第七章 贝叶斯分类器

1. **贝叶斯决策论**
   - 贝叶斯决策论是概率框架下实施决策的基本方法。对分类任务而言，在所有相关概率都已知的理想情形下，贝叶斯决策论考虑如何根据这些概率和误判损失来选择最优的类别标记
   - 假设有N种可能的类别标记，即$\mathcal{Y}=\{c_1,c_2,\cdots,c_N\}$，$\lambda_x$是将一个真是标记为$c_j$的样本误分类为$c_i$所产生的损失。基于后验概率$P(c_i|x)$可获得样本x分类为$c_i$所产生的期望损失，即样本x上的条件风险

$$R(c_i \mid \boldsymbol{x})=\sum_{j=1}^{N} \lambda_{ij} P(c_j \mid \boldsymbol{x}) \ . \tag{7.1}$$

     > 当我们知道样本的真实标记时，就能获得将其分类为c_i所产生的损失。此时真实标记未知，所求的条件风险$R(c_i|x)$是将样本x分类为c_i所产生的期望损失，即所有损失的平均值。因此根据期望的定义，即为每个损失乘以其对应的概率，在进行求和即可

   - 我们的任务是寻找一个判定准则$h:\mathcal{X}\mapsto\mathcal{Y}$以最小化总体风险

$$R\left(h\right)=\mathbb{E}_{\boldsymbol{x}}\left[R\left(h\left(\boldsymbol{x}\right) \mid \boldsymbol{x}\right)\right] \ . \tag{7.2}$$

     > $$R(h)=\mathbb{E}_{\boldsymbol{x}}\left[R(h(\boldsymbol{x}) \mid \boldsymbol{x})\right]=\sum_{\boldsymbol{x} \in D} R(h(\boldsymbol{x}) \mid \boldsymbol{x}) P(\boldsymbol{x})$$
     >
     > 即全体样本x的条件风险的期望

   - 显然，对于每个样本x，若h能最小化风险$R(h(x)|x)$，则总体风险$R(h)$也将被最小化，这就产生了贝叶斯判定准则：为最小化总体风险，只需在每个样本上选择那个能使条件风险$R(c|x)$最小的类别标记，即：

$$h^{*}(\boldsymbol{x})=\underset{c \in \mathcal{Y}}{\arg \min } R(c \mid \boldsymbol{x}) \ , \tag{7.3}$$

   - 若目标是最小化分类错误率，则误判损失λ_i可写为：

$$\lambda_{ij}=\begin{cases} 0, & \text{if } i=j \ ; \\ 1, & \text{otherwise,} \end{cases} \tag{7.4}$$

   - 此时条件风险为

$$R(c \mid \boldsymbol{x})=1-P(c \mid \boldsymbol{x}) \ , \tag{7.5}$$

$$
\begin{aligned}
R(c_i \mid \boldsymbol{x}) &= \sum_{j=1}^{N} \lambda_{ij} P(c_j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{i-1} P(c_j \mid \boldsymbol{x}) + \sum_{j=i+1}^{N} P(c_j \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{i-1} P(c_j \mid \boldsymbol{x}) + P(c_i \mid \boldsymbol{x}) + \sum_{j=i+1}^{N} P(c_j \mid \boldsymbol{x}) - P(c_i \mid \boldsymbol{x}) \\
&= \sum_{j=1}^{N} P(c_j \mid \boldsymbol{x}) - P(c_i \mid \boldsymbol{x}) \\
&= 1 - P(c_i \mid \boldsymbol{x})
\end{aligned}
$$

- 于是，最小化分类错误率的贝叶斯最优分类器为：

$$
h^*(\boldsymbol{x}) = \underset{c \in \mathcal{Y}}{\arg\max}\, P(c \mid \boldsymbol{x}) \ , \tag{7.6}
$$

