基于贝叶斯网络进行概率推理

1. 贝叶斯网络（何谓贝叶斯网络；从网络计算概率；如何构建贝叶斯网络；网络中的条件独立性）
2. 条件概率的有效表示（确定性结点；非确定性结点；连续变量）
3. 贝叶斯网络的精确推理（枚举；消去重复计算）
4. 贝叶斯网络的近似推理（直接采样；拒绝采样；加权；马尔科夫链采样）

祝恩

review: 天气不影响牙病

- 可以说天气并不受牙病变量影响。所以，下面的断言似乎是合理的：

  $P(cloudy \mid toothache, catch, cavity) = P(cloudy)$

- 据此，可以推断：

  $P(toothache, catch, cavity, cloudy) = P(cloudy)\, P(toothache, catch, cavity)$

- 实际上，可以写出通用公式：

  $\mathbf{P}(Toothache, Catch, Cavity, Weather) = \mathbf{P}(Toothache, Catch, Cavity)\, \mathbf{P}(Weather)$

review: 何谓独立性？

- “天气是独立于牙病问题的”这样的特性称为**独立性**，也称为**边缘独立性**或**绝对独立性**
- 两个命题a和b之间独立可以写作：

  $P(a|b) = P(a)$ or $P(b|a) = P(b)$
  or $P(a \wedge b) = P(a)P(b)$
- 变量X和Y之间独立可以写作：

  $\mathbf{P}(X|Y) = \mathbf{P}(X)$ or $\mathbf{P}(Y|X) = \mathbf{P}(Y)$ or
  $\mathbf{P}(X \wedge Y) = \mathbf{P}(X)\mathbf{P}(Y)$
- 独立性断言通常是基于问题的领域知识的。

review: 条件独立性

- 条件独立性。例如：
- **P**(*Toothache*, *Cavity*, *Cavity*)

= **P**(*Toothache*, *Catch* | *Cavity*) **P**(*Cavity*)
（乘法原则）

= **P**(*Toothache* | *Cavity*) **P**(*Catch* | *Cavity*) **P**(*Cavity*) （条件独立性）

- 按照这种方式，原来较大的概率表被分解成为三个较小的概率表

review: 条件独立性

- 给定第三个随机变量Z后，两个随机变量X和Y的条件独立性的一般定义是：
- $\mathbf{P}(X, Y \mid Z) = \mathbf{P}(X \mid Z)\ \mathbf{P}(Y \mid Z)$
- 也可以使用以下条件独立性的等价形式：
- $\mathbf{P}(X \mid Y, Z) = \mathbf{P}(X \mid Z)$ 和 $\mathbf{P}(Y \mid X, Z) = \mathbf{P}(Y \mid Z)$

review:

- 完全联合概率分布随着变量数目的增多会增大到不可操作的程度。

review:

- 完全联合概率分布随着变量数目的增多会增大到不可操作的程度。
- 变量之间的独立性和条件独立关系可以大大减少为定义完全联合概率分布所需指定的概率数目。

贝叶斯网络

Judea Pearl

Judea Pearl

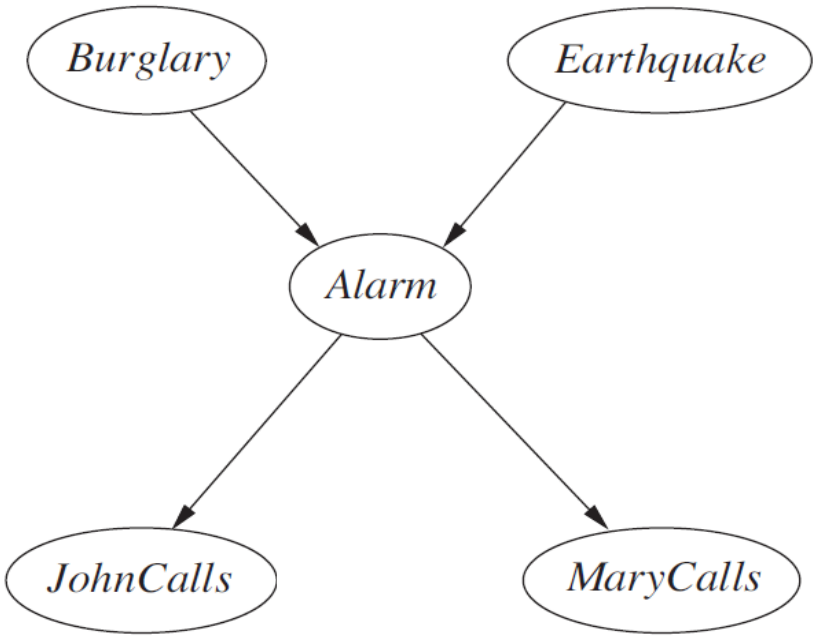
- 1985年提出贝叶斯网络的构想
- 1986年发表论文《fusion，propagation，and structuring in belief networks》，标志着数据能够进行因果推断
- 1988年发表相关著作标志着贝叶斯网络成为一种独立的研究数据因果关系的方法
- 著作《causality：models，reasoning，and inference》创立了因果推理演算法，为他赢得了2011英国伦敦经济和政治科学学院的lakatos奖，评语说他“为科学哲学做出了重大的杰出贡献”
- 2011年获得图灵奖

贝叶斯网络用于什么？

- **贝叶斯网络**用于表示变量之间的依赖关系。可以本质上表示任何完全联合概率分布，在许多情况下这种表示是简明扼要的。

防盗报警器问题

- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?
- 变量之间的依赖关系（因果关系）可表示成网络



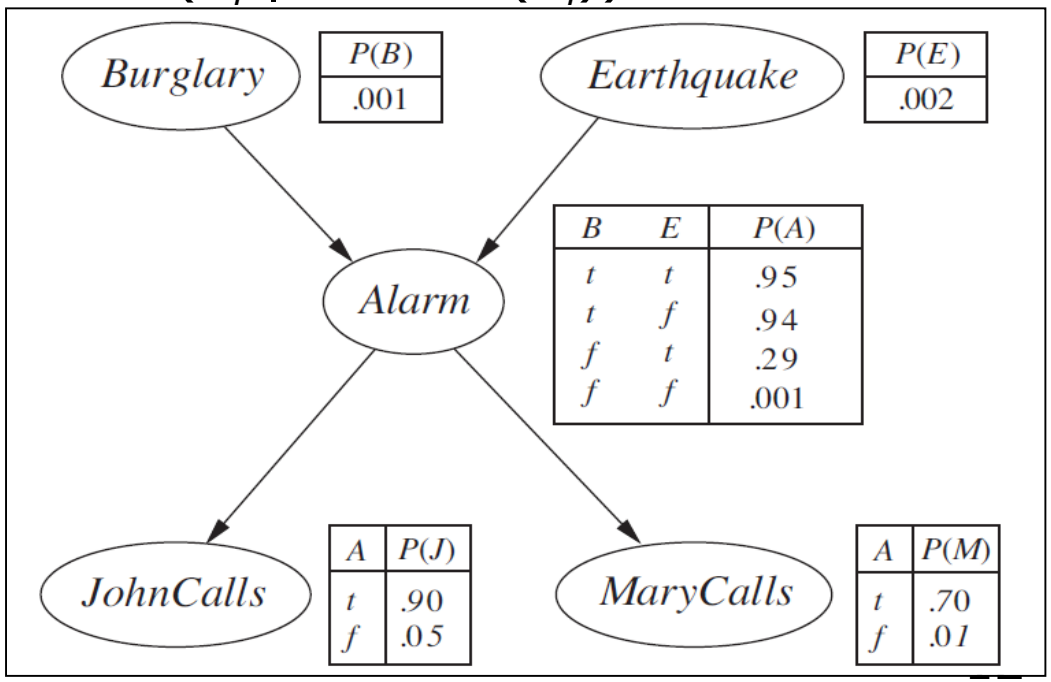
贝叶斯网络是什么？

1. 每个结点对应一个随机变量，这个变量可以是离散的或者连续的。
2. 一组有向边或箭头连接结点对。如果有从结点X指向结点Y的箭头，则称X是Y的一个*父结点*。图中没有有向回路（因此被称为有向无环图，或简写为DAG）。
3. 每个结点X_i有一个条件概率分布$\mathbf{P}(X_i \mid Parents(X_i))$，量化其父结点对该结点的影响。

- 由此可确定所有变量的完全联合概率分布

防盗报警器问题的贝叶斯网络

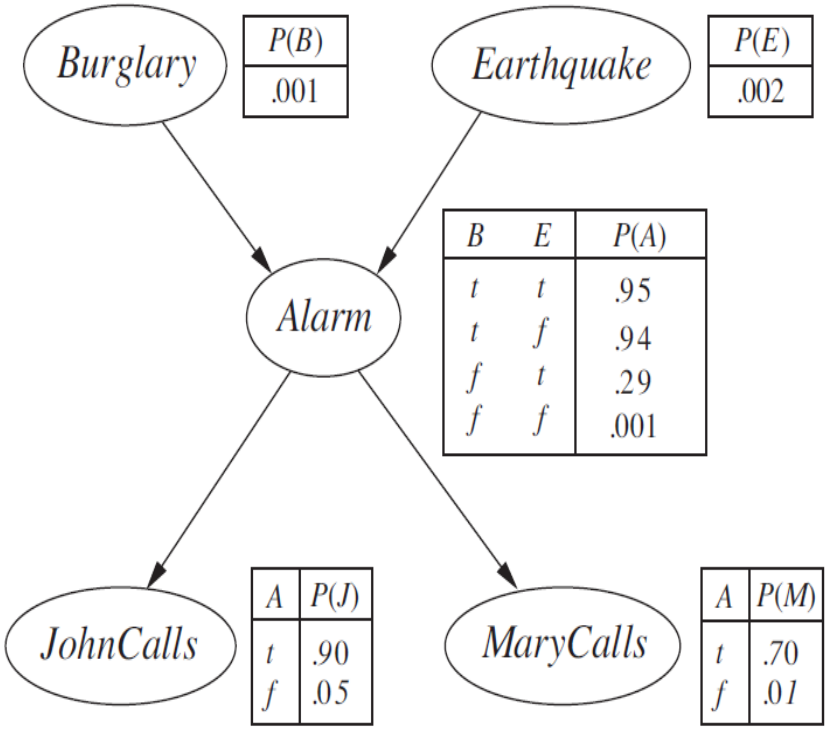
字母*B*、*E*、*A*、*J*、*M*分别表示*Burglary*、*Earthquake*、*Alarm*、*JohnCalls*、*MaryCalls*
每个变量右边是条件概率表CPT



