第5-2节　贝叶斯网络理论及方法

—— 网络结构学习(2)

5.4 基于得分的结构学习方法

- 基于得分的方法把结构学习问题作为一个优化问题来解决。
- 定义一个得分函数，根据训练数据对每个候选结构打分，然后搜索得到一个得分高的结构。

5.4.1 似然得分

5.4.1.1 似然评分

给定变量集合 $X=\{X_1, X_2, \ldots, X_N\}$，将模型结构 S 和模型参数 θ_S 定义为随机变量，构造基于上述两个参数的最大似然函数：

$$L(S|D) = \max_{l=1,2,\ldots h} \prod_{i=1}^{N} p(X_i|S_l, \theta_S)$$

学习目的就是从 h 个候选结构中，选择具有最大似然值的那个。用似然得分来表示：

$$score_L(S; D) = L(S|D)$$

5.4.1.2 似然评分与互信息的关系

考虑有两个独立变量 X 和 Y 的贝叶斯网结构 D_0，其似然得分为：

$$score(S_0|D) = \sum \log \hat{\theta}_X + \log \hat{\theta}_Y$$

而当存在 $X \rightarrow Y$ 的依赖关系时，其贝叶斯网结构 S_1 的似然得分为：

$$score(S_1|D) = \sum \log \hat{\theta}_X + \log \hat{\theta}_{Y|X}$$

两个得分之间的差异可以写为：

$$score(S_1|D) - score(S_0|D) = \sum \hat{P}(X,Y) \log \frac{\hat{P}(Y|X)}{\hat{P}(Y)} = I_{\hat{P}}(X;Y)$$

即存在 $X \rightarrow Y$ 依赖关系的结构 S_1 比 X, Y 相互独立的结构 S_0 的似然得分要高一个互信息的值。

5.4.1.3 最大似然得分的局限性

由于互信息 $I_P(X;Y) \geq 0$，则

$$score(S_1|D) - score(S_0|D) \geq 0$$

即似然得分给复杂结构的得分会高于简单结构的，它更偏好学习得到更复杂的网络结构，所以它倾向于支持增加节点互信息量的操作(比如加边)。

5.4.2 贝叶斯得分

5.4.2.1 边缘似然函数

将模型结构 S 和模型参数 θ_S 定义为随机变量，由贝叶斯公式可得：

$$p(S \mid D) = \frac{p(S) \cdot p(D \mid S)}{p(D)}$$

学习目的就是选择使后验概率 $p(S \mid D)$ 最大的网络结构。$p(S)$ 称为结构先验分布，是关于结构 S 的先验知识的概括，$p(D \mid S)$ 称为结构似然函数。

由于分母 $p(D)$ 不依赖于结构 S，所以选择后验概率最大的结构就是选择如下函数达到最大的结构：

$$\log p(S,D) = \log p(D \mid S) + \log p(S)$$

$\log p(S,D)$ 称为结构 S 的贝叶斯评分。一般结构先验分布 $p(S)$ 取均匀分布。$p(D \mid S)$ 称为边缘似然函数，记为 $L(S \mid D)$。于是确定网络结构的后验分布只需要为每一个可能的结构计算数据的边缘似然。

5.4.2.2 边缘似然与最大似然得分的区别

考虑到边缘似然 $p(D|S)$ 可以表示为：

$$p(D|S) = \int p(D|\theta_S, S) \cdot p(\theta_S|S)\, d\theta_S$$

上式表明：边缘似然 $p(D|S)$ 是结构 S 和参数 θ_S 构造的似然函数的平均值，而最大似然得分返回的是这个函数的最大值。

由于贝叶斯得分的似然函数是在参数 θ_S的各个不同取值上积分，相当于在参数 θ_S的各个不同取值上求平均，因此它可以克服最大似然得分的过拟合问题，泛化能力更强，可以学习得到更简单的网络结构。