- 即对每个样本x选择能使后验概率$P(c|x)$最大的类别标记
- 不难看出，欲使用贝叶斯判定准则来最小化决策风险，首先要获得后验概率。然而在限时任务中通常难以直接获取。从这个角度来看，机器学习所要实现的是基于有限的训练样本集尽可能准确的估计出后验概率。大体来说，主要分为两种策略
  1. 判别式模型：给定x，通过直接建模$P(c|x)$来预测c。前面介绍的决策树、BP神经网络、支持向量机等都可以归为判别式模型的范畴
  2. 生成式模型：先对联合概率分布$P(x,c)$建模，然后在由此获取$P(c|x)$。对于生成式模型，必然需要考虑

$$
P(c \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})} \ . \tag{7.7}
$$

  基于贝叶斯定理，也可写成

$$
P(c \mid \boldsymbol{x}) = \frac{P(c)\ P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})} \ , \tag{7.8}
$$

  这里，$P(c)$是先验概率，$P(x|c)$是样本x相对于类标c的类条件概率，或称为似然。$P(x)$是用于归一化的证据因子。对于给定样本x，证据因子$P(x)$与类标记无关，因此估计$P(c|x)$的问题就转化为如何基于训练数据来估计先验$P(c)$和似然$P(x|c)$

  > 这里对生成式模型进行进一步理解
  >
  > - 数据集按照联合概率分布$P(x,c)$采样而得，在样本已知的情况下，选择与x联合概率最大的类别标记即可
  > - 根据公式：$P(x,c) = P(c|x)p(x)$，给定样本x，概率$P(x)$为常数，因此最大化$P(x,c)$和$\frac{P(x,c)}{P(x)}$等价。因此最大化后验概率$P(c|x)$和最大化联合概率$P(x,c)$等下。这可进一步说明判别式模型和生成式模型的等价性。

  - 针对先验概率$P(c)$表达了样本空间中各类样本所占比例，根据大数定律，当训练集包含充足的独立同分布样本时，$P(c)$可通过各类样本出现的频率来进行估计。
  - 对于似然$P(x|c)$来说，由于它涉及关于x所有属性的联合概率，直接根据样本出现的概率来估计会遇到严重的困难，样本空间往往远大于训练样本数m，也就是说很多样本取值在训练集中根本没有出现，直接使用频率估计$P(x|c)$显然不行，因为未被观测到与出现概率为零通常是不同的。

2. **极大似然估计**

- 估计类条件概率的一种常用策略是先假定其具有某种确定的概率分布形式，在基于训练样本对概率分布的参数进行估计。
- 事实上，概率模型的训练过程就是参数估计过程，对于参数估计，统计学界的两个学派分别提供了不同的解决方案：频率主义学派认为参数虽然未知，但却是客观存在的固定值，因此可以通过优化似然函数等准则来确定参数值；而贝叶斯学派则认为参数是未观察到的随机变量，其本身也有分布，因此，可假定参数服从一个先验分布，然后基于观测到的数据来计算参数的后验分布。
- 令D_c为训练集中第c类样本组成的集合，假设这些样本是独立同分布的，则参数θ_c对于数据集D_c的似然是：

$$P(D_c \mid \boldsymbol{\theta}_c) = \prod_{\boldsymbol{x} \in D_c} P(\boldsymbol{x} \mid \boldsymbol{\theta}_c) \ . \tag{7.9}$$

对θ_c进行极大似然估计，就是去寻找最大化似然$P(D_c|\theta_c)$的参数$\hat{\theta}_c$。直观上来看，极大似然估计就是试图在θ_c所有可能的取值中，找到一个能使数据出现的可能性最大的值

- 上式的连乘操作容易造成下溢，通常使用对数似然

$$\begin{aligned} LL(\boldsymbol{\theta}_c) &= \log \ P(D_c \mid \boldsymbol{\theta}_c) \\ &= \sum_{\boldsymbol{x} \in D_c} \log P(\boldsymbol{x} \mid \boldsymbol{\theta}_c) \ , \end{aligned} \tag{7.10}$$