从贝叶斯网络计算联合概率分布

- $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$= ?$



从贝叶斯网络计算联合概率分布

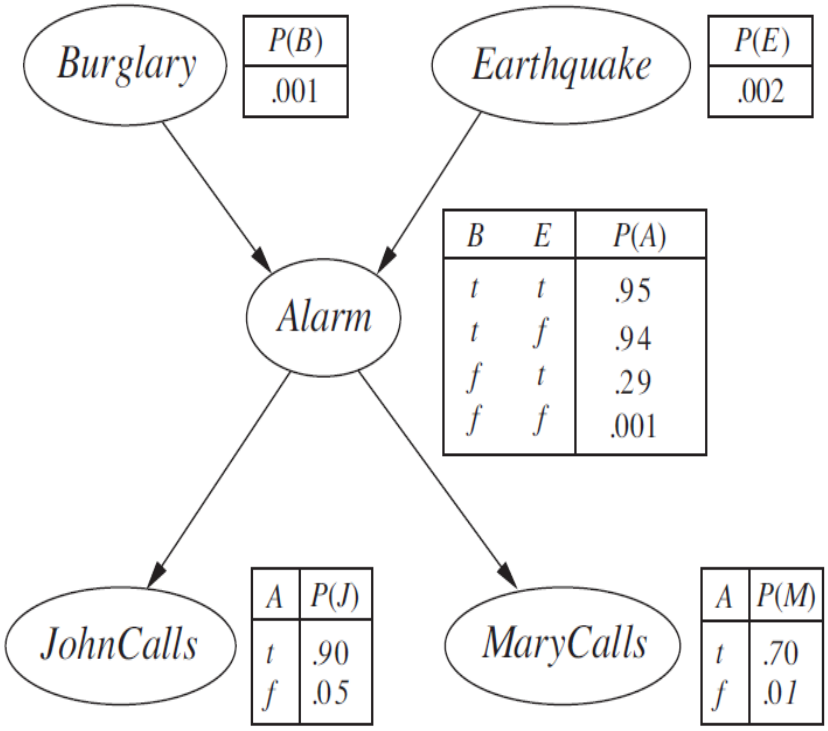
- $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

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

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

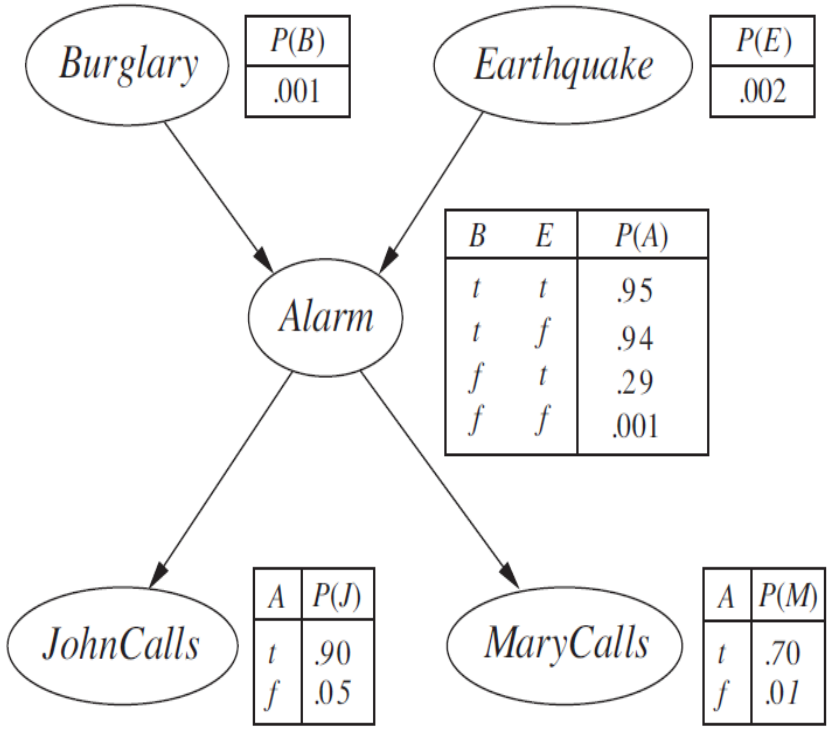
$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$

$= 0.00062$



从贝叶斯网络计算联合概率分布

- $P(a \wedge \neg b \wedge \neg e \wedge j \wedge m)$ =?



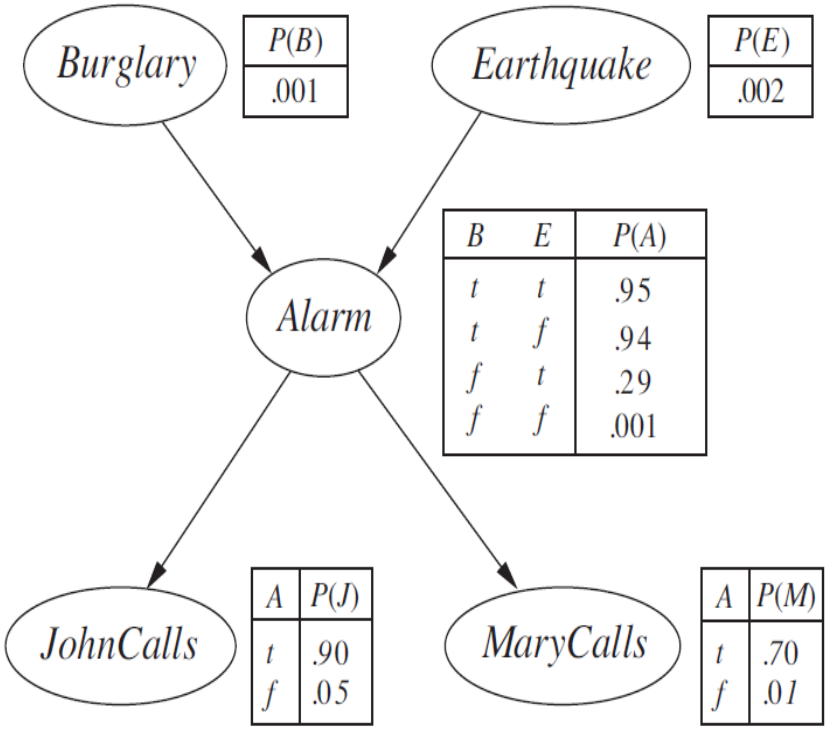
问题出在哪里？

- $P(a \wedge \neg b \wedge \neg e \wedge j \wedge m)$

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

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

$= 0.001 \times 0.999 \times 0.998 \times ? \times ?$



结点的排序

- $P(x_1, \ldots, x_n) =$
  $P(x_n \mid x_{n-1}, \ldots, x_1)\ P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \ldots P(x_2 \mid x_1)\ P(x_1)$

- 或写为：
- $P(x_n, \ldots, x_1) =$
  $P(x_n \mid x_{n-1}, \ldots, x_1)\ P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \ldots P(x_2 \mid x_1)\ P(x_1)$

- 若$Parents(X_i) \subseteq \{X_{i-1}, \ldots, X_1\}$，即序列$x_1, \ldots, x_n$中$X_i$的父节点都排在它的前面，则$\mathbf{P}(X_i \mid X_{i-1}, \ldots, X_1) = \mathbf{P}(X_i \mid Parents(X_i))$

- 从而

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

从贝叶斯网络计算联合概率分布

- 联合概率分布中的每个条目都可表示为贝叶斯网络的条件概率表（CPT）中适当元素的乘积

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

- 其中$parents(X_i)$表示$Parents(X_i)$的变量的出现在x_1, …, x_n中的取值。

如何构建贝叶斯网络

构建贝叶斯网络

1. 结点：首先确定变量集合。对变量进行排序得到$\{X_1,\ldots,X_n\}$，原因排列在结果之前。

2. 边： i从1到n，执行：

- 从$X_1,\ldots,.X_{i-1}$中选择X_i的父结点的最小集合，使得$Parents(X_i) \subseteq \{X_{i-1}, \ldots, X_1\}$满足。
- 在每个父结点与X_i之间插入一条边。
- 条件概率表（CPTs）：写出条件概率表$\mathbf{P}(X_i|Parents(X_i))$。

贝叶斯网络的紧致性

- 假设n个布尔变量，联合概率分布中将包含____个数值。若n=30，将有超过___个值。

- 假设每个随机变量受到至多k个其他随机变量的影响，其中k是某个常数。整个网络可以用至多_____个数值描述。若n=30，k=5，则只有____个值。

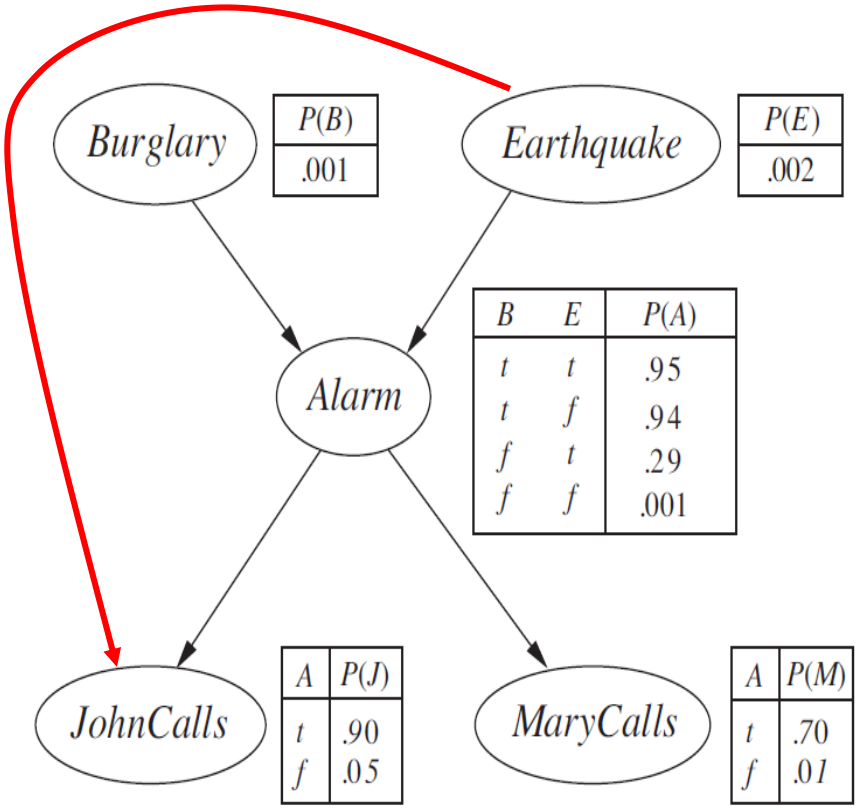
贝叶斯网络的紧致性

- 假设n个布尔变量，联合概率分布中将包含2^n个数值。若n=30，将有超过10亿个值。

- 假设每个随机变量受到至多k个其他随机变量的影响，其中k是某个常数。整个网络可以用至多$n2^k$个数值描述。若n=30，k=5，则只有960个值。

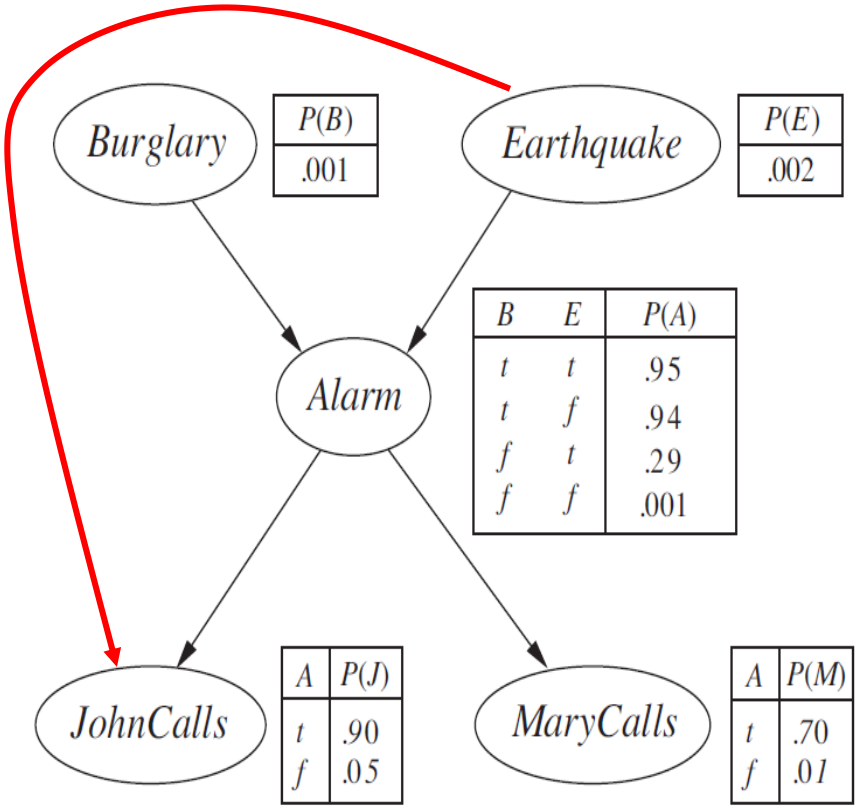
微弱的依赖关系

- 地震发生的时候，John和Mary即使听到了警报声也不会打电话，因为他们认为警报声是地震引起的而不是盗贼闯入引起的。是否需要增加从*Earthquake*到*JohnCalls*以及*MaryCalls*的边?



