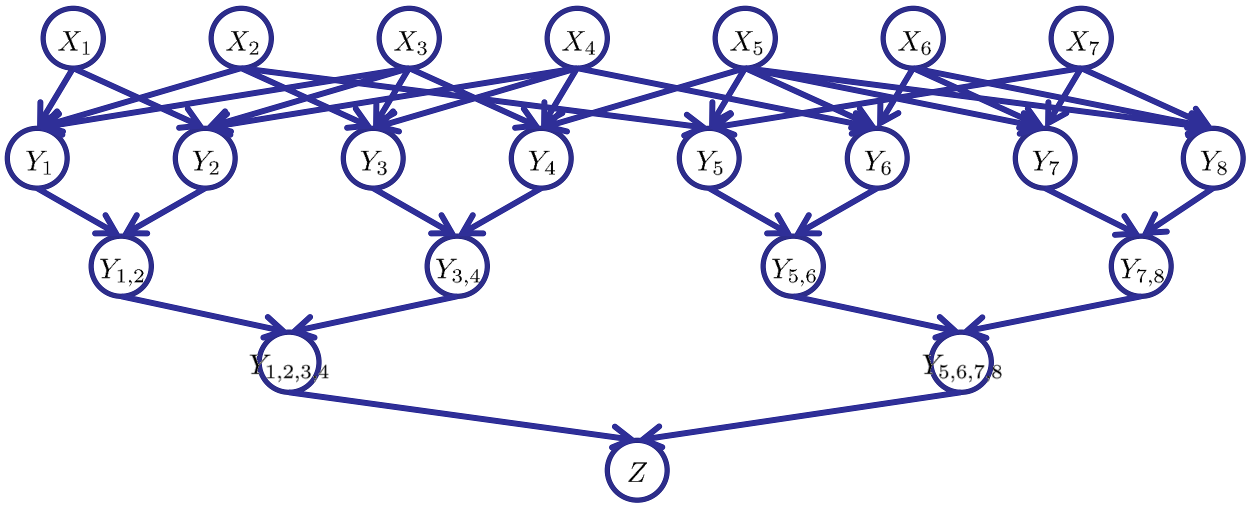
$$\ldots$$

$$Y_{7,8} = Y_7 \wedge Y_8$$

$$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$$

$$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$$

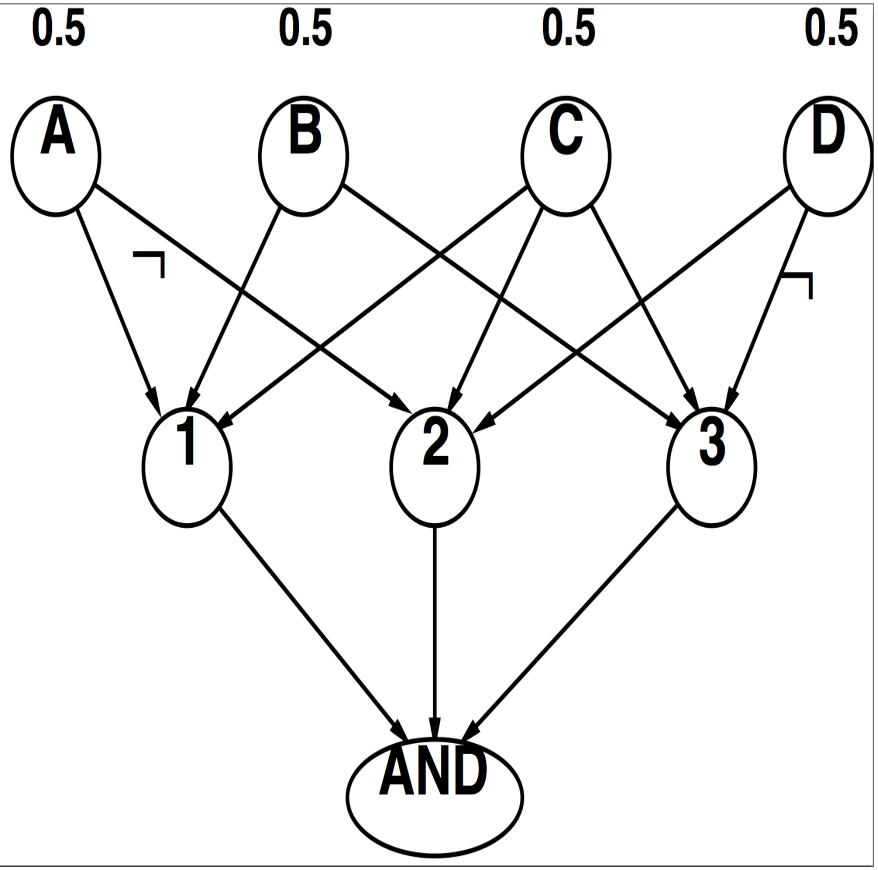
$$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$$