此时参数θ_c的极大似然估计：

$$\hat{\boldsymbol{\theta}}_c = \arg\max_{\boldsymbol{\theta}_c} \ LL(\boldsymbol{\theta}_c) \ . \tag{7.11}$$

3. 朴素贝叶斯分类器

3.1 朴素贝叶斯分类器

- 在第一节的讨论中，不难发现基于贝叶斯公式（7.8）来估计后验概率$P(c|x)$的主要困难在于类条件概率$P(x|c)$是所有属性上的联合概率，难以从有限的训练样本直接估计而得。为了避开这个障碍，朴素贝叶斯分类器采用了“属性条件独立性假设”：对已知类别，假设所有属性相互独立。
- 基于属性条件独立性假设，（7.8）可重写为

$$P(c \mid \boldsymbol{x}) = \frac{P(c) \ P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})} = \frac{P(c)}{P(\boldsymbol{x})} \prod_{i=1}^{d} P(x_i \mid c) \ , \tag{7.14}$$

对于所有类别来说$P(x)$相同，因此，基于式（7.6）的贝叶斯判定准则有：

$$h_{nb}(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} \ P(c) \prod_{i=1}^{a} P(x_i \mid c) \ , \tag{7.15}$$

上式即为朴素贝叶斯分类器的表达式

- 显然，朴素贝叶斯分类器的训练过程就是基于训练集D来估计勒先验概率$P(c)$，并为每个属性估计条件概率$P(x_i|c)$
- 令D_c为训练集D中第c类样本组成的集合，D_{c,x_i}表示D_c中在第i个属性上取值为x_i的样本组成的集合，则有：

$$P(c) = \frac{|D_c|}{|D|} . \tag{7.16}$$

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|} . \tag{7.17}$$

- 对于连续属性可考虑概率密度函数，假定$p(x_i|c) \sim \mathcal{N}(\mu_{c,i}, \sigma^2_{c,i})$，其中$\mu_{c,i}, \sigma^2_{c,i}$分别表示第c类样本在第i个属性上取值的均值和方差：

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma^2_{c,i}}\right) . \tag{7.18}$$

3.2 平滑

- 注意到，若某个属性值在训练集中没有与某个类别同时出现过，则直接基于式（7.17）进行概率计算，在进行判别会出现问题。由于在（7.15）上的连乘式计算为0，因此，无论该样本的其他属性是什么，分类结果为“怀瓜”。这显然不合理。
- 为了避免其他属性携带的信息被训练集中未出现的属性值抹去，在估计概率值时通常要进行“平滑”，常用拉普拉斯修正。令N表示训练集D中可能的类别是，N_i表示第i个属性可能的取值数，即如下所示

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N} , \tag{7.19}$$

$$\hat{P}(x_i \mid c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i} . \tag{7.20}$$

- 显然，拉普拉斯修正避免了因训练样本不充分而导致概率估计为零的问题，并且在训练集变大时，修正过程所引入的先验的影响也会逐渐变得可忽略，使得估值趋向于实际概率值。

4. **半朴素贝叶斯分类器**

4.1 条件互信息

- 互信息： 指的是两个随机变量之间的相关程度。 $I(X;Y) = H(X) - H(X|Y)$

  确定随机变量Y的值后，另一个随机变量X不确定性的削弱程度，因而互信息取值最小为0，意味着给定一个随机变量对确定一另一个随机变量没有关系，最大取值为随机变量的熵，意味着给定一个随机变量，能完全消除另一个随机变量的不确定性。这个概念和条件熵相对。

  where $\mathrm{H}(X)$ and $\mathrm{H}(Y)$ are the marginal entropies, $\mathrm{H}(X|Y)$ and $\mathrm{H}(Y|X)$ are the conditional entropies, and $\mathrm{H}(X, Y)$ is the joint entropy of X and Y. Note the analogy to the union, difference, and intersection of two sets, as illustrated in the Venn diagram. Because $I(X;Y)$ is non-negative, consequently, $\mathrm{H}(X) \geq \mathrm{H}(X|Y)$. Here we give the detailed deduction of $I(X;Y) = \mathrm{H}(Y) - \mathrm{H}(Y|X)$:

$$\begin{aligned}
I(X;Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x,y} p(x)p(y|x) \log p(y|x) - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x} p(x) \left(\sum_{y} p(y|x) \log p(y|x) \right) \\
&\quad - \sum_{y} \log p(y) \left(\sum_{x} p(x,y) \right) \\
&= -\sum_{x} p(x) \mathrm{H}(Y|X=x) - \sum_{y} \log p(y) p(y) \\
&= -\mathrm{H}(Y|X) + \mathrm{H}(Y) \\
&= \mathrm{H}(Y) - \mathrm{H}(Y|X).
\end{aligned}$$

- 条件互信息：$I(X;Y|Z) = H(X|Z) - H(X|Y;Z)$

4.2 半朴素贝叶斯分类器

- 为了降低贝叶斯公式中估计后验概率$P(c|x)$的困难，朴素贝叶斯分类器采用了属性条件独立性假设，但在现实生活中很难成立，人们尝试对属性独立性假设进行一定程度的放松，由此产生了一类"半朴素贝叶斯分类器"的学习方法
- 半朴素贝叶斯分类器的基本思想是适当考虑一部分属性间的相互依赖信息，从而既不需要进行完全联合概率计算，又不至于彻底忽略比较强的属性依赖关系。"独依赖（ODE）"是半朴素贝叶斯分类器最常用的一种策略。即假设每个属性在类别之外最多仅依赖于一个其他属性

$$P(c \mid \boldsymbol{x}) \propto P(c) \prod_{i=1}^{d} P(x_i \mid c, pa_i) \ , \tag{7.21}$$

其中pa_i为属性x_i所依赖的属性，称为x_i的父属性。于是，问题的关键转化为如何确定每个属性的父属性，不同的做法产生不同的独依赖分类器

1. SPODE： 假设所有属性都依赖于同一个属性，称为超父，然后通过交叉验证等模型选择方法来确定超父属性
2. TAN则是在最大带全生成树算法的基础上，通过以下步骤将属性间的依赖关系简化为树形结构

(1) 计算任意两个属性之间的条件互信息(conditional mutual information)

$$I(x_i, x_j \mid y) = \sum_{x_i, x_j;\ c \in \mathcal{Y}} P(x_i, x_j \mid c) \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c) P(x_j \mid c)} \ ; \tag{7.22}$$

(2) 以属性为结点构建完全图, 任意两个结点之间边的权重设为$I(x_i, x_j \mid y)$;

(3) 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;

(4) 加入类别结点 y, 增加从 y 到每个属性的有向边.

很容易看出，条件互信息刻画了属性x_i和x_j在已知类别情况下的相关系。因此通过最大生成树算法，TAN实际上保留了强相关属性之间的依赖性。

3. AODE是一种基于集成学习机制、更为强大的独依赖分类器，与SPODE通过模型选择确定超父属性不同，AODE尝试每个属性作为超父来构建SPODE，然后将具有足够训练数据支撑的集成起来作为最终结果：

$$P(c \mid \boldsymbol{x}) \propto \sum_{\substack{i=1 \\ |D_{x_i}| \geqslant m'}}^{d} P(c, x_i) \prod_{j=1}^{d} P(x_j \mid c, x_i) \ , \tag{7.23}$$