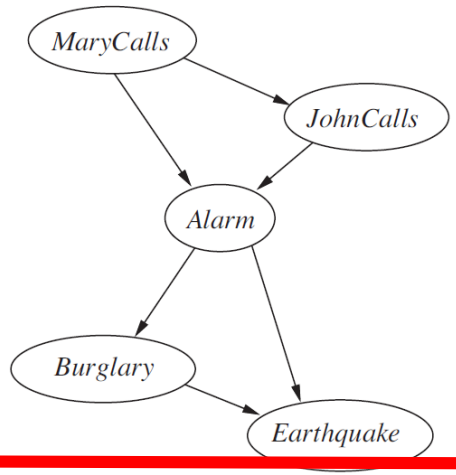
微弱的依赖关系可以去掉

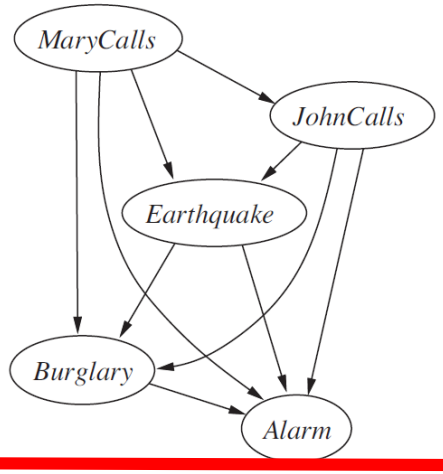
- 地震发生的时候，John和Mary即使听到了警报声也不会打电话，因为他们认为警报声是地震引起的而不是盗贼闯入引起的。是否需要增加从*Earthquake*到*JohnCalls*以及*MaryCalls*的边（条件概率表也会扩大）取决于更高精度概率的重要性与指定额外信息的代价之间的对比。
- 不值得为了一点点精度的提高而增加网络的复杂度。

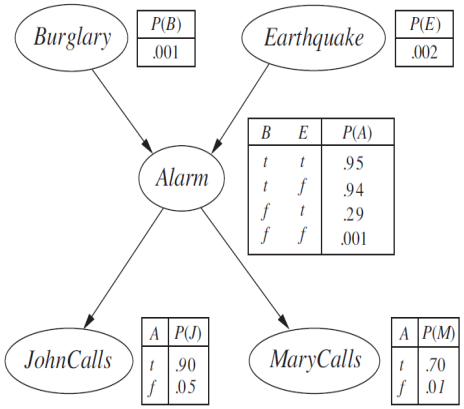


结点顺序不当会如何？

- 若“结果”作为“原因”的父节点



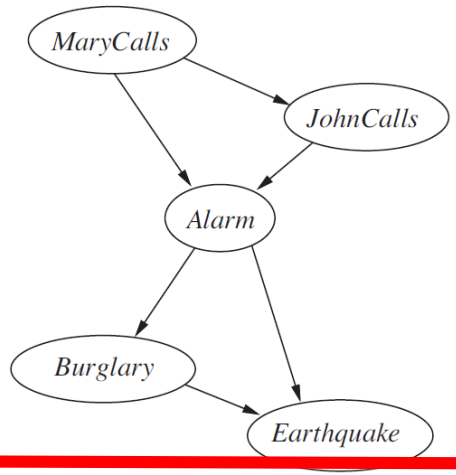


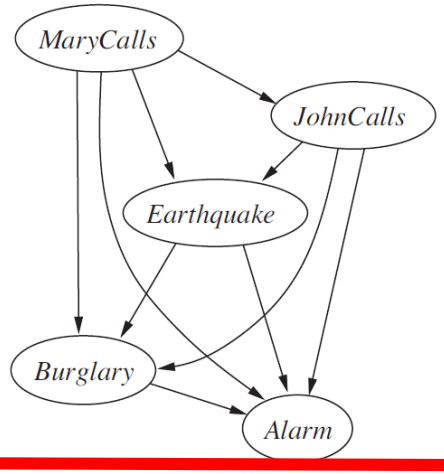


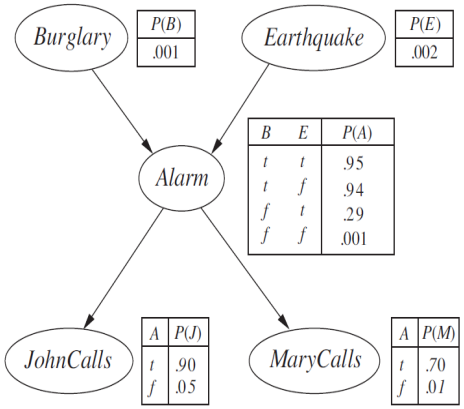
- 会有什么问题？P(原因|结果)

结点顺序不当会如何？

- 若“结果”作为“原因”的父节点

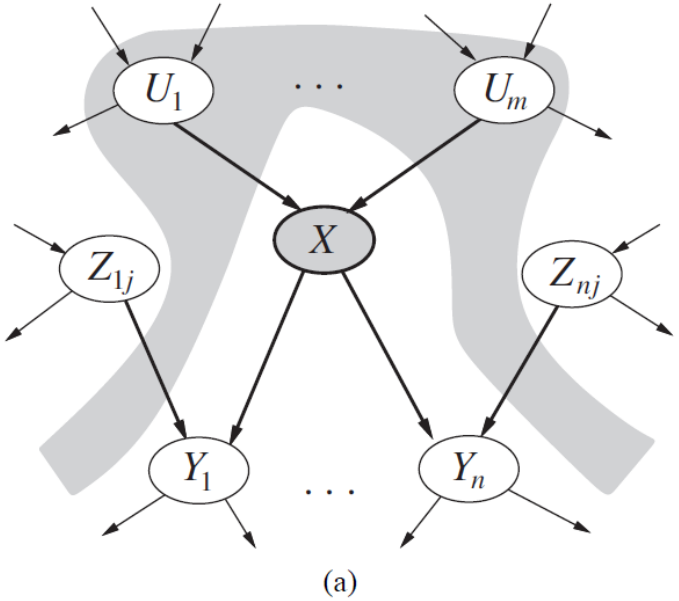






- 就像因果模型与诊断模型的区别，诊断概率**P**(E|A,B)很脆弱
- 坚持因果模型，只需要更少的概率值，这些值更容易获得

条件独立性



(a)

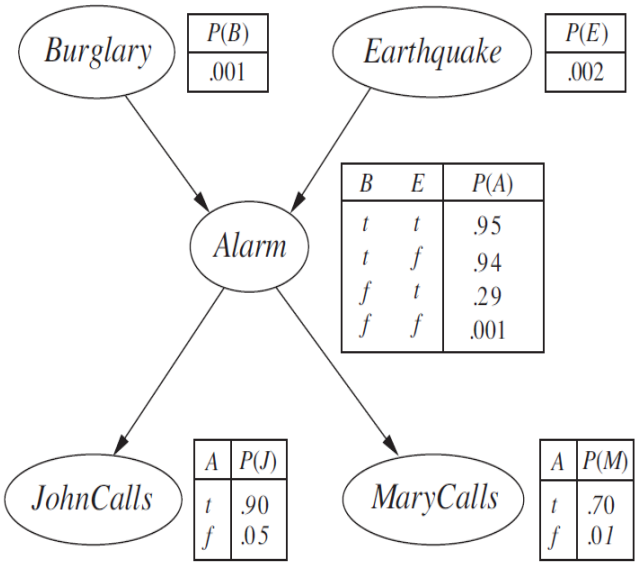
$P(x_n, \ldots, x_1) =$
$P(x_n \mid x_{n-1}, \ldots, x_1)\, P(x_{n-1} \mid x_{n-2}, \ldots, x_1)$
$\ldots P(x_2 \mid x_1)\, P(x_1)$

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

- 构造贝叶斯网络的方法，将我们带到结论：
  - 给定父结点，一个结点条件独立于它的其他祖先结点。
    P(X|U1,..Um,Parents(U1),..Parents(Um))=**P**(X|U1,..Um)
  - 给定父结点，每个变量条件独立于它的非后代结点
    P(X|U1,..Um,Parents(Y1),..Parents(Yn))=**P**(X|U1,..Um)

条件独立性

- 例如，证明：
- P(j|a,m)=P(j|a)

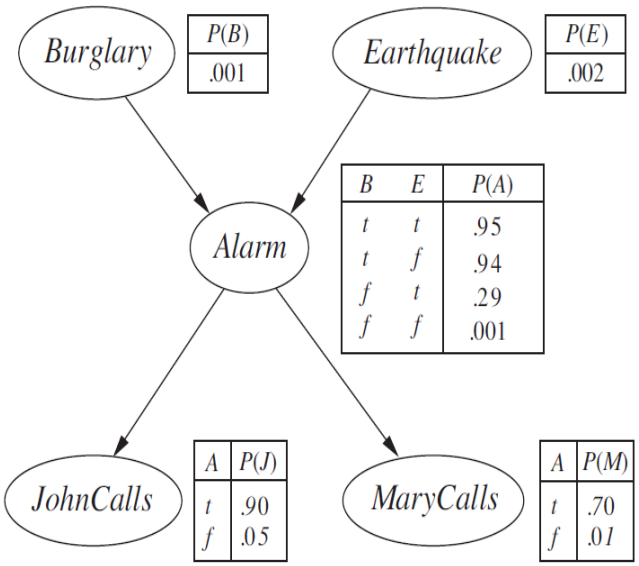


给定父结点，每个变量条件独立于它的非后代结点

条件独立性

- 例如，证明：
- P(j|a,m)=P(j|a)

