第七章 贝叶斯网络

§7.1 背景知识

§7.2 贝叶斯网

§7.3 应用举例

§7.1 背景知识

一、问题引入

二、概念定义

问题引入

贝叶斯方法回顾

- 长久以来人们对一件事情发生或不发生的概率，只有固定的0和1，要么发生，要么不发生，从来不考虑发生的概率和不发生的概率
- 有一个袋子，里面装着若干个白球和黑球，计算从袋子中取得白球的概率
- **频率派**：要么取得白球，要么取得黑球，概率是确定的值，为1/2
- **贝叶斯派**：概率是个不确定的值，跟袋子中白球和黑球的先验分布有关
- 对于两类问题，采用最小错误率贝叶斯决策，使得后验概率最大的决策

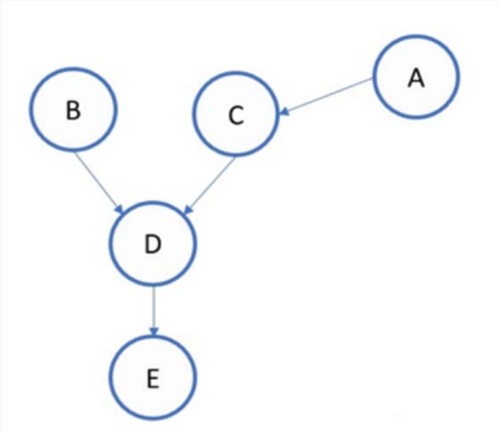
$$P(w_i|x) = \frac{p(x|w_i)P(w_i)}{p(x)} = \frac{p(x|w_i)P(w_i)}{\sum_{j=1}^{2} p(x|w_j)P(w_j)}, \quad i = 1,2$$

- 朴素贝叶斯分类**假设所有属性相互独立**，每个属性独立地对分类结果发生影响
- 半朴素贝叶斯分类适当考虑部分属性之间的相互依赖信息，常用的**独依赖估计策略**就是假设**每个属性在类别之外最多依赖一个其他属性**

概念定义

贝叶斯网络定义

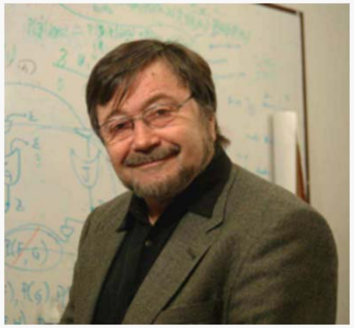
- 贝叶斯网络也称“信念网”(belief network)，它借助有向无环图来刻画属性之间的依赖关系，并使用条件概率表来描述属性的联合概率分布
- 一个贝叶斯网络B由结构G和参数Θ两部分构成，即$B = \langle G, \Theta \rangle$
  - G是一个**有向无环图**，其每个结点对应于一个属性，若两个属性有直接依赖关系，则由一条边连接起来
  - Θ是一个**条件概率表**，定量描述这种依赖关系，假设属性x_i在G中的父节点集为π_i，则Θ包含了每个属性的条件概率表示$\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$



概念定义

贝叶斯网络小典故

- 贝叶斯网络是一种概率图模型，于1985年由犹太裔美国计算机科学家Judea Pearl提出

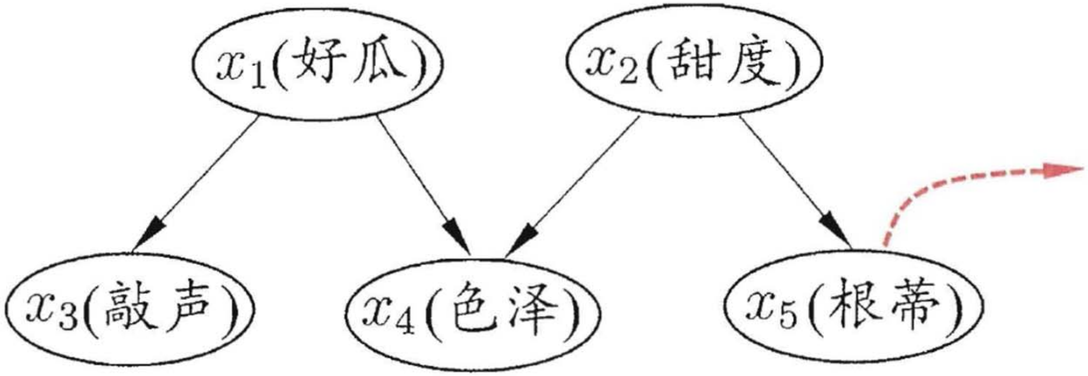


- 出生于特拉维夫，先后在以色列理工学院、罗格斯大学和布鲁克林理工学院获得电子工程专业学士、硕士和博士学位
- 1970年至今在加州大学洛杉矶分校计算机科学系任教至今
- 早期的人工智能研究专注以逻辑为基础的建模和推理，很难定量对不确定性事件进行表示和处理，Pearl在20世纪70年代将概率方法引入到人工智能，**开创了贝叶斯网络**的研究，**提出了信念传播**算法，**催生了概率图模型**这一类技术，以贝叶斯网为工具，**开创了因果推理**方面的研究，**获得2011年图灵奖**
- 儿子是《华尔街日报》记者，911追踪报道武装分子被残忍斩首

概念定义

贝叶斯网络举例

- 图中给出了西瓜问题的一种**贝叶斯网络结构**和属性“根蒂”的**条件概率表**。从图中网络结构可看出“色泽”直接依赖于“好瓜”和“甜度”，而“根蒂”则直接依赖于“甜度”，进一步从条件概率表能得到“根蒂”对“甜度”的量化依赖关系，如P(根蒂=硬挺|甜度=高)=0.1等



		硬挺	蜷缩
甜度	高	0.1	0.9
	低	0.7	0.3

§7.2 贝叶斯网

一、贝叶斯网的结构

二、贝叶斯网的学习

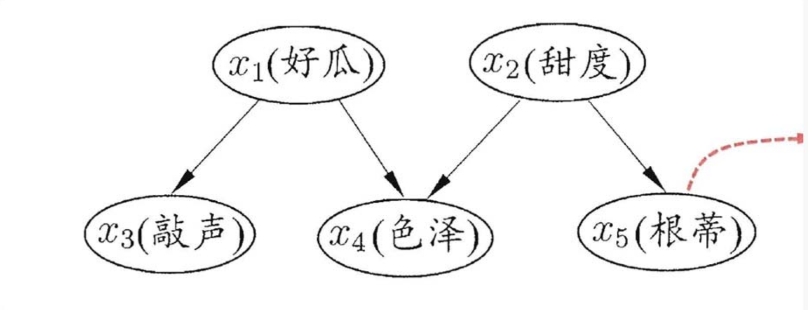
三、贝叶斯网的推断

贝叶斯网的结构

结构

- 贝叶斯网的结构**可有效表达属性之间的条件独立性**。给定父结点集，贝叶斯网假设每个属性与它的非后裔属性独立，于是$B=\langle G,\Theta\rangle$将属性$x_1,x_2,\ldots,x_d$的联合概率分布定义为

$$P_B(x_1,x_2,\ldots,x_d)=\prod_{i=1}^{d}P_B(x_i\mid\pi_i)=\prod_{i=1}^{d}\theta_{x_i|\pi_i}$$