其中D_{x_i}是在第i个属性上取值为x_i的样本的集合，m'为阈值常数。

$$
\begin{aligned}
P(c|\boldsymbol{x}) &= \frac{P(\boldsymbol{x},c)}{P(\boldsymbol{x})} \\
&= \frac{P\left(x_1,x_2,\ldots,x_d,c\right)}{P(\boldsymbol{x})} \\
&= \frac{P\left(x_1,x_2,\ldots,x_d|c\right)P(c)}{P(\boldsymbol{x})} \\
&= \frac{P\left(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_d|c,x_i\right)P\left(c,x_i\right)}{P(\boldsymbol{x})}
\end{aligned}
$$

$$
\begin{aligned}
P(c|\boldsymbol{x}) &\propto P(c,x_i)P(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_d|c,x_i) \\
&= P(c,x_i)\prod_{j=1}^{d}P(x_j|c,x_i)
\end{aligned}
$$

$$
P(c|\boldsymbol{x}) \propto \sum_{\substack{i=1 \\ |D_{x_i}|\geq m'}}^{d} P(c,x_i)\prod_{j=1}^{d}P(x_j|c,x_i)
$$

为什么$D_{x_i} \geq m'$?

下面用到了D_{c,x_i}, D_{c,x_i,x_j}，如果D_{x_i}过小，D_{c,x_i}, D_{c,x_i,x_j}会更小

显然AODE需要估计$P(c,x_i)$和$P(x_j|c,x_i)$

$$
\hat{P}(c,x_i) = \frac{|D_{c,x_i}|+1}{|D|+N_i}\ , \tag{7.24}
$$

$$
\hat{P}(x_j \mid c,x_i) = \frac{|D_{c,x_i,x_j}|+1}{|D_{c,x_i}|+N_j}\ , \tag{7.25}
$$

这里N_i是第i个属性可能的取值数，N_j是第j个属性可能的取值数

5. **贝叶斯网**

贝叶斯网亦称信念网，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。

具体来说, 一个贝叶斯网 B 由结构 G 和参数 Θ 两部分构成, 即 $B = \langle G, \Theta \rangle$. 网络结构 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, 则它们由一条边连接起来; 参数 Θ 定量描述这种依赖关系, 假设属性 x_i 在 G 中的父结点集为 π_i, 则 Θ 包含了每个属性的条件概率表 $\theta_{x_i|\pi_i} = P_B(x_i \mid \pi_i)$.

作为一个例子, 图 7.2 给出了西瓜问题的一种贝叶斯网结构和属性“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”量化依赖关系, 如 $P(\text{根蒂} = \text{硬挺} \mid \text{甜度} = \text{高}) = 0.1$ 等.

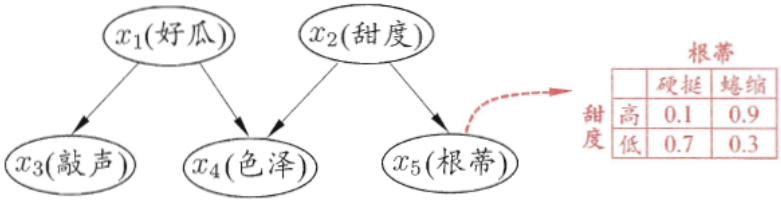

图 7.2 西瓜问题的一种贝叶斯网结构以及属性“根蒂”的条件概率表

5.1 结构

贝叶斯网结构有效的表达了属性间条件独立性。给定父节点数据集，贝叶斯网假设每个属性与他的非后裔属性独立，于是$B = \langle G, \Theta \rangle$将属性$x_1, x_2, \cdots, x_d$的联合概率分布定义为：

$$P_B(x_1, x_2, \ldots, x_d) = \prod_{i=1}^{d} P_B(x_i \mid \pi_i) = \prod_{i=1}^{d} \theta_{x_i|\pi_i} \,. \tag{7.26}$$

以图 7.2 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，
- 下图显示出贝叶斯网中三种变量之间的典型依赖关系