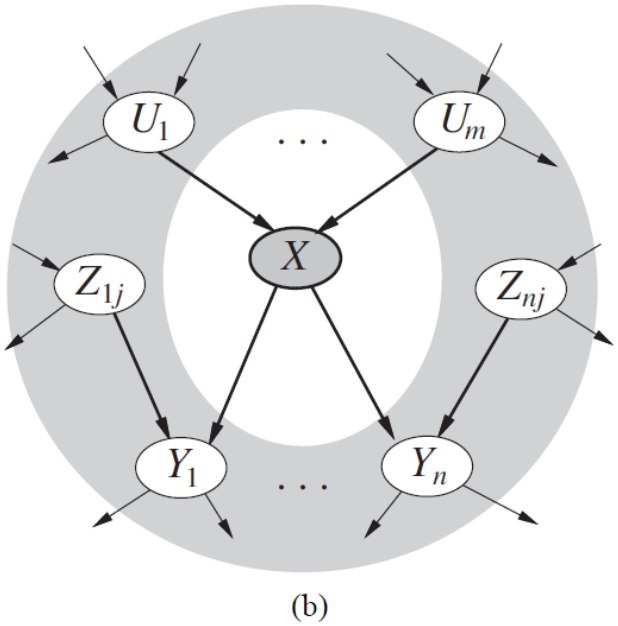
P(j|a,m)
= P(m,j,a)/P(a,m)
= P(j,m|a)P(a)/P(a,m)
= P(j|a)P(m|a)P(a)/P(a,m)
= P(j|a)P(m,a)/P(a,m)
= P(j|a)



给定父结点，每个变量条件独立于它的非后代结点

条件独立性

- 进一步可得出：
  - 给定一个结点的父结点、子结点、以及子结点的父结点——即给定它的**马尔可夫覆盖**（Markov blanket）——这个结点条件独立于网络中的所有其他结点。



(b)

P(X|Parents(X), Y1,…,Yn, Parents(Y1), … , Parents(Yn), Others)
= P(X|Parents(X), Y1,…,Yn, Parents(Y1), … , Parents(Yn))

有效表示条件概率

条件概率表的表示

- 即使最大父结点个数k很小(假设都为布尔变量)，要填满条件概率表仍然需要$O(2^k)$个数据

- 条件概率表的数据个数 有时 还可减少

确定性的结点

确定性结点

- 逻辑关系
  - 例如，父结点*Canadian*（加拿大人）、*US*（美国人）、*Mexican*（墨西哥人）与子结点*NorthAmerican*。子结点是其全部父结点的析取
- 数值关系
  - 例如，父结点是几个经销商销售一种特定型号汽车的价格，子结点是一个喜欢还价的人最后买这种型号汽车所出的价钱。子结点是其全部父结点值的最小值。

非确定性的结点

非确定性结点

- 噪声逻辑关系。标准的例子是**噪声或**（noisy-OR）关系。
- 例如：*Fever*为真，当且仅当*Cold*、*Flu*、或者*Malaria*为真。噪声或模型允许父结点与子结点之间的因果关系有可能被***抑制***，因此病人可能得了感冒却没有发烧。假设各**抑制概率**如下
  - $q_{\text{cold}} = P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6$ 感冒抑制下不发烧的概率
  - $q_{\text{flu}} = P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2$ 流感抑制下不发烧的概率
  - $q_{\text{malaria}} = P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1$ 疟疾抑制下不发烧的概率

cold而且flu抑制下不发烧的概率0.6x0.2

非确定性结点

- 噪声或模型假设所有可能的原因都已列出；假设每个父结点的抑制独立于其他父结点的抑制。给定这些假设，*Fever*为**假**的概率等于每个父结点的抑制概率的乘积。
  - $q_{\text{cold}} = P(\neg fever \mid cold, \neg flu, \neg malaria) = 0.6$
  - $q_{\text{flu}} = P(\neg fever \mid \neg cold, flu, \neg malaria) = 0.2$
  - $q_{\text{malaria}} = P(\neg fever \mid \neg cold, \neg flu, malaria) = 0.1$

Cold	Flu	Malaria	P(Fever)	P(¬Fever)
F	F	F		
F	F	T		**0.1**
F	T	F		**0.2**
F	T	T		
T	F	F		**0.6**
T	F	T		
T	T	F		
T	T	T		

非确定性结点

- 噪声或模型假设所有可能的原因都已列出；假设每个父结点的抑制独立于其他父结点的抑制。给定这些假设，*Fever*为**假**的概率等于每个父结点的抑制概率的乘积。
- 由抑制概率可推导出条件概率表

Cold	Flu	Malaria	P(Fever)	P(¬Fever)
F	F	F	? ?	?
F	F	T	? ?	**0.1**
F	T	F	? ?	**0.2**
F	T	T	? ?	?
T	F	F	? ?	**0.6**
T	F	T	? ?	?
T	T	F	? ?	?
T	T	T	? ?	?

非确定性结点

- 噪声或模型假设所有可能的原因都已列出；假设每个父结点的抑制独立于其他父结点的抑制。给定这些假设，*Fever*为**假**的概率等于每个父结点的抑制概率的乘积。
- 数据规模从$O(2^k)$ 降到了$O(k)$

Cold	Flu	Malaria	P(Fever)	P(¬Fever)
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	0.02 = 0.2 × 0.1
T	F	F	0.4	0.6
T	F	T	0.94	0.06 = 0.6 × 0.1
T	T	F	0.88	0.12 = 0.6 × 0.2
T	T	T	0.988	0.012 = 0.6 × 0.2 × 0.1

连续变量

	父结点离散	父结点连续
子结点离散		?
子结点连续	?	?

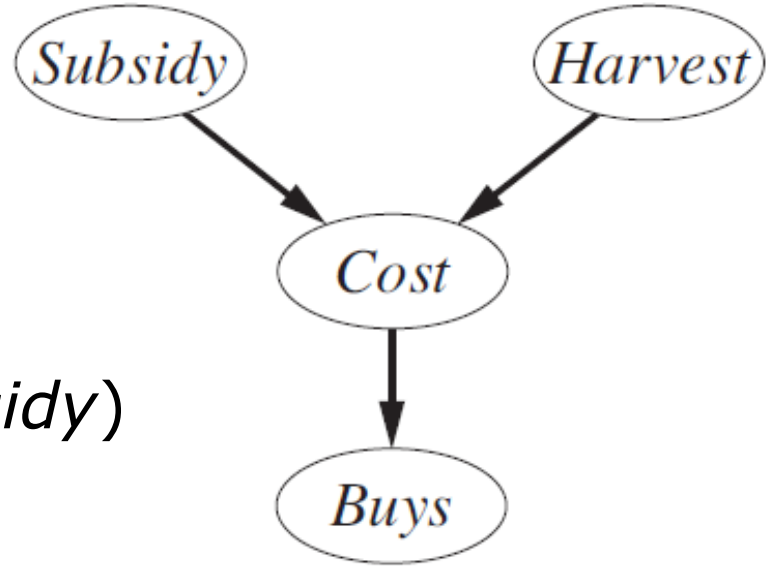
连续变量

- 可通过离散化来回避
- 或定义概率密度函数（参数化）
- 或用一组实例隐式地定义条件分布（非参数化），每个实例包含父结点与子结点变量的特定取值

	父结点离散	**父结点连续**
子结点离散	概率表	
子结点连续		

混合贝叶斯网络

- 同时包含离散随机变量和连续随机变量的网络称为**混合贝叶斯网络**
- 一个同时包含离散变量（*Subsidy(政府补助)*和*Buys*）和连续变量（*Harvest*和*Cost*）的简单网络



- 如何确定：
  P(*Cost* | *Harvest*, *Subsidy*)

	父结点离散	父结点连续
子结点离散	概率表	
子结点连续		？？？？

连续父结点 连续子结点

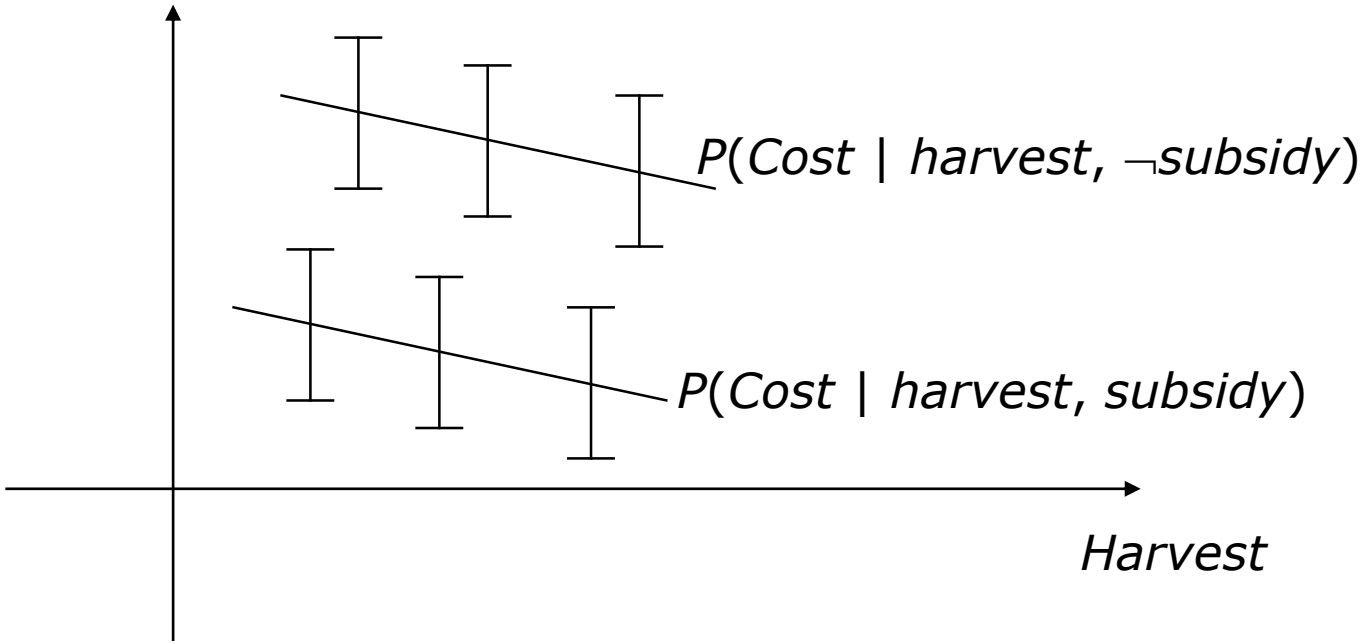
- 要确定**P**(*Cost* | *Harvest*, *Subsidy*)。
  分别确定*P*(*Cost* | *Harvest*, *subsidy*)以及*P*(*Cost* | *Harvest*, ¬*subsidy*)
- 线性高斯分布：子结点服从高斯分布，均值随父结点的值线性变化，标准差保持不变。

$$P(c \mid h, subsidy) = N(a_t h + b_t, \sigma_t^2)(c) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_t h+b_t)}{\sigma_t}\right)^2}$$

$$P(c \mid h, \neg subsidy) = N(a_f h + b_f, \sigma_f^2)(c) = \frac{1}{\sigma_f \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_f h+b_f)}{\sigma_f}\right)^2}$$