- 以右图西瓜属性为例，联合概率分布定义为

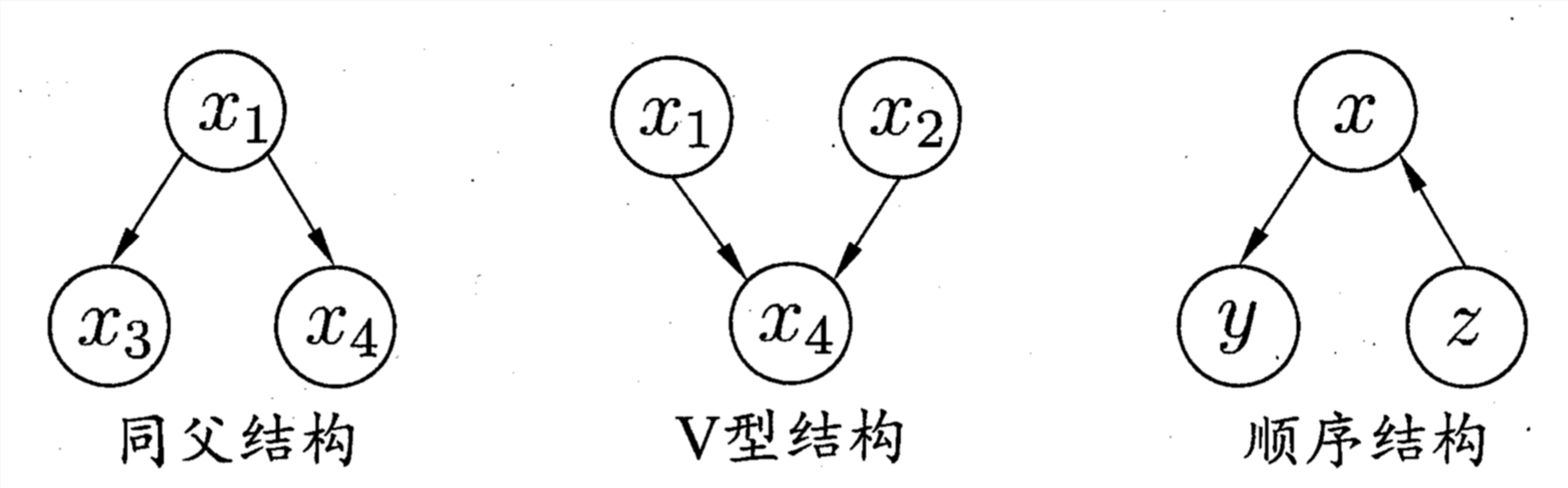
$$P(x_1,x_2,x_3,x_4,x_5)=P(x_1)P(x_2)P(x_3\mid x_1)P(x_4\mid x_1,x_2)P(x_5\mid x_2)$$

- 显然，x_3和x_4在给定x_1的取值时独立，x_4和x_5在给定x_2的取值时独立，分别简记为$x_3\perp x_4|x_1$和$x_4\perp x_5|x_2$

贝叶斯网的结构

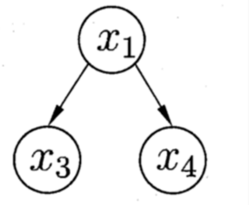
结构

- 贝叶斯网中变量之间的三种典型依赖关系

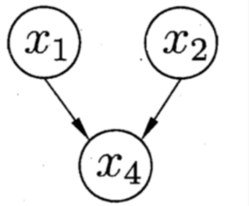


贝叶斯网的结构

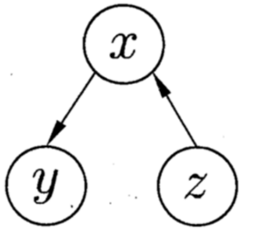
- 同父结构
  - 在同父结构中，给定父结点x_1的取值，则结点x_3与x_4条件独立（**当x_1发生时，x_3发生与否与x_4发生与否是无关的**）
- V型结构
  - V型结构也称冲撞结构，给定子结点x_4的取值，x_1和x_2必不独立
  - 若子结点x_4取值未知，则x_1和x_2独立，这样的独立性称为“边际独立性（marginal independence）记$x_1 \perp\!\!\!\perp x_2$
- 顺序结构
  - 在顺序结构中，给定结点x的值，则y与z条件独立



同父结构



V型结构



顺序结构

贝叶斯网的结构

- V型结构中，若x_4取值完全未知，证明x_1和x_2独立
  - 证明：由P8的贝叶斯网结构的定义，给定父结点集，贝叶斯网假设每个属性与它的非后裔属性独立，则有

$$P(x_1, x_2, x_4) = P(x_4 \mid x_1, x_2)P(x_1)P(x_2)$$

因此

$$P(x_1, x_2) = \sum_{x_4} P(x_1, x_2, x_4) = \sum_{x_4} P(x_4 \mid x_1, x_2)P(x_1)P(x_2) = P(x_1)P(x_2)$$

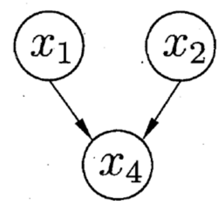
  - 需要注意的是，若x_4的取值已知$x_4 = a$，则有

$$P(x_1, x_2 \mid x_4 = a) = P(x_1)P(x_2)P(x_4 \mid x_1, x_2)/P(x_4 = a)$$

此时x_1, x_2的取值对于$x_4 = a$条件不独立

$\boldsymbol{x_1}$ ：学习努力程度，$\boldsymbol{x_2}$ ：课程难度，$\boldsymbol{x_4}$：期末考试成绩

在没有观察到期末考试成绩的时候，是否努力与课程难度显然是独立的；但已经考生得到高分的情况下，如果学习努力程度低，那么课程可能就不那么难，也就是说这两个变量不是独立的了

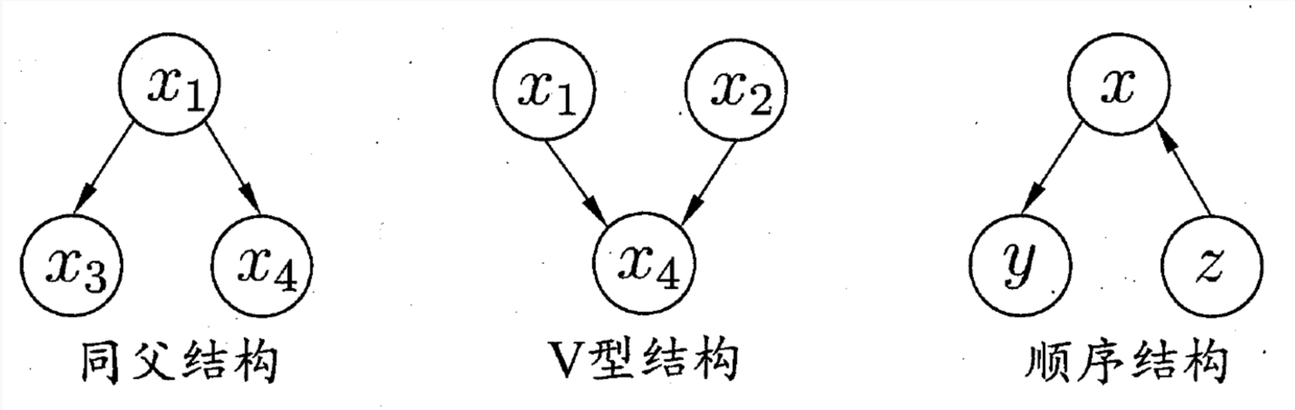


V型结构

贝叶斯网的结构

注意

- 事实上，一个变量取值的确定与否，能对另两个变量间的独立性发生影响，这个现象并非V型结构所特有
- 在同父结构中，条件独立性$x_3 \perp x_4 | x_1$成立，但若x_1的取值未知，则x_3和x_4就不独立，即$x_3 \perp\!\!\!\perp x_4$不成立
- 在顺序结构中，$y \perp z | x$ 但$y \perp\!\!\!\perp z$不成立