5.4.2.3 贝叶斯网的贝叶斯得分

假设由两个变量 X, Y 构造的贝叶斯网，它们之间的互信息量大小会影响边缘似然的值，进而取不同的贝叶斯得分，关系如图5.1所示。

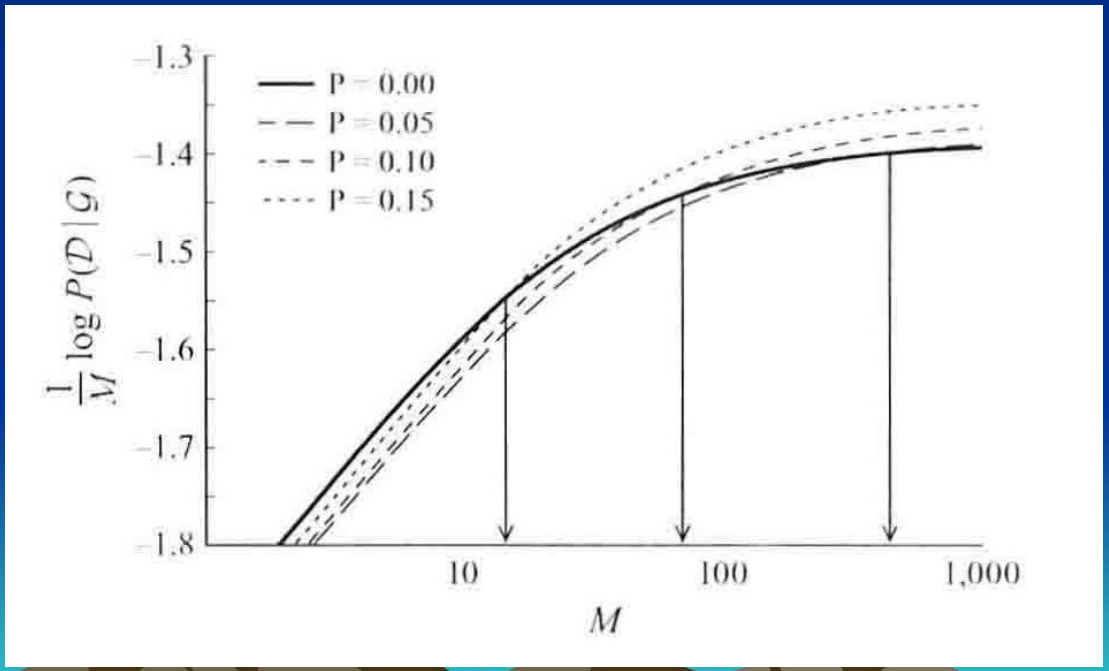


图5.1 互相关对贝叶斯得分的影响

从图 5.1可以看出，当样本数据量较大时，贝叶斯得分偏向于 X 与 Y 相互依赖的结构 $S_{X\leftrightarrow Y}$，当这种依赖性较强时($p=0.15$)，得分提高会更快速；

当样本数量较小时，对于本质上独立的 X 和 Y 变量来说，采样噪声带来的小幅波动不会影响简单结构的贝叶斯得分较高的结果，因此可以解决小样本下最大似然得分过拟合的问题。

5.4.2.4 几种常见的贝叶斯评分标准

在无约束多项分布、参数独立、采用Dirichlet先验和数据完整的前提下，数据的边缘似然正好等于每对(i, j) 的边界似然的乘积，即

$$L(S \mid D)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij.})}{\Gamma(\alpha_{ij.}+m_{ij.})}\cdot\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+m_{ijk})}{\Gamma(\alpha_{ijk})}$$

其中：m_{ijk}是 D 中满足 $x_i=k, \pi(x_i)=j$ 的样本个数，而

$$m_{ij*}=\sum_{k=1}^{r_i}m_{ijk} \qquad \alpha_{ij*}=\sum_{k=1}^{r_i}\alpha_{ijk}$$

1) CH评分标准

对上式两边取对数，得：

$$l(S \mid D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\log \frac{\Gamma(\alpha_{ij\cdot})}{\Gamma(\alpha_{ij\cdot} + m_{ij\cdot})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})} \right]$$