	父结点离散	父结点连续
子结点离散	概率表	
子结点连续		？？？？

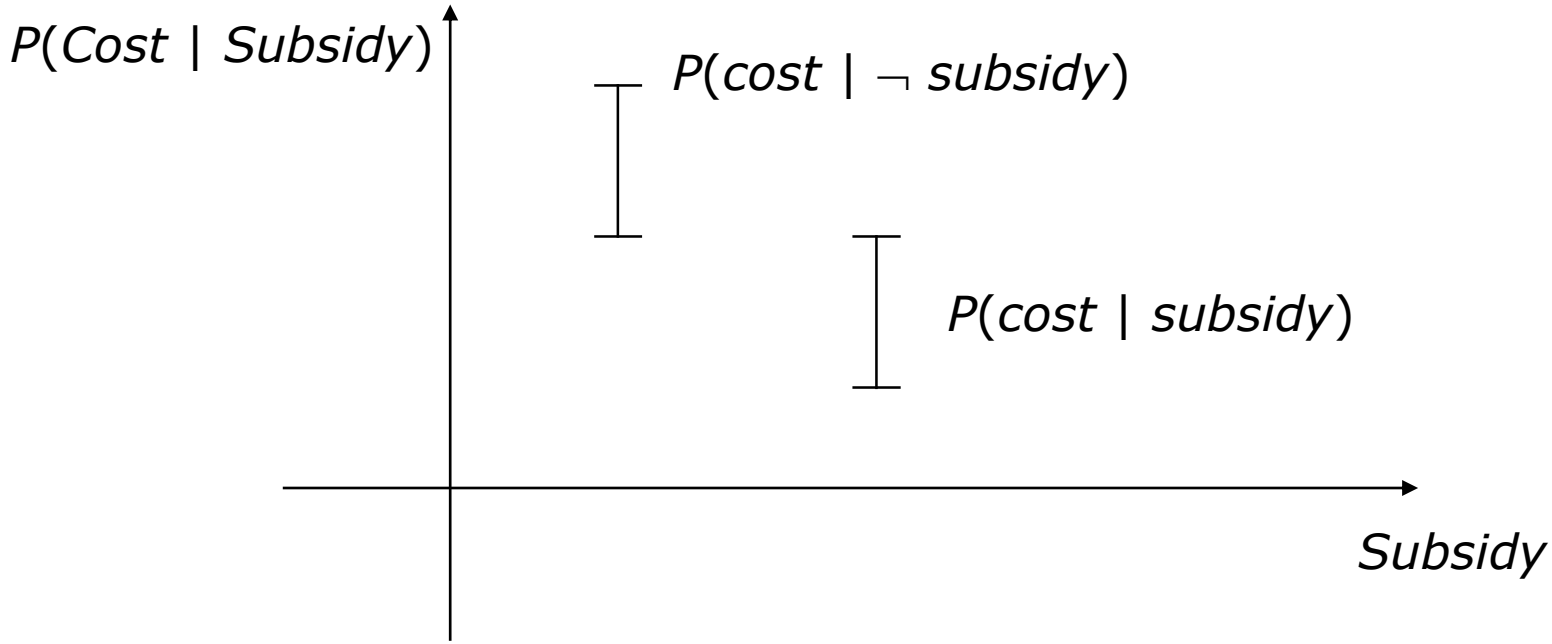
连续父结点 连续子结点



	父结点离散	父结点连续
子结点离散	概率表	
子结点连续	？？？？	

离散父结点 连续子结点

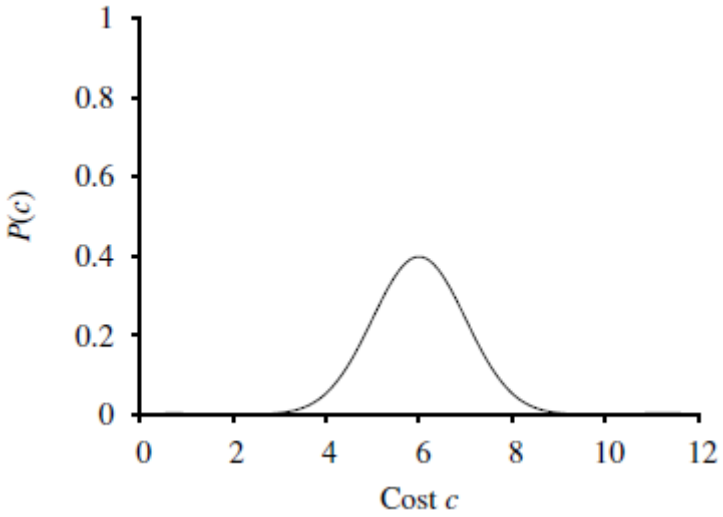
- 当父结点是离散变量，子结点是连续变量，就定义一个条件高斯分布：给定全部离散变量（父结点）的任意赋值，连续变量（子结点）的概率分布是一个多元高斯分布

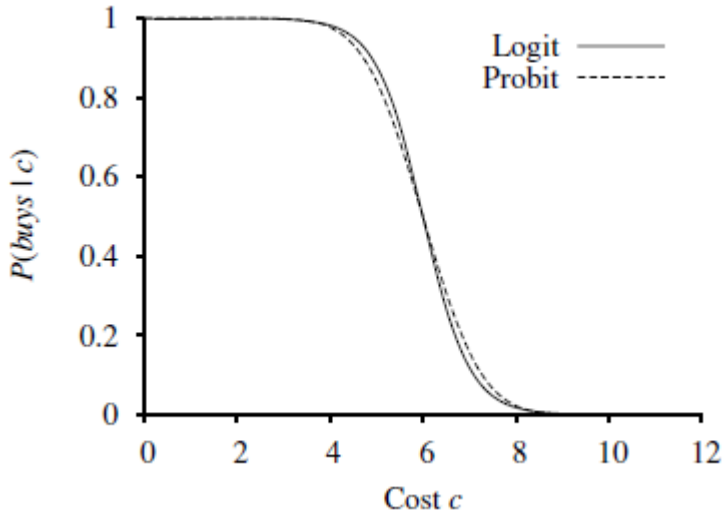


	父结点离散	父结点连续
子结点离散	概率表	？？？？
子结点连续		

连续父结点 离散子结点

- 考虑$P(buys \mid Cost = c)$
- 左：cost的正态分布
- 右：给定cost的正态分布下的buys的概率单位分布和逻辑单位分布





推理任务是什么？

贝叶斯网络的精确推理

推理任务

- 给定一组**证据变量**的赋值后，计算一组**查询变量**的后验概率分布。
- 每次只考虑一个查询变量；算法可以容易地扩展到有多个查询变量的情况。
- X表示查询变量；**E**表示证据变量集$E_1, E_2, \ldots, E_m$，**e**表示一个观察到的特定事件；**Y**表示非证据非查询变量集$Y_1, Y_2, \ldots, Y_l$（有时称为**隐藏变量**）。全部变量是$\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$，典型的查询是询问$\mathbf{P}(X \mid \mathbf{e})$。

通过枚举进行推理

- 任何条件概率都可以通过将完全联合概率分布中的某些项相加而计算得到

$$\mathbf{P}(X \mid \mathbf{e}) =$$

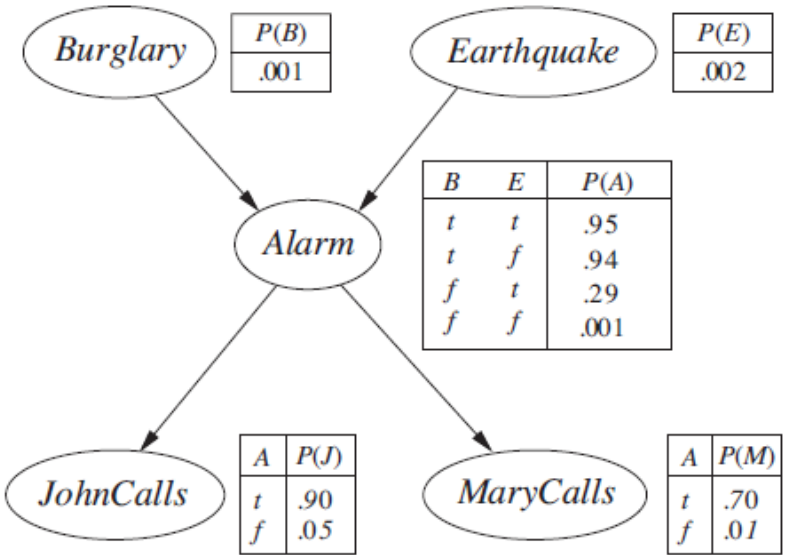
通过枚举进行推理

- 任何条件概率都可以通过将完全联合概率分布中的某些项相加而计算得到

$$\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha\sum_{\mathbf{y}}\mathbf{P}(X,\mathbf{e},\mathbf{y})$$

- $P(x, \mathbf{e}, \mathbf{y})$可以写成网络中的条件概率的乘积形式

- 考虑查询$\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$



防盗报警问题

- 考虑查询**P**(*Burglary* | *JohnCalls* = *true*, *MaryCalls* = *true*)