贝叶斯网的结构

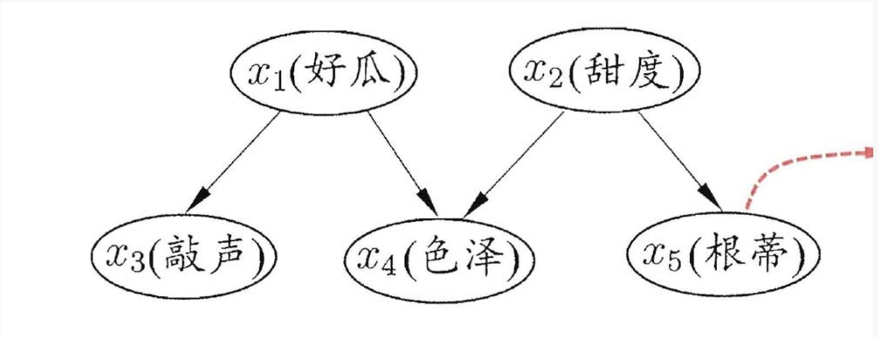
有向分离

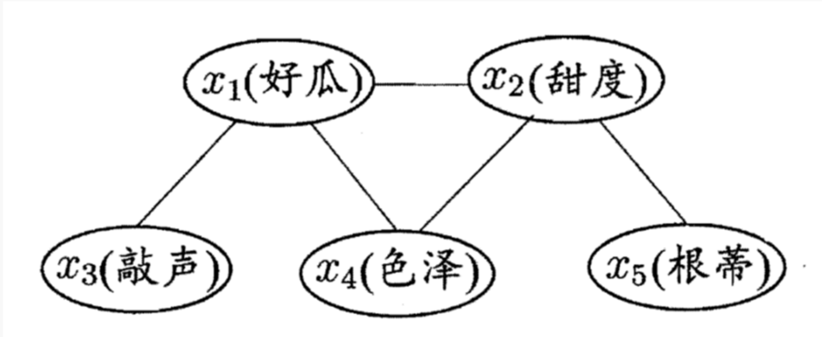
- 为了分析有向图中变量间的条件独立性，可使用“**有向分离**”把一个有向图转变为一个无向图
  - 找出有向图中的所有V型结构，在V型结构的两个父结点之间加上一条无向边
  - 将所有有向边改为无向边
- 由此产生的无向图称为“道德图”，令父结点相连的过程称为“道德化”

贝叶斯网的结构

有向分离

- 基于道德图能直观、迅速地找到变量间的条件独立性。假定道德图中有变量x, y和变量集合$z = \{z_i\}$，若变量x和y能在图上被z分开，即从道德图中将变量集合z去除后，x和y分属两个连通分支，则称变量x和y被z有向分离，$x \perp y | z$成立
- 例如，左图的道德图如右图所示，从图中能容易地找出所有的条件独立关系

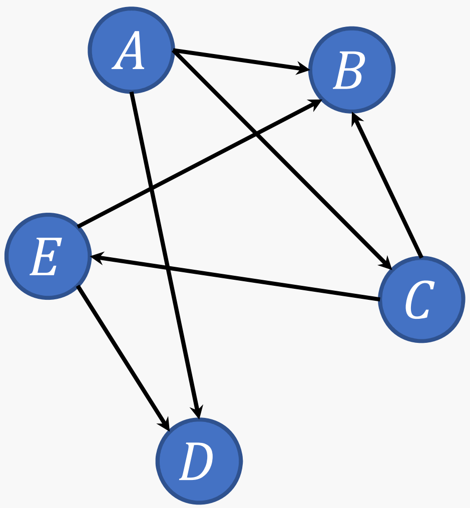


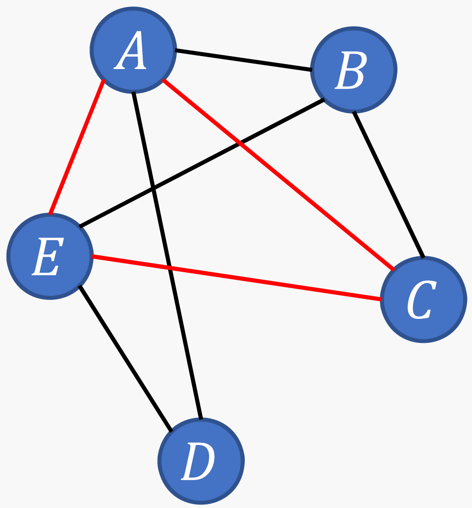


$x_3 \perp x_4 | x_1$，$x_4 \perp x_5 | x_2$，$x_3 \perp x_2 | x_1$，$x_3 \perp x_5 | x_1$，$x_3 \perp x_5 | x_2$

贝叶斯网的结构

- 思考题：对下面的网络做有向分离





贝叶斯网的学习

学习

- 若网络结构已知，即属性间的依赖关系已知，则贝叶斯网的学习过程相对简单，只需**通过对训练样本“计数”，估计出每个结点的条件概率表**即可
- 现实应用中我们往往**事先并不知道网络结构**，于是贝叶斯网学习的首要任务就是**根据训练数据集来找出结构最“恰当”的贝叶斯网**
- “评分搜索”是求解这一问题的常用办法

- 网络结构已知
  - 训练样本计数估计条件概率
- 网络结构未知
  - 利用评分搜索确定网络结构

贝叶斯网的学习

评分搜索

- 先定义一个评分函数(score function)，来评估贝叶斯网与训练数据的契合程度，然后基于这个评分函数来寻找结构最优的贝叶斯网
- 评分函数引入了关于我们**希望获得什么样贝叶斯网的归纳偏好**
- 常用评分函数通常基于信息论准则，将学习问题看作一个数据压缩任务，学习的目标是**找到一个能以最短编码长度描述训练数据的模型**，编码长度包括了描述模型自身所需的字节长度和使用该模型描述数据所需的字节长度

明确归纳偏好 → 设计评分函数 → 寻找最优结构

贝叶斯网的学习

评分搜索

- 应选择那个综合编码长度(包括描述网络和编码数据)最短的贝叶斯网，这就是“最小描述长度” (Minimal Description Length，简称MDL) 准则