上式右边所给出的量称为结构 S 的Cooper-Herskovits评分，简称 CH 评分。如果假设结构先验分布是均匀分布，那么用贝叶斯评分选择模型就等于是用CH评分来选择模型。

在使用CH评分之前，首先需要选定先验分布中的超参数α_{ijk}。需要对每一个可能的结构都要提供一个α_{ijk}的先验分布，但是由于候选结构太多无法一一罗列。实际可行的方法是规定一个等价样本量α和一个先验贝叶斯网N_0，然后利用下式得到先验分布中的超参数α_{ijk}：

$$\alpha_{ijk} = \alpha \cdot p_{N_0}(x_i = k \mid \pi(x_i) = j)$$

2) BIC评分标准

- 在大样本前提下，利用拉普拉斯泰勒级数，将对数似然函数在极值点邻域展开，获得其多元正态函数积分的近似表示。在此基础上，度量结构与数据的拟合程度。BIC评分函数如下：

$$\log p(D \mid S) \approx \log p(D \mid S, \theta^*) - \frac{1}{2}\log |A| + \log p(\theta^* \mid S) + \frac{d}{2}\log(2\pi)$$

BIC评分函数的第一项是模型N的对数似然度，度量结构与数据的拟合程度。第二项是模型复杂度的罚项，防止数据与结构的过度拟合。第三、四项与样本数据无关，一般将其略去。

因此，直观上基于BIC评分的就是选择既与数据拟合，又比较简单的模型。

3) 其它几种评分标准

1. MDL评分。最短描述长度的简称，它通过贝叶斯网分析数据中的规律对数据进行压缩，从而降低数据的编码长度。因此，可以通过检测某个结构下分析的数据是否压缩成功，来度量该结构。

2. AIC评分。它寻找一个贝叶斯网结构 N^*，使得 $p_{N^*}(x)$ 与 $p(x)$ 之间的 Kullback-Leibler (KL) 距离最短。KL 距离也叫做相对熵 (Relative Entropy)，它衡量的是相同事件空间里的两个概率分布的差异情况。

3. HVL评分。将数据D随机分成训练样本集D_t和验证样本集D_v。对于某个贝叶斯结构S，首先采用训练样本集D_t对其参数θ进行估计，得到一个贝叶斯网(S, θ^t)，然后计算验证样本集D_v的对数似然度：

$$\mathrm{HVL}(S \mid D_v, D_t) = \log p(D_v \mid S, \theta^t)$$

这称为结构S的验证数据似然度。

4. CVL评分。多次计算结构模型的HVL评分，每次都按不同方式将数据集 D 划分为D_t 和D_v，然后计算各次所得评分的平均值，并将其作为结构 S 的最后评分。它比HVL评分更具鲁棒性，但其计算复杂度也高出HVL评分数倍。在大样本情况下，CVL准则与AIC准则等价。

5.4.3 启发式搜索结构

- 常用的搜索算法是启发式局部搜索算法，这种方法从给定的初始网络结构 (可以是空网络结构、随机指定的网络结构、先验网络结构等) 开始，通过增加、删除和转向操作使得局部最大化，再逐渐扩展到整个网络。
  - K2算法：通过逐渐加边寻找评分高的模型
  - 爬山法：利用搜索算子逐步对模型做局部修改
  - 结构EM算法：当数据有缺失值时对结构和参数同时优化

5.4.3.1 K2算法

- Cooper和Herskovits在1992提出的 K2 算法是最早的贝叶斯网络结构学习算法之一。K2算法的目的是要寻找CH评分高的模型，它用一个变量排序ρ和一个正整数 u 来限制搜索空间：

(1) S 中任一变量的父节点个数不超过u；

(2) ρ是 S 的一个拓扑序。

为简化模型评分的计算，K2假设所有参数先验分布都是均匀分布。这意味着CH评分中的超参数α_{ijk} 以及$\alpha_{ij.}$都是1。

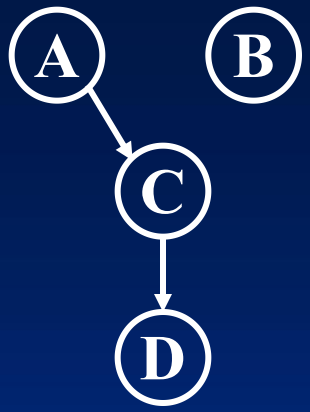
K2算法的出发点是一个包含所有节点，但却没有边的无边图。在搜索过程中，K2按顺序逐个考察 ρ 中的变量，确定其父节点，然后添加相应的边。