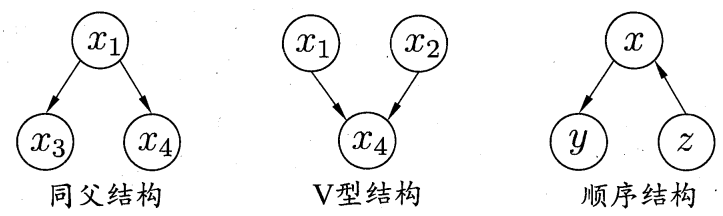

图 7.3 贝叶斯网中三个变量之间的典型依赖关系

在同父结构中，给定父节点x_1的取值，则x_3和x_4条件独立。在顺序结构中，给定x的值，则y和z条件独立。V型结构亦称“冲撞结构”，给定子结点x_4的取值，x_1和x_2必不独立；奇妙的是，若x_4的取值未知，则V型结构下x_1和x_2却是相互独立的

$$
\begin{aligned}
P(x_1, x_2) &= \sum_{x_4} P(x_1, x_2, x_4) \\
&= \sum_{x_4} P(x_4 \mid x_1, x_2) P(x_1) P(x_2) \\
&= P(x_1) P(x_2) \ . \qquad (7.27)
\end{aligned}
$$

这样的独立性称为边际独立性。记作：$x_1 \perp\!\!\!\perp x_2$.

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. 例如在同父结构中, 条件独立性 $x_3 \perp x_4 \mid x_1$ 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, $y \perp z \mid x$, 但 $y \perp\!\!\!\perp z$ 不成立.

- 为了分析有向图中变量间的条件独立性，可使用“有向分离”，我们先把有向图变成一个无向图
  - 找出有向图中所有V型结构，在V型结构的两个父结点之间加上一个无向边
  - 将所有有向边改成无向边
- 由此产生的无向图称为“道德图”，令父节点相连的过程称为“道德化”
- 假定道德图中有变量x，y和变量集合$\{z_i\}$，若变量x和y能在图上被$\mathbf{z}$分开，即从道德图中将变量集合$\mathbf{z}$去除后，x和y分属两个连通分支，则称变量x和y被$\mathbf{z}$有向分离，即$x \perp y|\mathbf{z}$

5.2 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。但在现实应用中我们往往并不知晓网络结构。贝叶斯网络的首要任务就是根据训练数据集找出结构最“恰当”的贝叶斯网。
- “评分搜索”是求解这一问题的常用方法。即我们定义一个评分函数，以此来评估贝叶斯网与训练数据的契合程度，然后基于这个评价函数来寻找结构最优的贝叶斯网络。
- 常用评分函数通常基于信息论准则，此类准则将学习问题看作一个数据压缩任务，学习的目标在于找到一个能以最短编码长度描述训练数据的模型。此时的编码长度包括模型自身所需要的字节长度和该模型描述数据所需的字节长度。对于贝叶斯学习来说，模型就是一个贝叶斯网，同时每个贝叶斯网描述了一个在训练数据上的概率分布，自有一套编码机制能使那些经常出现的样本有更短的编码。于是，我们应选择综合编码长度最短的贝叶斯网，这就是最小描述长度准则。

  给定训练集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, 贝叶斯网 $B = \langle G, \Theta \rangle$ 在 D 上的评分函数可写为

  $$
  s(B \mid D) = f(\theta)|B| - LL(B \mid D) \ , \qquad (7.28)
  $$

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

  $$
  LL(B \mid D) = \sum_{i=1}^{m} \log P_B \left(\boldsymbol{x}_i\right) \qquad (7.29)
  $$

  是贝叶斯网B的对数似然，显然第一项计算编码贝叶斯网B所需的字节数，第二项是计算B对于的概率分布P_B需多少字节来描述D。
- 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，学习任务退化为极大似然估计。

- 不然发现，若贝叶斯网$B=\langle G,\Theta\rangle$的网络结构G固定，则评分网络的第一项为常数，此时，最小化评分函数等价于对参数Θ的极大似然估计。
- 不幸的是，从所有可能的网络结构看见搜索最优贝叶斯网结构是一个NP难问题，有常用两种策略能在有限时间内求得近似解： 贪心法—从某个结构出发，每次调整一条边，直到评分函数不再降低；给网络结构施加约束来减小搜索空间。