贝叶斯网的学习

□ 评分搜索

- 给定训练集$D = \{x_1, x_2, \dots, x_m\}$，贝叶斯网$B = \langle G, \Theta \rangle$在$D$上的评分函数可写为

$$s(B \mid D) = f(\theta)|B| - LL(B \mid D),$$

其中$|B|$是贝叶斯网的参数个数，$f(\theta)$表示描述每个参数θ所需的字节数，

$$LL(B \mid D) = \sum_{i=1}^{m} \log P_B(x_i)$$

是贝叶斯网B的对数似然

- 第一项是**计算编码贝叶斯网B所需的字节数**，第二项是**计算B所对应的概率分布P_B所需的字节数**
- 学习就转化为一个优化任务，即寻找一个贝叶斯网B使评分函数$s(B \mid D)$最小

贝叶斯网的学习

□ 评分搜索

- 若 $f(\theta)=1$，即每个参数用1编码位描述，则得到AIC (Akaike Information Criterion)评分函数

$$\mathrm{AIC}(B \mid D)=|B|-LL(B \mid D)$$

- 若 $f(\theta)=\frac{1}{2}logm$，即每个参数用$\frac{1}{2}\log m$字节描述，则得到BIC(Bayesian Information Criterion)评分函数

$$\mathrm{BIC}(B \mid D)=\frac{\log m}{2}|B|-LL(B \mid D)$$

贝叶斯网的学习

评分搜索

- 若 $f(\theta)=0$，即不计算对网络进行编码的长度，则评分函数退化为负对数似然，相应的学习任务退化为极大似然估计
- 若贝叶斯网 $B=\langle G,\Theta\rangle$ 的网络结构G固定，则评分函数$s(B\mid D)$的第一项为常数，此时最小化 $s(B\mid D)$ 等价于对参数Θ的极大似然估计
- 参数$\theta_{x_i|\pi_i}$能直接在训练数据D上通过经验估计获得，即

$$\theta_{x_i|\pi_i}=\hat{P}_D(x_i\mid\pi_i)$$

其中 $\hat{P}_D(\cdot)$ 是D上的经验分布。

- 为了最小化评分函数$s(B\mid D)$，只需对网络结构进行搜索，而候选结构的最优参数可直接在训练集上计算得到

贝叶斯网的学习

评分搜索

- 从所有可能的网络结构空间搜索**贝叶斯网最优结构**是一个NP难问题，难以快速求解
- 有两种常用的策略能在有限时间内求得近似解

  第一种是贪心法，从某个网络结构出发，每次调整一条边(增加、删除或调整方向），直到评分函数值不再降低为止

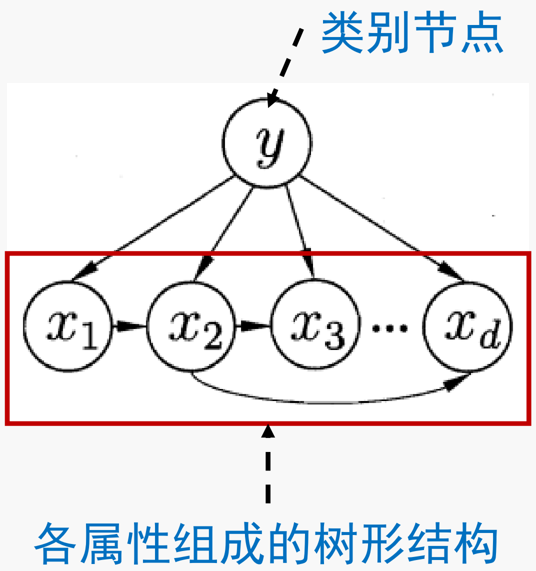
  第二种是通过给网络结构施加约束来削减搜索空间，例如将网络结构限定为树形结构等

贝叶斯网的学习

- 树扩展贝叶斯模型(Tree Augmented Naïve Bayes, TAN)
  - 基于最大带权生成树算法，将属性间的依赖关系约简为**树形结构**
  - 步骤：
    - 计算任意两个属性之间的条件互信息 $I(X_i, X_j|Y)$

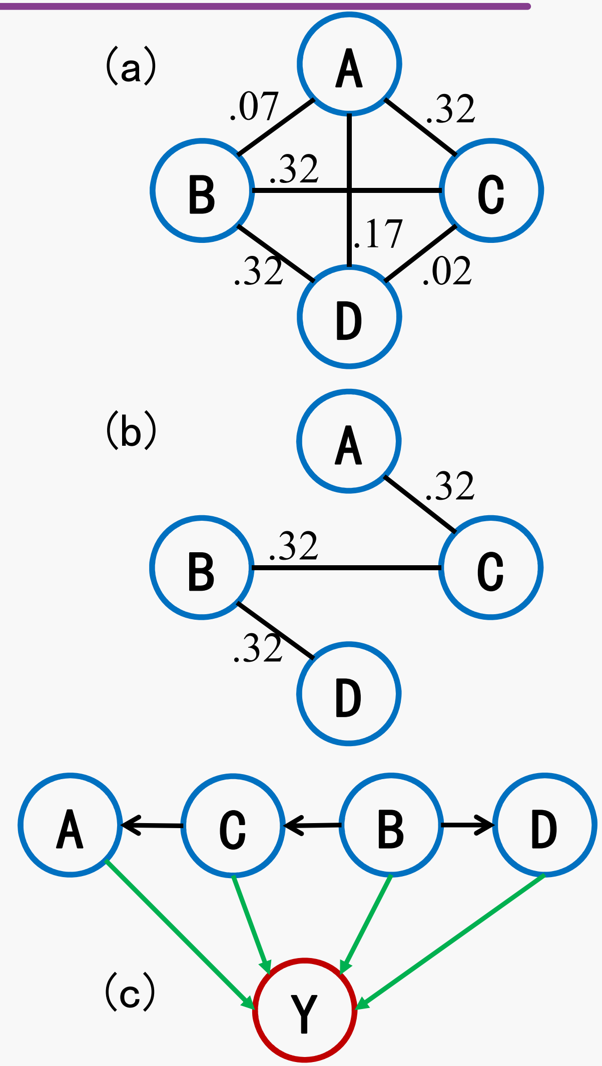
$$I(X_i, X_j|Y) = \sum_{x_i, x_j; c \in \mathcal{Y}} P(x_i, x_j|c) \log \frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)}$$

    - 以属性为节点构建完全图，任意两个节点之间的边权设为 $I(X_i, X_j|Y)$
    - 构建此完全图的最大带权生成树
    - 挑选根变量，将边置为有向
    - 加入类别节点 Y，增加每个属性到 Y 的有向边



贝叶斯网的学习

- 树扩展贝叶斯模型(Tree Augmented Naïve Bayes)
  - 假设已经由数据集计算得到了各属性之间的互信息，则可构建如右图所示带权无向完全图(a)
  - 搜索图(a)的“最大带权生成树”，得到图(b)
  - 选取某一属性作为根结点，将各边置为有向边，并加入类别节点，得到图(c)
- TAN算法得到的树是令似然函数值最大的树
  - 树形结构：减小贝叶斯网的复杂度
  - 最大权：最大化似然函数
  - TAN算法取得了贝叶斯网复杂度和精准度之间的权衡



