电子科技大学
University of Electronic Science and Technology of China

贝叶斯网中的结构学习

于忠靖

Data Mining Lab, Big Data Research Center, UESTC
Email：junmshao@uestc.edu.cn
http://staff.uestc.edu.cn/shaojunming

概述

数据挖掘实验室
Data Mining Lab

1. 简介
    - 问题定义
    - 方法概述
2. 基于约束的方法
3. 结构评分
    - 似然评分
    - 贝叶斯评分
    - 单变量的边缘似然
    - 贝叶斯网络的贝叶斯评分
    - 先验
4. 结构搜索

简介

数据挖掘实验室 Data Mining Lab

问题定义

目标：仅从给定的数据中学习网络模型（重构网络）。
假设：数据根据未知分布采样，且满足独立同分布；数据能完全被观测到

学习网络结构的原因：
知识发现：探索网络中依赖关系，获得领域中变量的依赖关系。
密度估计：估计潜在分布的统计模型。将得到的网络结构能泛化到新样本上。

方法概述

数据挖掘实验室
Data Mining Lab

无事先指定结构的学习方法：

- ## 基于约束的结构学习

思想：将贝叶斯网看作独立关系的表示，尝试对数据中**条件依赖**和**条件独立**关系进行检验，找到最好解释的某个网络。

方法：给出一个满足**独立关系的分布**，寻找该分布对于的**I-map**。

- ## 基于得分的结构学习

思想：将贝叶斯网视为统计模型，然后把学习作为**模型选择**问题来处理。

方法：考虑可能的**网络结构集合**，并定义测量模型对观测数据拟合程度的**得分函数**，选**最高**者。

基于约束的方法

数据挖掘实验室
Data Mining Lab

总体框架：

找到该领域最好最小I-map

主要技术问题：

如何回答数据中变量间的条件独立性查询。

方法：假设检验（单边假设检验）

如何设计好的决策规则（判定是否拒绝输入）

偏差度量

定义偏离零假设的度量

构造网络

使用Build-PDAG算法

基于约束的方法

单边假设检验（如何评估决策规则）

- 零假设：数据从分布 $P^*(X,Y)=P^*(X)P^*(Y)$ 中采样生成。一般令 $P^*(X,Y)=\hat{P}(X)\hat{P}(Y)$ 来生成零假设($\hat{P}(\cdot)\approx P^*(\cdot)$)。

- 检验数据是否符合零假设➔找到一个决策规则的过程。
  选择好决策规则后可计算错误拒绝概率：

$$P(\{D:R(D)=\mathrm{Re}\ ject\}\mid H_0,M)$$

  规则R有错误拒绝的概率*p*， *1-p*代表拒绝某个假设的决策置信度。

基于约束的方法

偏差度量（如何设计决策规则）

标准框架：定义偏离零假设的度量。

例如：独立随机变量：X、Y；

数据计数：$M[x,y]$ 接近于 $M\cdot\hat{P}(x)\cdot\hat{P}(y)$(计数的期望值)

度量

χ^2 **统计量**：$$d_{\chi^2}(D)=\sum_{x,y}\frac{(M[x,y]-M\cdot\hat{P}(x)\cdot\hat{P}(y))^2}{M\cdot\hat{P}(x)\cdot\hat{P}(y)}$$

$d_{\chi^2}(D)=0$ 完全拟合独立性假设的数据。

互信息量：$$d_I(D)=I_{\hat{P}_D}(X;Y)=\frac{1}{M}\sum_{x,y}M[x,y]\log\frac{M[x,y]}{M[x]M[y]}$$

设计规则：$$R_{d,t}(D)=\begin{cases}\text{接受，} & d(D)\le t\\ \text{拒绝，} & d(D)>t\end{cases}$$