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

- *Burglary* = *true*的情况

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

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

$P(b \mid j, m) = \alpha \times 0.00059224$,

$\neg b$的相应计算结果为$\alpha \times 0.0014919$

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

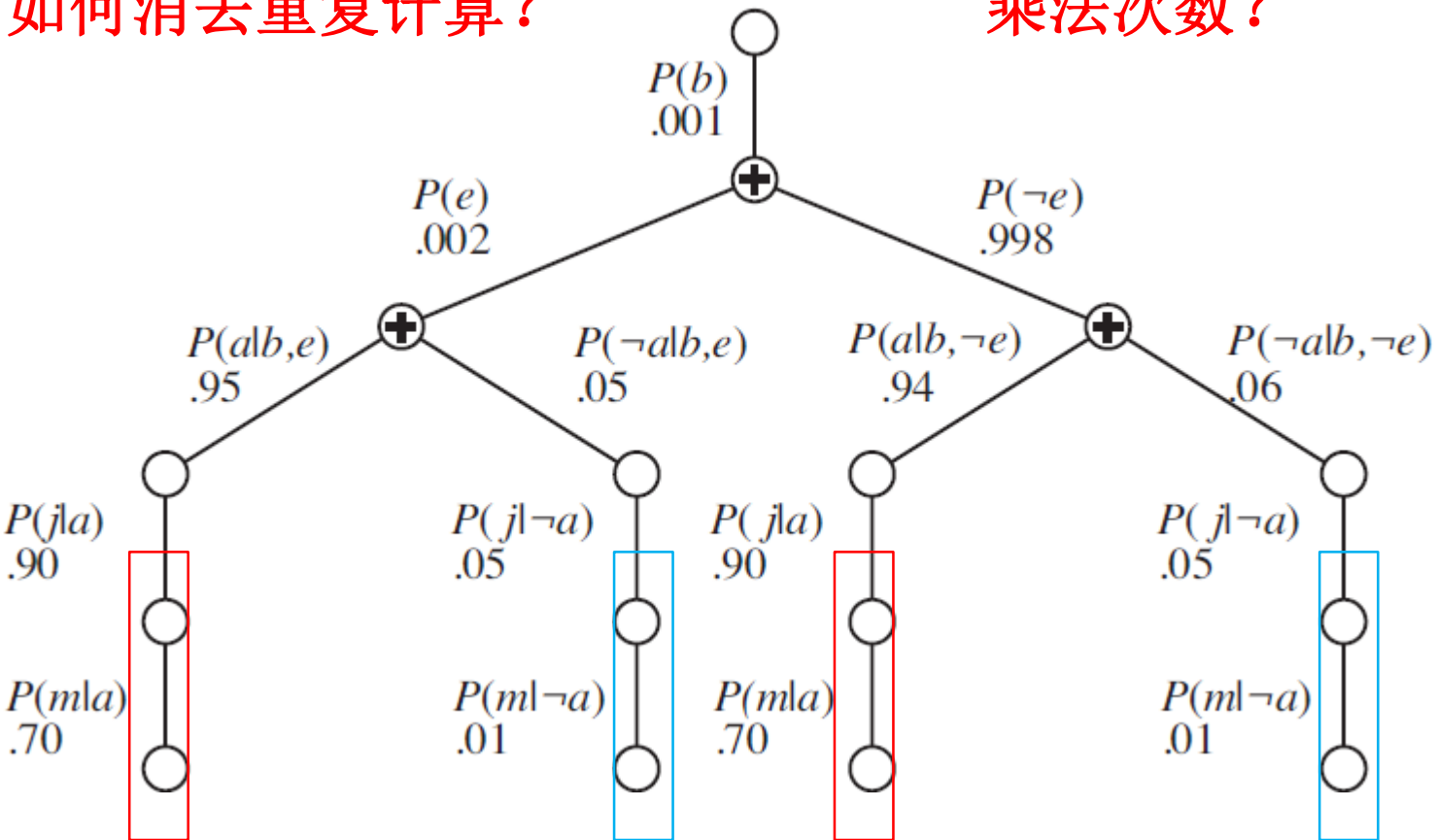
存在重复计算吗？

存在重复计算

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

如何消去重复计算？

乘法次数？



消去求和变量——避免重复计算

$$\mathbf{P}(B|j,m)=\alpha\,\underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)}\sum_e\underbrace{P(e)}_{\mathbf{f}_2(E)}\underbrace{\sum_a\underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)}\underbrace{P(j|a)}_{\mathbf{f}_4(A)}\underbrace{P(m|a)}_{\mathbf{f}_5(A)}}_{\mathbf{f}_6(B,E)}}_{\mathbf{f}_7(B)}$$

$$\mathbf{P}(B|j,m)=\alpha\mathbf{f}_1(B)\times\sum_e\mathbf{f}_2(E)\times\sum_a\mathbf{f}_3(A,B,E)\times\mathbf{f}_4(A)\times\mathbf{f}_5(A)$$

“×”不是普通的矩阵相乘，而是**逐点相乘**

如何消去求和变量？

消去求和变量——避免重复计算

$$\mathbf{P}(B|j,m)=\alpha\,\underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)}\sum_e\underbrace{P(e)}_{\mathbf{f}_2(E)}\underbrace{\sum_a\underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)}\underbrace{P(j|a)}_{\mathbf{f}_4(A)}\underbrace{P(m|a)}_{\mathbf{f}_5(A)}}_{\mathbf{f}_6(B,E)}}_{\mathbf{f}_7(B)}$$

$$\mathbf{P}(B|j,m)=\alpha\mathbf{f}_1(B)\times\sum_e\mathbf{f}_2(E)\times\sum_a\mathbf{f}_3(A,B,E)\times\mathbf{f}_4(A)\times\mathbf{f}_5(A)$$

$\mathbf{f}_6(B,E)$ “×”不是普通的矩阵相乘，而是**逐点相乘**

$$=\sum_a\mathbf{f}_3(A,B,E)\times\mathbf{f}_4(A)\times\mathbf{f}_5(A)$$

$$=(\mathbf{f}_3(a,B,E)\times\mathbf{f}_4(a)\times\mathbf{f}_5(a))+(\mathbf{f}_3(\neg a,B,E)\times\mathbf{f}_4(\neg a)\times\mathbf{f}_5(\neg a))$$

$\mathbf{f}_7(B)$

$$=\sum_e\mathbf{f}_2(E)\times\mathbf{f}_6(B,E)$$

$$=(\mathbf{f}_2(e)\times\mathbf{f}_6(B,e))+(\mathbf{f}_2(\neg e)\times\mathbf{f}_6(B,\neg e))$$

$$\mathbf{P}(B|j,m)=\alpha\mathbf{f}_1(B)\times\mathbf{f}_7(B)$$

乘法次数？

逐点相乘实例

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

消元顺序

$$\mathbf{P}(B|j,m)=\alpha\,\underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)}\sum_e\underbrace{P(e)}_{\mathbf{f}_2(E)}\sum_a\underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)}\underbrace{P(j|a)}_{\mathbf{f}_4(A)}\underbrace{P(m|a)}_{\mathbf{f}_5(A)}$$

- 先消A还是先消E？

消元顺序

$$\mathbf{P}(B|j,m)=\alpha\,\underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)}\sum_e\underbrace{P(e)}_{\mathbf{f}_2(E)}\sum_a\underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)}\underbrace{P(j|a)}_{\mathbf{f}_4(A)}\underbrace{P(m|a)}_{\mathbf{f}_5(A)}$$

- 如果先消A再消E：

$$\mathbf{P}(B|j,m)=\alpha\mathbf{f}_1(B)\times\sum_e\mathbf{f}_2(E)\times\sum_a\mathbf{f}_3(A,B,E)\times\mathbf{f}_4(A)\times\mathbf{f}_5(A)$$

- 如果先消E再消A：

$$\mathbf{P}(B|j,m)=\alpha\mathbf{f}_1(B)\times\sum_a\mathbf{f}_4(A)\times\mathbf{f}_5(A)\times\sum_e\mathbf{f}_2(E)\times\mathbf{f}_3(A,B,E)$$

- 不同顺序决定了不同的时间和空间要求

有些求和变量可直接删除

- 查询：**P**(*JohnCalls* | *Burglary* = *true*)

$$\mathbf{P}(J|b)=\alpha P(b)\sum_{e}P(e)\sum_{a}P(a|b,e)\mathbf{P}(J|a)\sum_{m}P(m|a)$$

- $\sum_m P(m \mid a)$等于1！*M*和这个查询*无关*。即使把结点*MaryCalls*从网络中删除，查询**P**(*JohnCalls* | *Burglary* = *true*)的结果也不会发生变化。
- 既非查询变量的祖先亦非证据变量的祖先的变量都和查询无关。

$$\mathbf{P}(X|\mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

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

大规模网络中可能难以高效计算

随机采样算法（蒙特卡洛算法）

贝叶斯网络的近似推理

蒙特卡洛算法

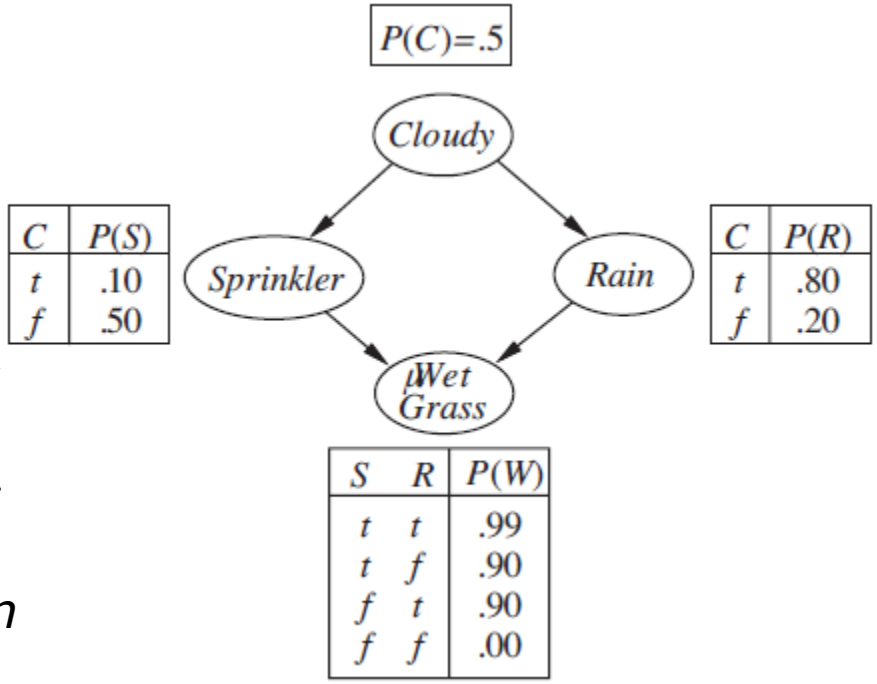
- 大规模多连通网络中的精确推理是不实际的。随机采样算法（也称为**蒙特卡洛算法**），能够给出一个问题的近似解，精度依赖于所生成的采样点的多少。

- 将描述两个算法家族：直接采样和马尔可夫链采样。

- 如何通过采样计算概率$P(X)$？

直接采样

- 按照拓扑顺序依次对每个变量进行采样。变量值被采样的概率分布依赖于父结点已得到的赋值。
- [*Cloudy*, *Sprinkler*, *Rain*, *WetGrass*] :
- 从**P**(*Cloudy*) = ⟨0.5, 0.5⟩ 中采样*Cloudy* ，假设返回*true*。
- 从**P**(*Sprinkler* | *Cloudy* = *true*) = ⟨0.1, 0.9⟩中采样*Sprinkler* ，假设返回*false*。
- 从**P**(*Rain* | *Cloudy* = *true*) = ⟨0.8, 0.2⟩ 中采样*Rain* ，假设返回*true*。
- 从**P**(*WetGrass* | *Sprinkler* = *false*, *Rain* = *true*) = ⟨0.9, 0.1⟩中采样*WetGrass* ，假设返回*true*。
- 这样，PRIOR-SAMPLE返回事件[*true*, *false*, *true*, *true*]。



直接采样

- 令$S_{PS}(x_1, \ldots, x_n)$为PRIOR-SAMPLE算法生成一个特定事件的概率。
- 假设总共有N个样本，令$N_{PS}(x_1, \ldots, x_n)$为特定事件$x_1, \ldots, x_n$在样本集合中出现的次数。$P(x_1, \ldots, x_m) \approx N_{PS}(x_1, \ldots, x_m) / N$
- 我们期望它和总样本数N的比在大样本极限下收敛到的它的期望值，与采样概率一致

$$\lim_{N\to\infty} \frac{N_{PS}(x_1,\ldots,x_n)}{N} = S_{PS}(x_1,\ldots,x_n) = P(x_1,\ldots,x_n)$$

- 如何通过采样计算条件概率$P(X \mid \mathbf{e})$？

拒绝采样

- 给定一个易于采样的分布，为一个难于采样的分布生成采样样本。
- 计算条件概率$P(X \mid \mathbf{e})$
- 首先，根据网络指定的先验分布生成采样样本。然后，它**拒绝所有与证据不匹配的样本**。最后，在剩余样本中通过统计$X = x$的出现的频次而计算出估计概率
- 可能是低效的。如何避免这种低效率？

似然加权

- **似然加权**（likelihood weighting）只生成与证据**e**一致的事件，从而避免拒绝采样算法的低效率。
- 固定证据变量**E**的值，然后只对非证据变量采样。这保证了生成的每个事件都与证据一致。
- 在对查询变量的分布进行计数之前，每个事件以它与证据吻合的似然（相似性）为权值。

似然加权

function LIKELIHOOD-WEIGHTING(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$

inputs: X, the query variable
e, observed values for variables **E**
bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
N, the total number of samples to be generated

local variables: **W**, a vector of weighted counts for each value of X, initially zero