清華大学

贝叶斯网的学习

- 请以西瓜数据集为训练集，基于BIC准则构建一个贝叶斯网
  - 对于某个节点X_i，设它的可能取值数目为r_i，它的父节点集为Π_i，父节点可能的取值组合数目为q_i，则

$$|B| = \sum_{i=1}^{n} q_i(r_i - 1)$$

概率和为1

  - 在数据集D中，父节点集Π_i取值组合为第 j 种组合的样本数目为m_{ij}，同时节点X_i取第 k 个值的样本数为m_{ijk}，则

$$\log P_B(X) = \sum_{i=1}^{n} \log P(X_i|\Pi_i)$$

$$= \sum_{i=1}^{n} \log \frac{m_{ijk}}{m_{ij}}$$

单样本

数据集

最大似然估计得到的网络参数，即对样本的计数

$$LL(B|D) = \sum_{s=1}^{|D|} \log P_B\left(X^{(s)}\right)$$

$$= \sum_{s=1}^{|D|} \sum_{i=1}^{n} \log \frac{m_{ij_s k_s}}{m_{ij_s}}$$

$$= \sum_{i=1} \sum_{j=1} \sum_{k=1} m_{ijk} \log \frac{m_{ijk}}{m_{ij}}$$

贝叶斯网的学习

- 以西瓜数据集2.0为训练集，试基于BIC准则构建一个贝叶斯网
  - 由上述讨论，可得 $BIC(B|D)$ 的表达式如下

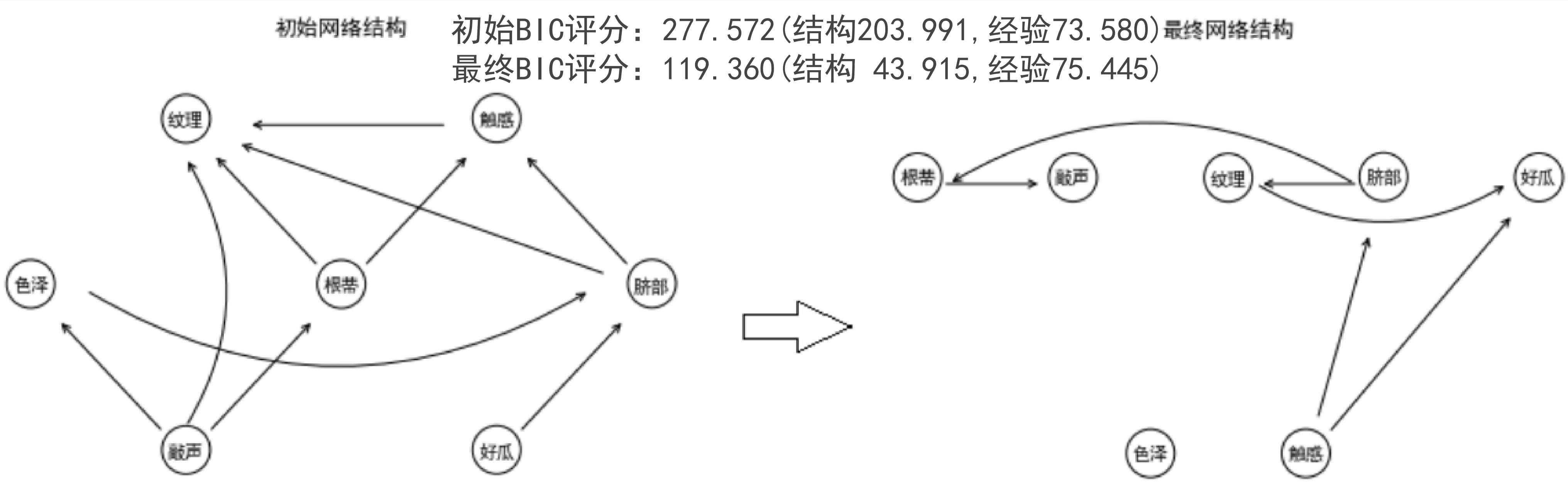
$$BIC(B|D) = \frac{\log m}{2}\sum_{i=1}^{n} q_i(r_i - 1) - \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} m_{ijk}\log\frac{m_{ijk}}{m_{ij}}$$

  - 之后，我们使用“下山法”搜索**BIC**评分最小的贝叶斯网
    - 每一步随机选取一个边进行调整：加边、减边、转向
    - 需要注意的是，加边和转向后可能会引入环，要予以避免

```
初始化网络结构B,
设置参数eta和tao,
for loop in range(step):
    随机选取需要调整连接边的两个节点xi和xj,
    对Bij值进行修改,
    if 新网络有环:
        continue
    if  BIC值减小 or  rand()<eta*exp(-loop/tao):
        更新网络, 输出当前BIC值
    else:
        continue
输出最终得到的网络结构和参数。
```

贝叶斯网的学习

- 以西瓜数据集2.0为训练集，试基于BIC准则构建一个贝叶斯网
  - 下面是上述算法某一次运行的结果：

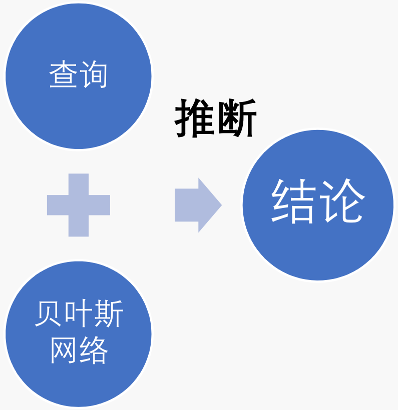


- “色泽”是一个独立的属性，是否是“好瓜”主要由“纹理”和“触感”两个属性决定

贝叶斯网的推断

□ 推断

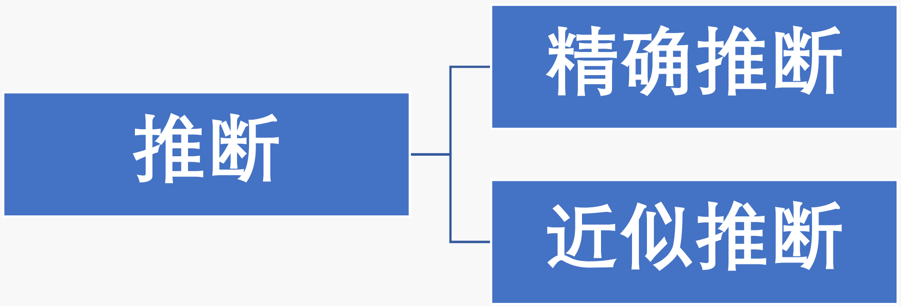
- 贝叶斯网训练好之后， 就能用来回答查询， 即通过一些属性变量的观测值来推测其他属性变量的取值
- 例如在西瓜问题中，如果我们观测到一个西瓜是**色泽青绿**、**敲声浊响**、**根蒂蜷缩**，我们想知道它是否成熟、甜度如何
- 通过已知变量 的 观测值来推测待查询变量的过程称为推断， 这 里 已知变量的观测值被称为证据