5.3 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值。这样通过已知变量观测者来推测待查询变量的过程称为推断，已知变量观测值称为证据。
- 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，不幸的是，这样的精确推断已被证明是NP难的。当网络结点较多、连接稠密时，难以进行精确推断，此时需借助近似推断，降低精度要求，在有限时间内求得近似解。现实应用中，贝叶斯网的近似推断常使用吉布斯采样来完成。

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

6. **EM算法**（详细详解见：统计学习方法第九章）

- 前面的讨论中，我们一直假设样本所有属性变量的值都已被检测到，即训练样本是完整的。但在现实应用中往往会遇到不完整的训练样本，即训练样本的属性变量值未知。在这种存在未观测变量的情形下，是否仍能对模型参数进行估计呢？
- 未观测变量的学名是"隐变量"，令X表示已观测变量集，Z表示隐变量集，Θ表示模型参数。若欲对Θ做极大似然估计，则应最大化对数似然

$$LL(\Theta\mid\mathbf{X},\mathbf{Z})=\ln P(\mathbf{X},\mathbf{Z}\mid\Theta)\ . \tag{7.34}$$

- 然而由于Z是隐变量，上式无法直接求解，此时我们可通过对Z计算期望，来最大化已观测数据的对数"边际似然"：

$$LL(\Theta\mid\mathbf{X})=\ln P(\mathbf{X}\mid\Theta)=\ln\sum\nolimits_{\mathbf{Z}}P(\mathbf{X},\mathbf{Z}\mid\Theta)\ . \tag{7.35}$$

- EM算法是常用估计参数隐变量的利器，它是一种迭代式的方法，其基本思想是：若参数Θ已知，则可根据训练数据推断出最优隐变量Z的值（E步）；反之，若Z的值已知，则可方便对参数Θ做极大似然估计(M步)

  于是, 以初始值 Θ^0 为起点, 对式(7.35), 可迭代执行以下步骤直至收敛:

  - 基于 Θ^t 推断隐变量 $\mathbf{Z}$ 的期望, 记为 $\mathbf{Z}^t$;
  - 基于已观测变量 $\mathbf{X}$ 和 $\mathbf{Z}^t$ 对参数 Θ 做极大似然估计, 记为 Θ^{t+1};

  这是EM算法的原型

- 进一步，若我们不是取Z的期望，而是基于Θ^t计算隐变量Z的概率分布$P(Z|X,\Theta^t)$，则EM算法的两个步骤是：

- - **E** 步 (Expectation): 以当前参数 Θ^t 推断隐变量分布 $P(\mathbf{Z} \mid \mathbf{X}, \Theta^t)$, 并计算对数似然 $LL(\Theta \mid \mathbf{X}, \mathbf{Z})$ 关于 $\mathbf{Z}$ 的期望

$$Q(\Theta \mid \Theta^t) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\Theta^t}\, LL(\Theta \mid \mathbf{X}, \mathbf{Z}) \ . \tag{7.36}$$

  - **M** 步 (Maximization): 寻找参数最大化期望似然, 即

$$\Theta^{t+1} = \arg\max_{\Theta}\ Q(\Theta \mid \Theta^t) \ . \tag{7.37}$$

  > 统计学习方法中对Q函数定义更为清晰,如下
  >
  > $$\begin{aligned} Q(\theta, \theta^{(i)}) &= E_Z[\log P(Y, Z \mid \theta) \mid Y, \theta^{(i)}] \\ &= \sum_Z \log P(Y, Z \mid \theta) P(Z \mid Y, \theta^{(i)}) \end{aligned}$$

  简要来说,EM算法使用两个步骤交替计算,第一步是期望(E)步,利用当前估计的参数值来计算对数似然的期望;第二步是最大化(M)步,寻找能使E步产生的似然期望最大化的参数值。然后，新得到的参数值重新被用于E步，直至收敛到局部最优解。

- 事实上，隐变量估计问题也可通过梯度下降法等优化算法求解，但由于求和的项数将随着隐变量的数目以指数级上升，会给梯度计算带来麻烦，而EM算法则可看作一种非梯度优化方法