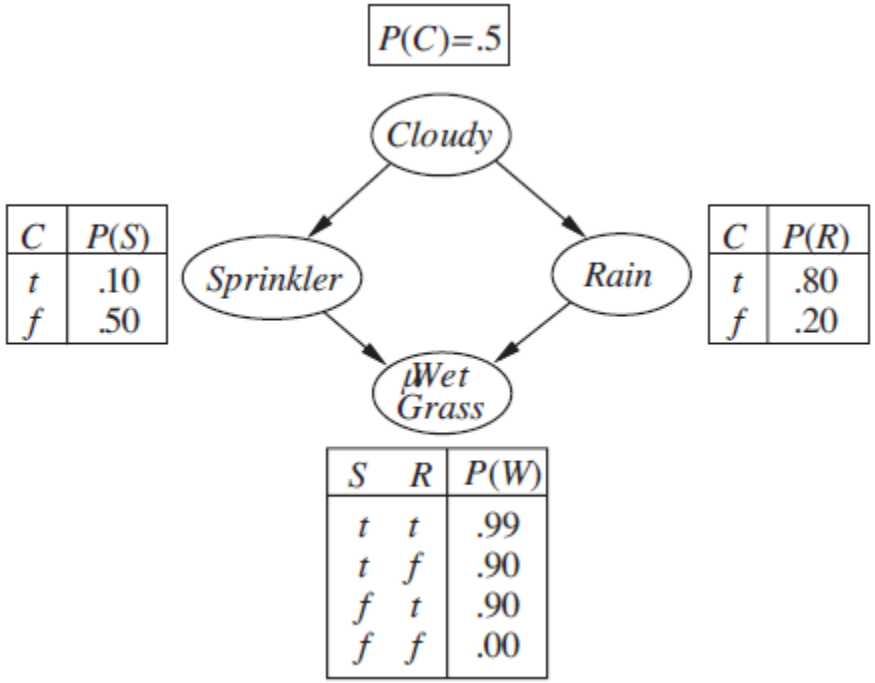
for $j = 1$ to N **do**
  x, $w \leftarrow$ WEIGHTED-SAMPLE(bn, **e**)
  $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in **x**
return NORMALIZE(**W**)

function WEIGHTED-SAMPLE(bn, **e**) **returns** an event and a weight

$w \leftarrow 1$; **x** $\leftarrow$ an event with n elements initialized from **e**
foreach variable X_i **in** $X_1, \ldots, X_n$ **do**
  if X_i is an evidence variable with value x_i in **e**
    then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
    else **x**$[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
return **x**, w

似然加权

- 求解查询**P**(*Rain* | *Cloudy* = *true*, *WetGrass* = *true*)，假设变量的拓扑顺序是*Cloudy*、*Sprinkler*、*Rain*、*WetGrass*。过程是这样的：首先，将权值*w*设为1.0。然后生成一个事件：



*1. Cloudy*是一个证据变量，其值为*true*。因此我们设置

$w \leftarrow w \times P(Cloudy = true) = 0.5$

*2. Sprinkler*不是一个证据变量，因此从**P**(*Sprinkler*|*Cloudy*=true) = ⟨0.1, 0.9⟩中采样；假设返回*false*。

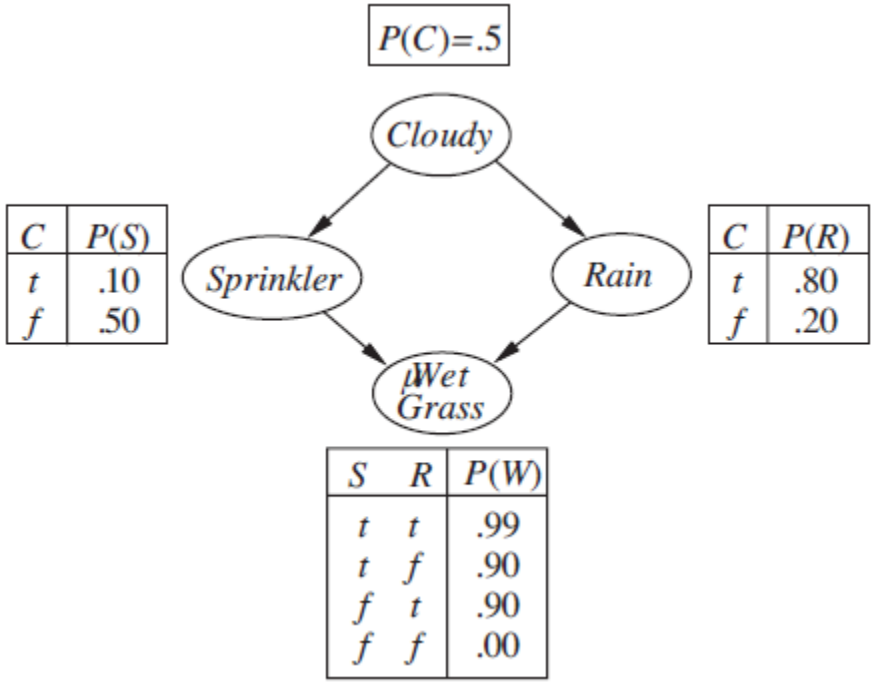
似然加权

3. 类似地，从**P**(*Rain* | *Cloudy* = *true*) = ⟨0.8, 0.2⟩ 中采样；假设返回*true*。

*4. WetGrass*是一个证据变量，其值为*true*。因此我们设置

$w \leftarrow w \times P(WetGrass = true \mid Sprinkler = false, Rain = true) = 0.5 \times 0.9 = 0.45$。

这里WEIGHTED-SAMPLE返回权值为0.45的事件[*true*, *false*, *true*, *true*]，它将被计入*Rain* = *true*中去。



马尔可夫链采样

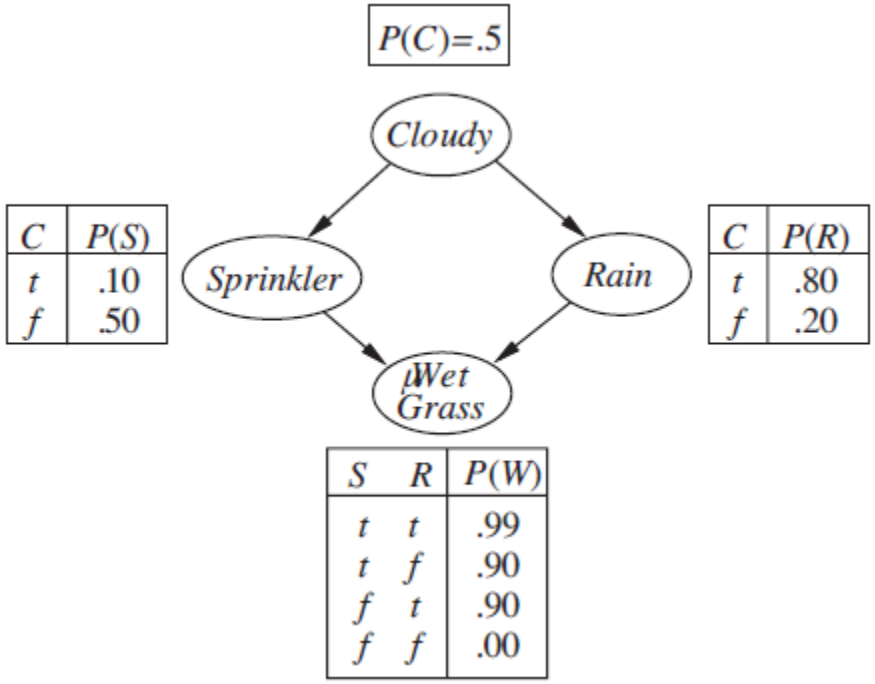
- 把**马尔可夫链蒙特卡洛**(MCMC)算法想象成：在特定的当前状态，每个变量的取值都已确定，然后随机修改当前状态，从而生成下一个状态

- Gibbs采样是一种特殊形式的MCMC算法，它特别适合贝叶斯网络。

Gibbs采样算法

- 贝叶斯网络的Gibbs采样算法从任意的状态出发，通过(给定马尔可夫覆盖)对一个非证据变量X_i随机采样而生成下一个状态。对X_i的采样条件依赖于X_i的马尔可夫覆盖中的变量的当前值。

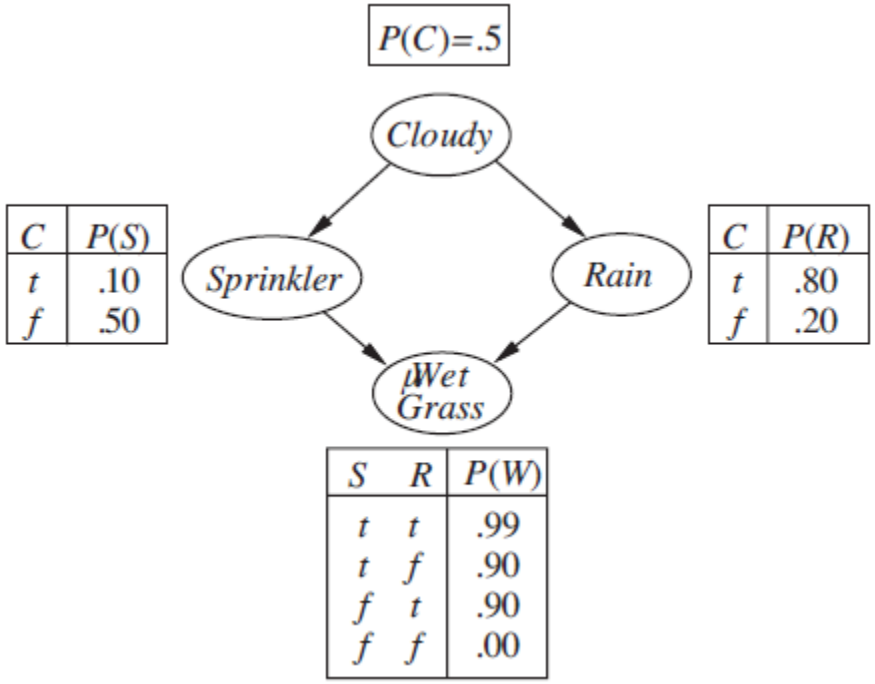
- 例如，查询**P**(*Rain* | *Sprinkler* = *true*, *WetGrass* = *true*)。



Gibbs采样算法

- 例如，查询**P**(*Rain* | *Sprinkler* = *true*, *WetGrass* = *true*)。

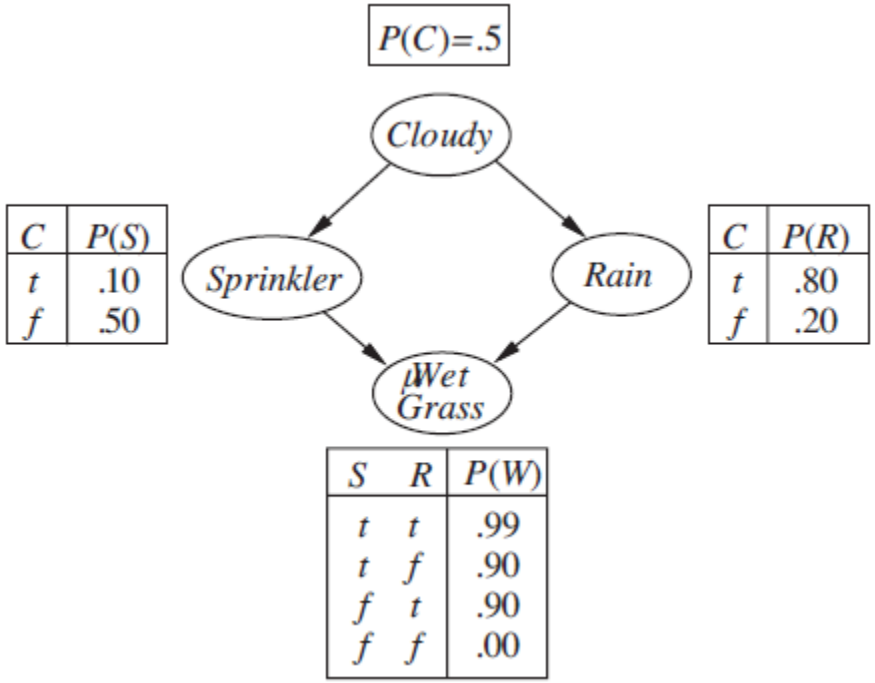
- 首先初始化：证据变量*Sprinkler*和*WetGRass*固定为它们的观察值，非证据变量*Cloudy*和*Rain*随机地初始化——比如*true*和*false*。因此，初始状态为
[*Cloudy* =*true*, *Sprinkler* =*true*, *Rain* =*false*, *WetGrass* =*true*]。