贝叶斯网的推断

□ 推断

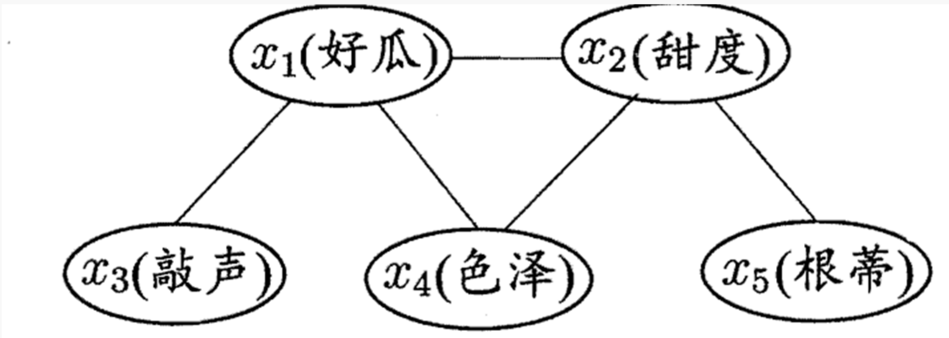
- 最理想的是直接根据贝叶斯网定义的联合概率分布来精确计算后验概率，然而这样的“精确推断”已被证明是NP难问题
- 当网络结点较多、连接稠密时，难以进行精确推断，需借助“近似推断”，通过降低精度的要求，在有限时间内求得近似解
- 在现实应用中，贝叶斯网的近似推断常采用**吉布斯采样**来完成，这是一种随机采样方法



贝叶斯网的推断

吉布斯采样

- 令 $Q=\{Q_1,Q_2,\ldots,Q_n\}$ 表示待查询变量，$E=\{E_1,E_2,\ldots,E_k\}$ 为证据变量，已知其取值为 $e=\{e_1,e_2,\ldots,e_k\}$。目标是计算后验概率 $P(Q=q\mid E=e)$，其中 $q=\{q_1,q_2,\ldots,q_n\}$ 是待查询变量的一组取值
- 以西瓜问题为例，待查询变量为 $Q=\{$ 好瓜，甜度 $\}$，证据变量为 $E=\{$ 色泽，敲声，根蒂 $\}$，且已知其取值为 $e=\{$ 青绿，浊响，蜷缩 $\}$，查询的目标值是 $q=\{$ 是，高 $\}$，即这是好瓜且甜度高的概率有多大



贝叶斯网的推断

□ 吉布斯采样

- 用于难以直接采样时从某一多变量概率分布中近似抽取样本序列
- 先随机产生一个与证据 $E = e$ 一致的样本 q^0 作为初始点，然后从当前样本出发产生下一个样本
- 在第 t 次采样中，先假设 $q^t = q^{t-1}$，然后对非证据变量逐个进行采样改变其取值，采样概率根据贝叶斯网 B 和其他变量的当前取值(即 $Z = z$)计算获得

输入: 贝叶斯网 $B = \langle G, \Theta \rangle$;
采样次数 T;
证据变量 $\mathbf{E}$ 及其取值 $\mathbf{e}$;
待查询变量 $\mathbf{Q}$ 及其取值 $\mathbf{q}$.

过程:

$n_q = 0$
$\mathbf{q}^0 =$ 对 $\mathbf{Q}$ 随机赋初值
for $t = 1, 2, \ldots, T$ **do**
  for $Q_i \in \mathbf{Q}$ **do**
    $\mathbf{Z} = \mathbf{E} \cup \mathbf{Q} \setminus \{Q_i\}$;
    $\mathbf{z} = \mathbf{e} \cup \mathbf{q}^{t-1} \setminus \{q_i^{t-1}\}$;
    根据 B 计算分布 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$;
    $q_i^t =$ 根据 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$ 采样所获 Q_i 取值;
    $\mathbf{q}^t =$ 将 $\mathbf{q}^{t-1}$ 中的 q_i^{t-1} 用 q_i^t 替换
  end for
  if $\mathbf{q}^t = \mathbf{q}$ **then**
    $n_q = n_q + 1$
  end if
end for

输出: $P(\mathbf{Q} = \mathbf{q} \mid \mathbf{E} = \mathbf{e}) \simeq \frac{n_q}{T}$

图 7.5 吉布斯采样算法

贝叶斯网的推断

吉布斯采样

- 经过T次采样得到与q一致的样本共有n_q个，则可近似估算出后验概率

$$P(Q=q \mid E=e) \simeq \frac{n_q}{T}$$

输入： 贝叶斯网 $B = \langle G, \Theta \rangle$;
采样次数 T;
证据变量 $\mathbf{E}$ 及其取值 $\mathbf{e}$;
待查询变量 $\mathbf{Q}$ 及其取值 $\mathbf{q}$.

过程：

$n_q = 0$
$\mathbf{q}^0 =$ 对 $\mathbf{Q}$ 随机赋初值
for $t = 1, 2, \ldots, T$ **do**
  for $Q_i \in \mathbf{Q}$ **do**
    $\mathbf{Z} = \mathbf{E} \cup \mathbf{Q} \setminus \{Q_i\}$;
    $\mathbf{z} = \mathbf{e} \cup \mathbf{q}^{t-1} \setminus \{q_i^{t-1}\}$;
    根据 B 计算分布 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$;
    $q_i^t =$ 根据 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$ 采样所获 Q_i 取值;
    $\mathbf{q}^t =$ 将 $\mathbf{q}^{t-1}$ 中的 q_i^{t-1} 用 q_i^t 替换
  end for
  if $\mathbf{q}^t = \mathbf{q}$ **then**
    $n_q = n_q + 1$
  end if
end for

输出： $P(\mathbf{Q} = \mathbf{q} \mid \mathbf{E} = \mathbf{e}) \simeq \frac{n_q}{T}$

图 7.5 吉布斯采样算法

贝叶斯网的推断

吉布斯采样

- 吉布斯采样是在贝叶斯网所有变量的联合状态空间与证据 $E=e$ 一致的子空间中进行 "**随机漫步**" (random walk)，每一步仅依赖于前一步的状态，是一个 "**马尔可夫链**" (Markov chain)
- 在一定条件下，无论从什么初始状态开始，马尔可夫链第 t 步的状态分布在 $t \to \infty$ 时必收敛于一个平稳分布(stationary distribution)
- 对于吉布斯采样来说，这个分布恰好是 $P(Q \mid E=e)$。因此，在 T 很大时，吉布斯采样相当于根据 $P(Q \mid E=e)$ 采样，从而保证了收敛于 $P(Q=q \mid E=e)$
- 由于马尔可夫链通常需很长时间才能趋于平稳分布，因此吉布斯采样算法的收敛速度较慢。若贝叶斯网中存在极端概率 "0" 或 "1"，则不能保证马尔可夫链存在平稳分布，此时吉布斯采样会给出错误的估计结果

吉布斯采样应用举例

吉布斯采样

- 假设离散随机变量X，Y服从联合分布$\mathbf{Pr}(X = x, Y = y) = p_{X,Y}(x, y)$：

$X\backslash Y$	0	1
0	0.6	0.1
1	0.15	0.15

- 则条件分布为：

$$\Pr(X = 0 \mid Y = 0) = \frac{p_{X,Y}(X = 0, Y = 0)}{p_Y(Y = 0)} = \frac{0.6}{0.6 + 0.15} = 0.8$$