- If we can answer P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution.

- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

最差情况复杂度? 从 SAT 问题约简过来



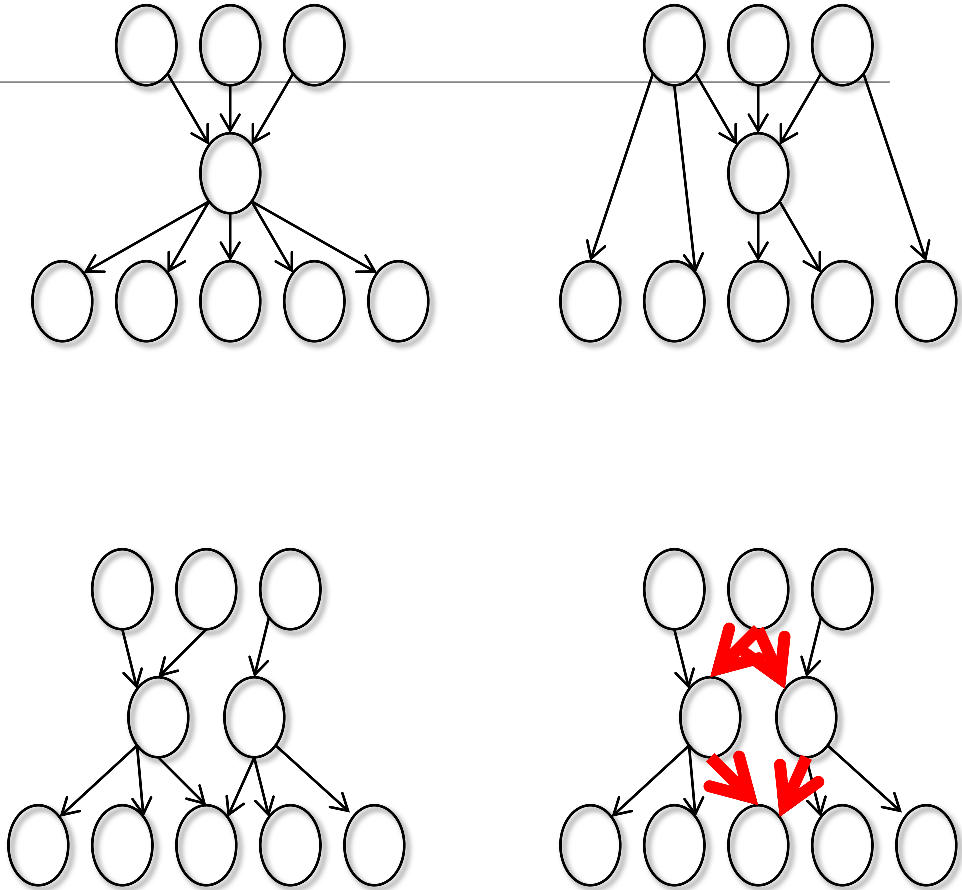
- 合取范式(CNF)的子句:
  - A v B v C
  - C v D v ¬A
  - B v C v ¬D
- *P(AND)* > 0 当且仅当 所有子句是可满足的
  - => NP-难度
- P(AND) = S x 0.5^n , S 是使该合取范式满足的变量赋值的组和数
  - => #P-难度 （至少和其对应的NP-难度问题一样难，或更难的）

最差情况复杂度？从 SAT 问题约简过来

- 如果我们能够回答P(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，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