对某一节点 x_j，假设它的父节点个数还未达到 u，就继续在 ρ 中为它寻找父节点。将那些排在 x_j 之前，但却还不是 x_j 父节点的节点中选出使得新家族的CH评分达到最大的 x_i，然后与旧家族的CH评分比较：如果 $CH_{new} > CH_{old}$，则把 x_i 添加为 x_j 的父节点；否则停止为 x_j 寻找父节点。

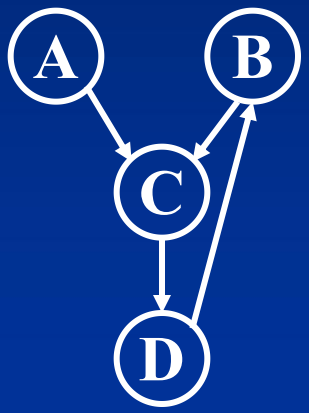
5.4.3.2 爬山法

(1) 从无边模型出发，利用搜索算子 (加边、减边、转边) 对当前模型进行局部修改 (不能形成有向圈)，得到一系列候选模型；

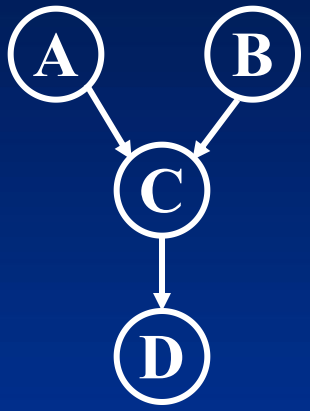
(2) 计算每个候选模型的评分，将其中分数最高的模型的评分与修改前的模型比较，选择评分高的作为当前模型。



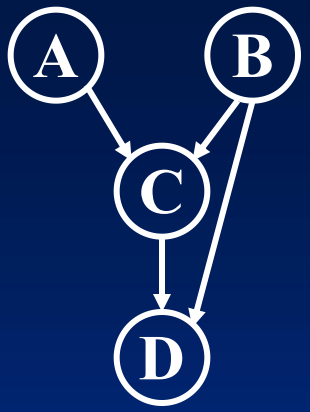
减边$B \rightarrow C$



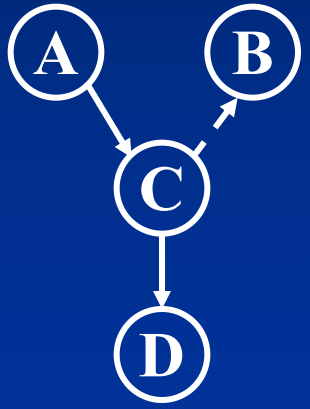
加边$D \rightarrow B$,导致有向圈，不允许



当前模型



加边$B \rightarrow D$



转边$B \rightarrow C$

图5.2 爬山法所用的3个搜索算子

爬山法可以使用任何评分函数，但它有一个缺点：容易陷入局部最优，从而找不到全局最优。

克服此缺点的一个方法是多次运行爬山法，每次都从一个随机产生的新结构开始，最后取各次运行结果中最优的那个作为最后结果，这叫做随机重复爬山法。

5.4.3.3 缺值数据下的结构学习算法SEM

从某初始模型结构和参数出发开始迭代，在进行了 t 次迭代得到了(ξ^t, θ^t)后，第 $t+1$ 次迭代由以下两个步骤组成：

① 基于 (ξ^t, θ^t) 对数据进行修补，使之完整；

② 基于修补后的完整数据对模型及参数进行优化，得到 $(\xi^{t+1}, \theta^{t+1})$。

SEM算法先固定模型结构，进行数次参数优化后，再进行一次结构加参数优化，如此交替进行。