- 同理：

$$\Pr(X = 0 \mid Y = 0) = 0.8 \qquad \Pr(Y = 0 \mid X = 0) = 6/7$$
$$\Pr(X = 1 \mid Y = 0) = 0.8 \qquad \Pr(Y = 1 \mid X = 0) = 1/7$$
$$\Pr(X = 0 \mid Y = 1) = 0.4 \qquad \Pr(Y = 0 \mid X = 1) = 0.5$$
$$\Pr(X = 1 \mid Y = 1) = 0.6 \qquad \Pr(Y = 1 \mid X = 1) = 0.5$$

吉布斯采样应用举例

```
#returns 1 with probability p, and 0 with probability 1-p
rbernoulli=function(p){return(1*runif(1)<p)}

# sample from distribution X given Y above
sample_XgivenY = function(y){
 if(y==0){
   x = rbernoulli(0.2) # returns 1 with probability 0.2; otherwise 0
 } else {
   x = rbernoulli(0.6)
 }
  return(x)
}
#' sample from distribution Y given X above
sample_YgivenX = function(x){
  if(x==0){
    y = rbernoulli(1/7)
  } else {
    y = rbernoulli(0.5)
  }
  return(y)
}
set.seed(100)
niter = 1000
X = rep(0,niter)
Y = rep(0,niter)
X[1]=1
Y[1]=1 # start from (1,1)
for(i in 2:niter){
  X[i] = sample_XgivenY(Y[i-1])
  Y[i] = sample_YgivenX(X[i])
}
res = data.frame(X=X,Y=Y)
```

吉布斯采样代码

	X	Y
1	1	1
2	1	1
3	1	1
4	1	1
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	1	0

前20次采样结果

采样分布

```
table(data.frame(X=X,Y=Y))/niter
```

```
   Y
X         0       1
  0 0.617 0.092
  1 0.154 0.137
```

X**Y	**0**	**1**
0	0.6	0.1
1	0.15	0.15

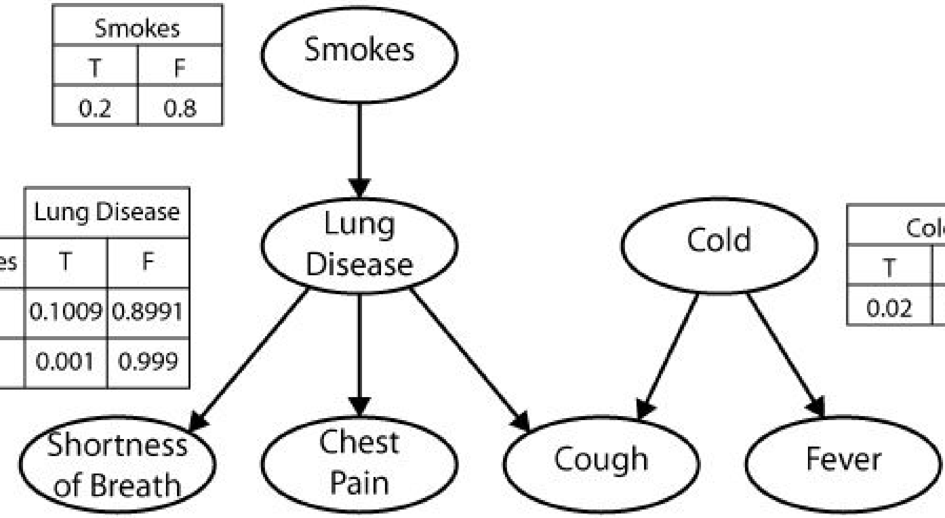
§7.3 应用举例

一、医学数据分析

二、卫星故障预测

应用举例：医学数据分析

- 医学数据分析
  - **任务：**根据症状诊断疾病
  - **困难：**
    - 多种症状因果关系
    - 多种疾病并发关系



Smokes	
T	F
0.2	0.8

	Lung Disease	
Smokes	T	F
T	0.1009	0.8991
F	0.001	0.999

Cold	
T	F
0.02	0.98

	Shortness of Breath	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

	Chest Pain	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

		Cough	
Lung Disease	Cold	T	F
T	T	0.7525	0.2475
T	F	0.505	0.495
F	T	0.505	0.495
F	F	0.01	0.99

	Fever	
Cold	T	F
T	0.307	0.693
F	0.01	0.99

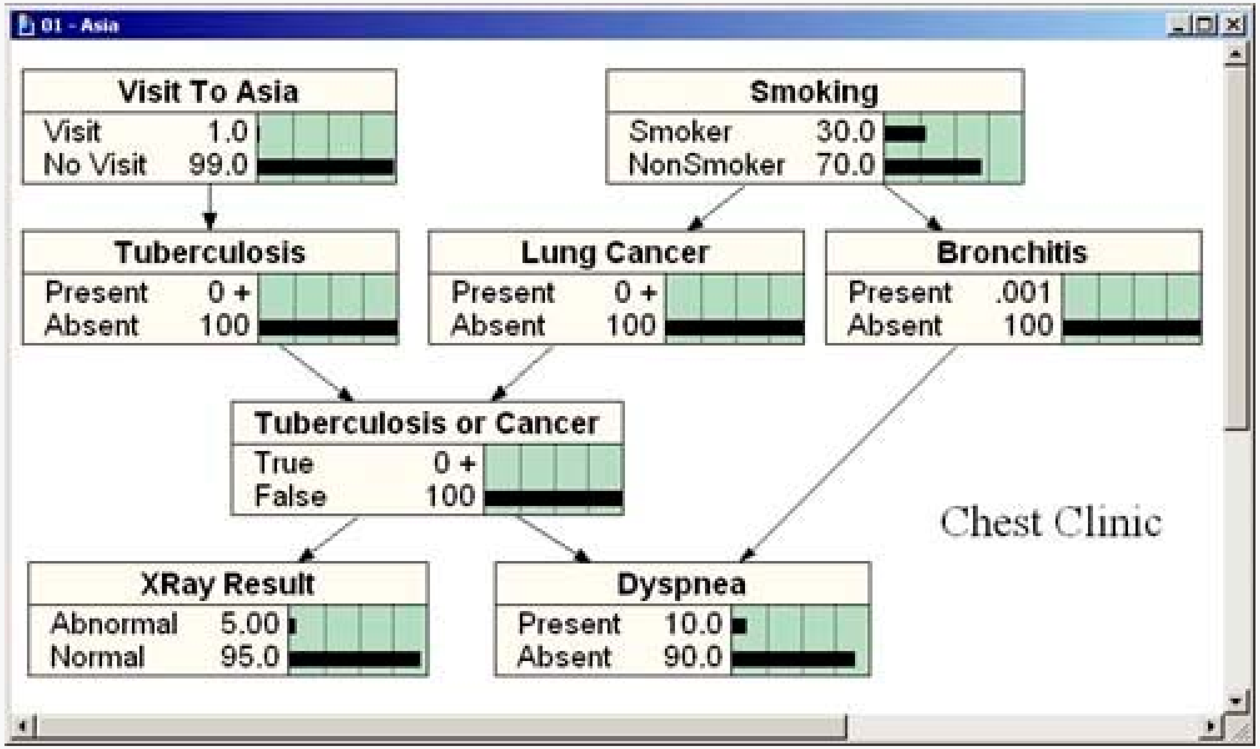
应用举例：医学数据分析

医学数据分析

- 假想你是洛杉矶市一名新毕业的医生，专攻肺部疾病。你决定建立一个胸部疾病诊所，主治肺病及相关疾病。大学课本已经中告诉你肺癌、肺结核和支气管炎的发生比率以及这些疾病典型的临床症状、病因等，于是可以根据课本里的理论知识建立自己的贝叶斯网
- 根据如下数据信息：
  - 美国有30%的人吸烟
  - 每10万人中就就有70人患有肺癌
  - 每10万人中就就有10人患有肺结核
  - 每10万人中就就有800人患有支气管炎
  - 10%人存在呼吸困难症状，大部分人是哮喘、支气管炎和其他非肺结核、非肺癌性疾病引起