其中，t 决定决策规则的错误拒绝概率。

基于约束的方法

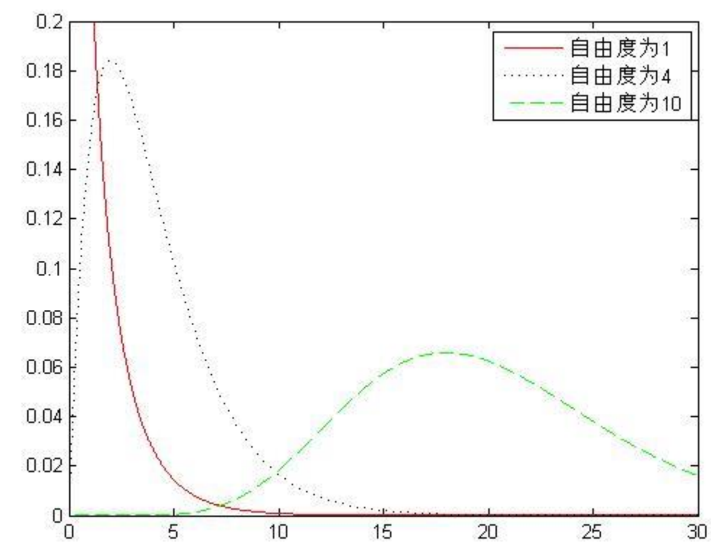
对 χ^2 统计量进行**扩展**以检验条件独立性：

假设希望给定*Z*时检验*X* 和*Y*是否独立。

零假设 H_0 是 $P^*(X,Y,Z)=\hat{P}(Z)\hat{P}(X\mid Z)\hat{P}(Y\mid Z)$

χ^2 统计量是：

$$d_{\chi^2}(D)=\sum_{x,y,z}\frac{(M[x,y,z]-M\cdot\hat{P}(z)\hat{P}(x\mid z)\hat{P}(y\mid z))^2}{M\hat{P}(x\mid z)\hat{P}(y\mid z)}$$



基于约束的方法

以上介绍了如何检测变量之间的独立性检测问题。得到变量间关系之后就可以**构建网络**了。

利用Build-PDAG算法构建网络：

详细见第3章算法3.5

- 存在**错误决策风险**，特别多重假设检验时，错误结论数量增加。
- Build-PDAG对以上错误很**敏感**，从而误导PDAG的生成。

结构得分

该类方法将结构学习看做一个**优化问题**来解决。

框架：

得分函数（本节）

对训练数据每个候选结构打分

↓

搜索一个得分高的结构

结构搜索（下节）

得分函数：

- 似然得分
- 贝叶斯得分

结构得分(似然得分)

数据挖掘实验室 Data Mining Lab

直观上，使数据尽可能代表的模型。

目标： 找到最大化似然图 ***G*** 和参数 θ_G (模型为$\langle G, \theta_G \rangle$ 组成)。

给定图结构G最大化似然参数：

$$\max_{\mathcal{G},\theta_{\mathcal{G}}} L(<\mathcal{G},\theta_{\mathcal{G}}>:\mathcal{D}) = \max_{\mathcal{G}}[\max_{\theta_{\mathcal{G}}} L(<\mathcal{G},\theta_{\mathcal{G}}>:\mathcal{D})]$$

$$= \max_{\mathcal{G}}[L(<\mathcal{G},\hat{\theta}_{\mathcal{G}}>:\mathcal{D})]$$

反之

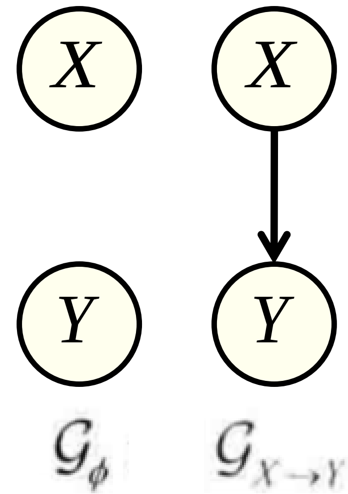
使用G的MLE参数时，寻找达到最大似然的图结构G

定义：$score_L(G:D) = l(\hat{\theta}_G : D)$

$l(\hat{\theta}_G : D)$是似然对数，$\hat{\theta}_G$是 G 的最大似然参数。

结构得分(似然得分)

信息论解释：



- G_0: X与Y独立

$$\text{score}_L(\mathcal{G}_0 : \mathcal{D}) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}$$

- G_1: 存在边 $X \to Y$

$$\text{score}_L(\mathcal{G}_1 : \mathcal{D}) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]}$$

- 得分差异：

$$\text{score}_L(\mathcal{G}_1 : \mathcal{D}) - \text{score}_L(\mathcal{G}_0 : \mathcal{D}) = \sum_m \log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]}$$

注： $x[m]$表示变量X为m状态。$\hat{\theta}_{x|y}$ 表示 $P(x)$ 的最大似然估计，$\hat{\theta}_{x|y}$ 是 $P(y|x)$ 的最大似然估计。

结构得分(似然得分)

数据挖掘实验室 Data Mining Lab

$$\mathrm{score}_L(\mathcal{G}_1:\mathcal{D})-\mathrm{score}_L(\mathcal{G}_0:\mathcal{D})=\sum_m \log\hat{\theta}_{y[m]|x[m]}-\log\hat{\theta}_{y[m]}$$