Gibbs采样算法

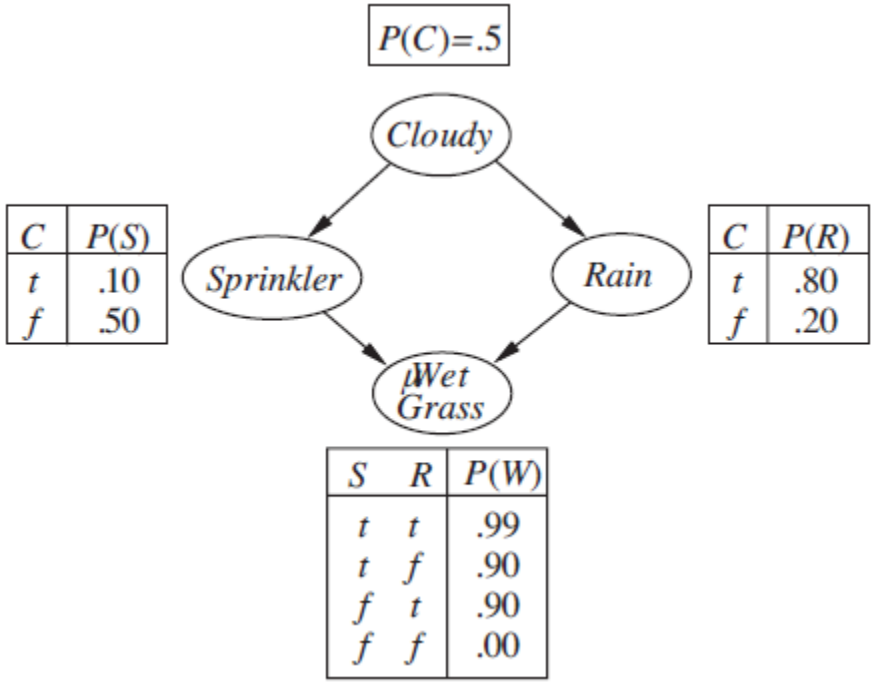
- 然后以任意顺序对非证据变量采样（循环执行）

1. 对*Cloudy*采样，给定它的马尔可夫覆盖变量的当前值：从**P**(*Cloudy* | *Sprinkler* = *true*, *Rain* = *false*)中采样。假设采样结果为*Cloudy* = *false*。那么新的当前状态是[*false*, *true*, *false*, *true*]。



Gibbs采样算法

2. 对*Rain*采样，给定它的马尔可夫覆盖变量的当前值：从 **P**(*Rain* | *Cloudy* = *false*, *Sprinkler* = *true*, *WetGrass* = *true*)中采样。假设采样结果为*Rain* = *true*。新的当前状态是[*false*, *true*, *true*, *true*]。



- 如果该过程访问了20个*Rain*为真的状态和60个*Rain*为假的状态，则所求查询的解为 NORMALIZE(⟨20, 60⟩) = ⟨0.25, 0.75⟩

Gibbs采样算法

- 给定马尔可夫覆盖，对变量进行采样"
- 证明结论：

$$P(x_i' \mid mb(X_i)) = \alpha P(x_i' \mid parents(X_i)) \times \prod_{Y_j \in Childen(X_i)} P(y_j \mid parents(Y_j))$$

- $mb(X_i)$表示X_i的马尔可夫覆盖$MB(X_i)$中各变量的取值

Gibbs采样算法

$$P(x_i \mid mb(X_i))$$
$$= \alpha P(x_i \mid parents(X_i)) \prod_i P(y_i \mid x_i, \mathbf{z}_i)$$
$$= \alpha P(x_i \mid parents(X_i)) \prod_i P(y_i \mid parents(Y_i))$$

- 证明：

$$\mathbf{P}(X_i \mid mb(X_i))$$
$$= \mathbf{P}(X_i \mid parents(X_i), \mathbf{y}, \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha \mathbf{P}(X_i, parents(X_i), \mathbf{y}, \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha \mathbf{P}(\mathbf{y} \mid X_i, parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l) \mathbf{P}(X_i \mid parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$\mathbf{P}(parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha' \mathbf{P}(\mathbf{y} \mid X_i, parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l) \mathbf{P}(X_i \mid parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha' \mathbf{P}(X_i \mid parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l) \mathbf{P}(\mathbf{y} \mid X_i, parents(X_i), \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha' \mathbf{P}(X_i \mid parents(X_i)) \mathbf{P}(\mathbf{y} \mid X_i, \mathbf{z}_1, ..., \mathbf{z}_l)$$
$$= \alpha' \mathbf{P}(X_i \mid parents(X_i)) \prod_i \mathbf{P}(y_i \mid X_i, \mathbf{z}_i)$$

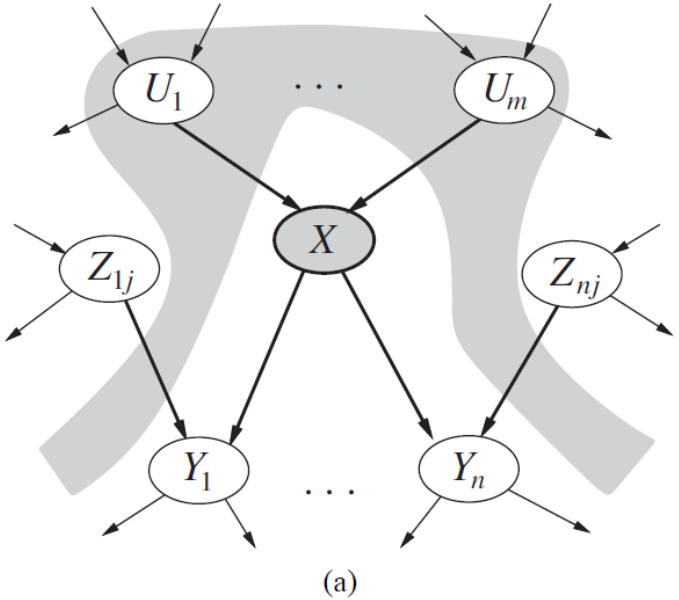
给定父结点，条件独立于非后代结点

给定父结点，条件独立其他祖先结点

y_i的父结点在X_i, z_i中，
$y_j (j \neq i)$是y_i的非后代结点

Y=Children(Xi)
y=children(Xi)
Parents(Yi)={Xi,**Z**i}
parents(Yi)={xi,**z**i}

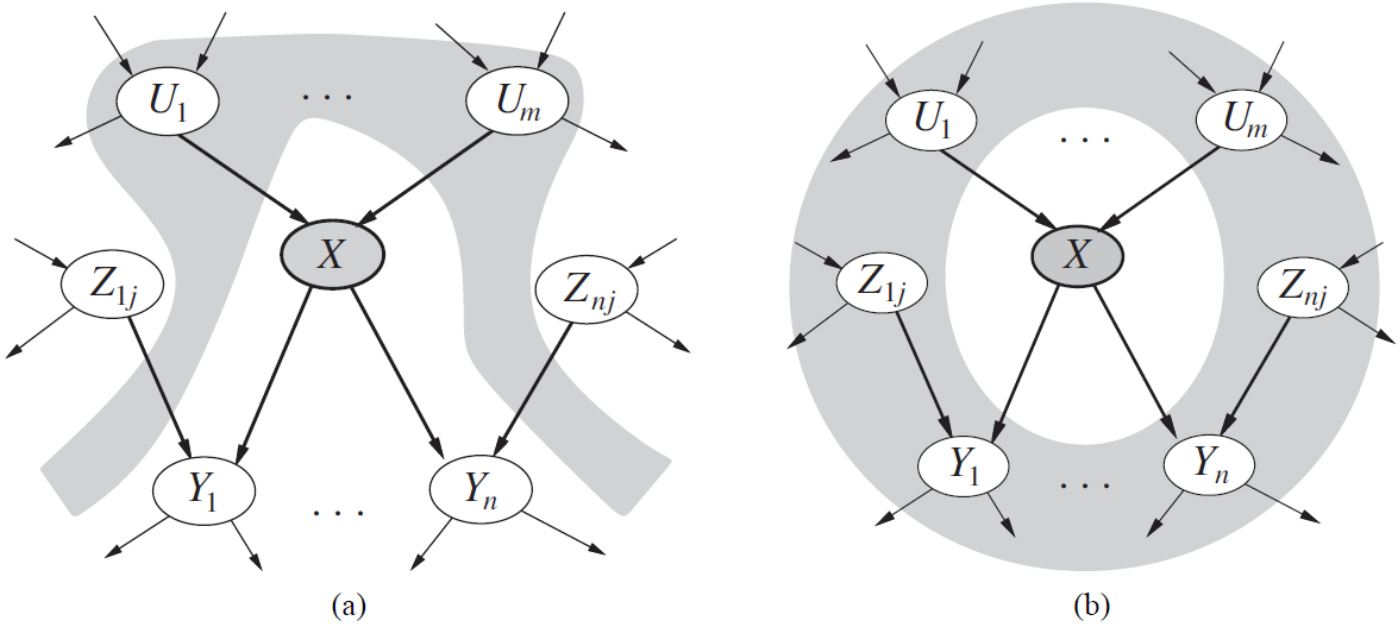
Review：条件独立性



(a)

- 构造贝叶斯网络的方法，将我们带到结论：
  - 给定父结点，一个结点条件独立于它的其他祖先结点。
    P(X|U1,..Um,Parents(U1),..Parents(Um))=**P**(X|U1,..Um)
  - 给定父结点，每个变量条件独立于它的非后代结点
    P(X|U1,..Um,Parents(Y1),..Parents(Yn))=**P**(X|U1,..Um)

Review：条件独立性



- 给定一个结点的父结点、子结点、以及子结点的父结点——也就是说，给定它的**马尔可夫覆盖**（Markov blanket）——这个结点条件独立于网络中的所有其他结点。

summary

1. 贝叶斯网络（何谓贝叶斯网络；从网络计算概率；如何构建贝叶斯网络；网络中的条件独立性）
2. 条件概率的有效表示（确定性结点；非确定性结点；连续变量）
3. 贝叶斯网络的精确推理（枚举；消去重复计算）
4. 贝叶斯网络的近似推理（直接采样；拒绝采样；加权；马尔科夫链采样）