应用举例：医学数据分析

- 医学数据分析
  - 建立贝叶斯网模型



应用举例：医学数据分析

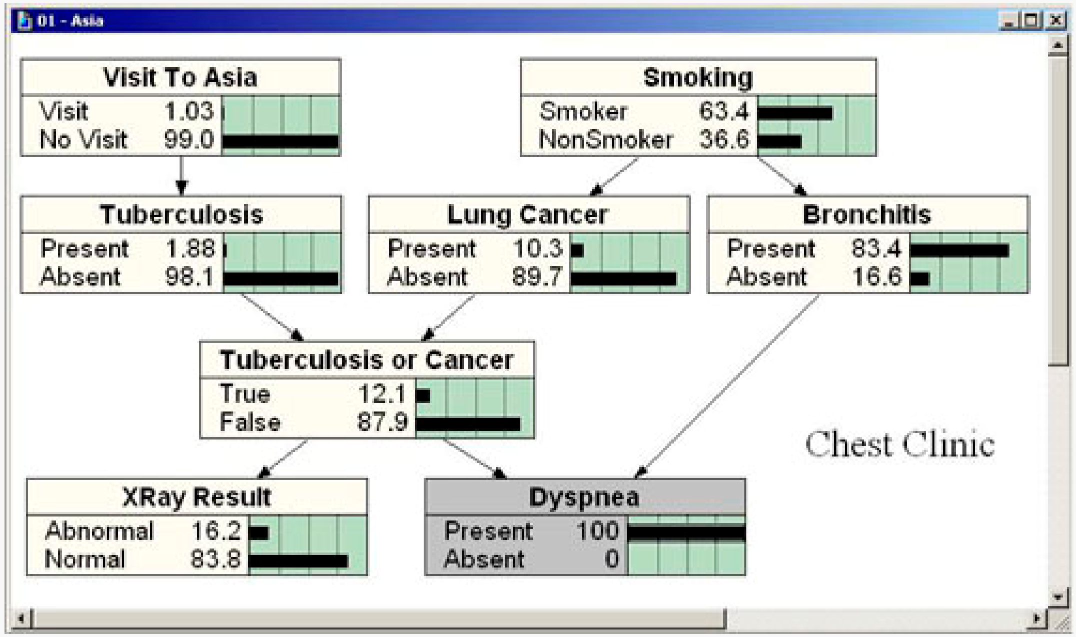
医学数据分析

- 上述模型反映了一个来诊所求医的新患者，为诊断之前我们没有这个患者的任何信息。而当我们向患者咨询信息时，贝叶斯网中的概率就会自动调整，这就是贝叶斯推理最完美和最强大之处
- 贝叶斯网络最强大之处在于**从每个阶段结果所获得的概率都是数学与科学的反映**。换句话说，假设我们了解了患者的足够信息，根据这些信息获得统计知识，网络就会告诉我们合理的推断

应用举例：医学数据分析

□ 医学数据分析

- 现在看看如何增加个别病人信息调节概率。一个女病人进入诊所，我们开始和她谈论。她告诉我们她呼吸困难。我们将这个信息输入到网络。我们相信病人的信息，认为其存在100%呼吸困难



应用举例：医学数据分析

医学数据分析

- 可以观察到，一旦病人有呼吸困难症状，三种疾病的概率都增大了，因为这些疾病都有呼吸困难的症状。我们的病人存在这样的症状，某种程度上我们会推断这三种疾病可能性比较大，也增加了我们患者有严重疾病认识的信念
- 明显增大的是支气管炎，从45%到83.4%. 为什么会有如此大的增长呢？因为支气管炎病比癌症和肺结核更常见。只要我们相信患者有严重的肺部疾病，那最支气管炎的可能性会更大些
- 病人是抽烟者的几率也会随之增大，从50%到63.4%
- X光照片不正常的几率也会上涨，从11%到16%

应用举例：卫星故障预测

卫星故障预测

序号	属性1	属性2	属性3	属性4	属性5	属性6	属性7	属性8	属性9	属性10	状态
1	-0.035	-0.035	-0.027	-0.355	0.011	-0.415	-0.929	1.836	1.619	1.055	正常
2	-0.035	-0.134	-0.027	-0.339	0.011	0.072	-2.215	-0.396	-0.731	-0.233	正常
3	-0.035	-0.134	-0.027	0.687	0.011	-0.017	1.735	1.322	1.282	0.944	正常
4	-0.035	-0.134	-0.027	0.699	0.011	0.108	0.934	1.322	1.192	0.944	正常
5	-0.035	-0.035	-0.027	-0.344	0.011	-0.138	0.428	1.322	1.668	1.217	正常
6	-0.035	0.064	-0.027	-0.073	0.011	0.042	-0.005	-0.957	-1.315	-1.234	异常
7	-0.035	0.064	-0.027	-0.104	0.011	-0.048	-0.278	-0.957	-1.315	-1.399	异常
8	-0.035	0.064	-0.027	0.279	0.011	0.067	-0.555	0.170	-0.141	-0.395	异常
9	-0.035	0.064	-0.027	0.077	0.011	0.353	0.040	0.972	0.714	0.279	异常
10	-0.035	0.037	-0.027	-0.011	0.011	-0.011	-0.184	-0.396	-0.731	-0.626	异常

课程大作业

一、任务介绍

二、评分标准

任务介绍

- 数据集：PASCAL VOC2012数据集的子集（1713张图片）
  - 提供RGB图片、检测框和框对应的类别信息
  - 2007_000559至2012_001051作为训练集，其余作为验证集
- 基础任务（80%）：目标检测
  - 预测输入图片中的物体位置（框）和类别
  - 评价指标：验证集的 mAP(mean average precision)

- 进阶任务（20%）：基于检测框的弱监督实例分割
  - 仅利用检测框作为监督信号，完成实例分割任务
  - 预测输入图片中每个实例的像素级mask
  - 评价指标：验证集的 mask AP
  - 参考：BoxInst: High-Performance Instance Segmentation with Box Annotations

评分标准

- 满分20分
  - 性能5分：性能满足要求、合理即可得满分，不必纠结于传统/深度学习方法
  - 代码10分：需要能够跑通并复现出提交的结果，需要自行实现的部分会查重
  - 报告5分：需要涵盖各部分代码的功能、代码运行过程、方案描述、分析等
  - 若自行提出了创新点且证明性能优越，可酌情加分（总分不超过20分）
  - 不看重绝对的检测率，方案合理、代码整洁有条理、报告完整有逻辑即可
  - 报告字数控制在3000字以内
- 抽查部分同学考核
  - 代码与方案细节进行提问
  - 现场测试代码以及准确率
  - 对任务设计方案理解程度