写出每个数据*x,y*

$$\mathrm{score}_L(\mathcal{G}_1:\mathcal{D})-\mathrm{score}_L(\mathcal{G}_0:\mathcal{D})=\sum_{x,y} M[x,y]\log\hat{\theta}_{y|x}-\sum_y M[y]\log\hat{\theta}_y$$

$$M[x,y]=M\cdot\hat{P}(x,y) \qquad \hat{\theta}_{y|x}=\hat{P}(y\mid x)$$

$$M[y]=M\cdot\hat{P}(y) \qquad \hat{\theta}_y=\hat{P}(y)$$

$$\mathrm{score}_L(\mathcal{G}_1:\mathcal{D})-\mathrm{score}_L(\mathcal{G}_0:\mathcal{D})=M\sum_{x,y}\hat{P}(x,y)\log\frac{\hat{P}(y\mid x)}{\hat{P}(y)}=M\cdot\boldsymbol{I}_{\hat{P}}(X;Y)$$

似然取决于变量间的互信息量。互信息量越大，依赖性越强

其中 $I_{\hat{P}}(X,Y)$是分布 $\hat{P}$ 中X 与 Y的互信息。

结构得分(似然得分)

推广到一般网络结构：

似然得分可以分解如下：

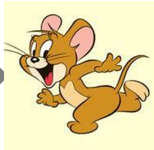
$$\text{score}_L(\mathcal{G}:\mathcal{D}) = M\sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i;\text{Pa}_{X_i}^{\mathcal{G}}) - M\sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i)$$

证明：定义： $score_L(G:D) = l(\hat{\theta}_G : D)$

$$\ell(\hat{\boldsymbol{\theta}}_{\mathcal{G}}:\mathcal{D}) = \sum_{i=1}^{n}\left[\sum_{u_i \in \text{Val}(\text{Pa}_{X_i}^{\mathcal{G}})} \sum_{x_i} M[x_i, \boldsymbol{u}_i] \log \hat{\theta}_{x_i|\boldsymbol{u}_i}\right]$$

$\boldsymbol{U}_i = \text{Pa}_{X_i}$

$$\frac{1}{M}\sum_{\boldsymbol{u}_i}\sum_{x_i} M[x_i, \boldsymbol{u}_i] \log \hat{\theta}_{x_i|\boldsymbol{u}_i}$$



$$\boldsymbol{I}_{\hat{P}}(X_i;\mathbf{U}_i) - \boldsymbol{H}_{\hat{P}}(X_i)$$

网络中变量的熵，不依赖网络结构

网络似然度量了**变量与其父节点的依赖强度**，即倾向于每个变量的父节点对变量有信息量的网络

结构得分(似然得分)

数据挖掘实验室
Data Mining Lab

最大似然得分**局限**

偏好更复杂的网络

$$\text{score}_L(\mathcal{G}_{X\to Y}:\mathcal{D})-\text{score}_L(\mathcal{G}_{\phi}:\mathcal{D})=M\cdot \boldsymbol{I}_{\hat{P}}(X;Y)$$

$$\forall D \qquad \text{score}_L(\mathcal{G}_{X\to Y}:\mathcal{D})\geq \text{score}_L(\mathcal{G}_{\phi}:\mathcal{D})$$ 互信息量非负

$G_{X\to Y}$ 为X 是Y 的父节点网络。

当**条件独立**在经验分布中**恰好确切成立**，最大似然网络将不表现出条件独立性。统计噪声使得这种情况几乎不存在。因此，似然**过拟合**了训练数据，**泛化能力降低**。

有没有更好的得分函数，使得更好的描述结构得分？

结构得分(贝叶斯得分)

数据挖掘实验室
Data Mining Lab

贝叶斯原则：对任何事物的不确定，都应为其确定一个分布。

结构 **参数**

定义：结构先验 $P(G)$，参数先验 $P(\theta_G \mid G)$

$$P(\mathcal{G} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{G}) P(\mathcal{G})}{P(\mathcal{D})}$$

贝叶斯得分定义：

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \underline{\log P(\mathcal{G})}$$

对比似然得分，贝叶斯得分有更好的**泛化能力，防止过拟合**？

结构得分(贝叶斯得分)

窥探一下

$$\mathrm{score}_B(\mathcal{G}:\mathcal{D}) = \boxed{\log P(\mathcal{D}\mid\mathcal{G})} + \log P(\mathcal{G})$$

考虑参数的不确定性

边缘似然：$P(\mathcal{D}\mid\mathcal{G}) = \int_{\Theta_{\mathcal{G}}} \underline{P(\mathcal{D}\mid\theta_{\mathcal{G}},\mathcal{G})}\,\underline{P(\theta_{\mathcal{G}}\mid\mathcal{G})}\,d\theta_{\mathcal{G}}$

给定网络时数据的似然

网络参数值的先验分布

边缘似然与最大似然差异：
- 最大似然：返回这个函数的**最大值**
- 边缘似然：是这个函数的**平均值**，基于先验度量 $P(\theta_{\mathcal{G}}\mid\mathcal{G})$ 做的平均。

最大似然：给定数据集D时最可能选择的参数 $\hat{\theta}$ 。
贝叶斯：提供多种选择及每种选择的可能性的度量。

结构得分(贝叶斯得分)

边缘似然很N，如何在简单情况下计算呢？

-----单个变量的边缘似然

例子：在实验室是否出现Jarry：

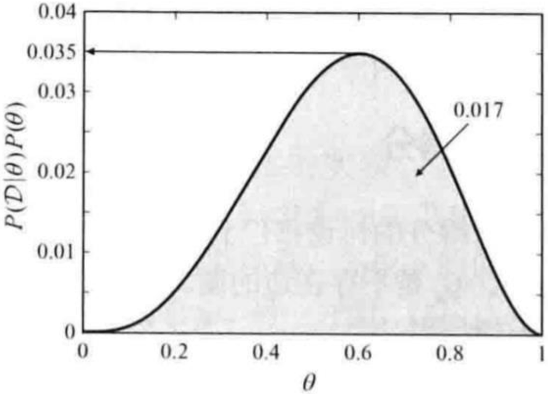
X ={出现，不出现}，其先验 $Dirichlet(\alpha_1,\alpha_0)$

出现次数：$M[1]$ ，不出现次数:$M[0]$ 。

给定数据D={出现，不出现，不出现，出现，..., 出现}。

最大似然：$P(\mathcal{D}\mid\hat{\theta})=\left(\frac{M[1]}{M}\right)^{M[1]}\cdot\left(\frac{M[0]}{M}\right)^{M[0]}$

边缘似然：$P(x[m+1]=H\mid x[1],\cdots,x[m])=\frac{M^m[1]+\alpha_1}{m+\alpha}$ （计数过程）



若D={出现，不出现，不出现，出现，出现};

最大似然：0.035；边缘似然：0.017.

太naïve ！

结构得分(贝叶斯得分)

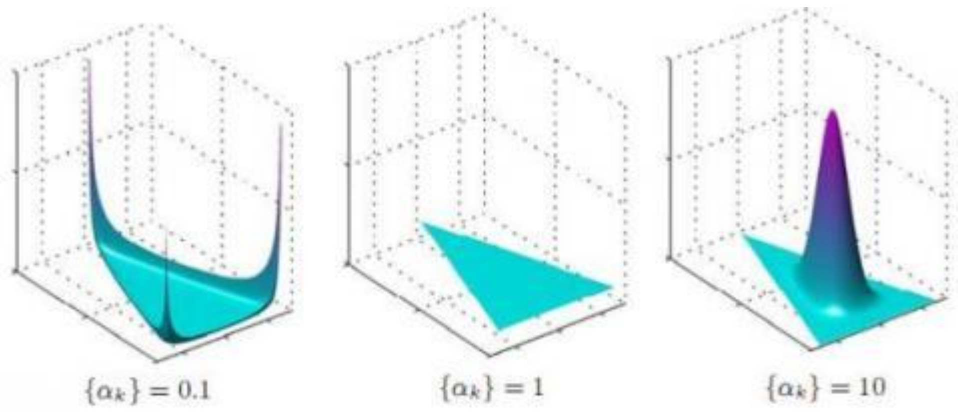
对于**先验分布**为 *Dirichlet*

通过之前讲的 *Dirichlet* 分布

是可以得到

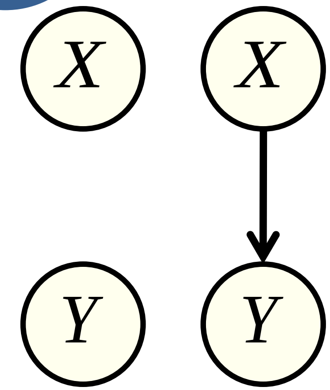
$$P(x[1],\cdots,x[M]) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+M)} \cdot \prod_{i=1}^{k} \frac{\Gamma(\alpha_i + M[x^i])}{\Gamma(\alpha_i)}$$

$$= P(\mathcal{D} \mid \mathcal{G})$$



结构得分(贝叶斯得分)

推广：



$\mathcal{G}_\phi$ $\qquad \mathcal{G}_{X\to Y}$

$$\mathcal{G}_\phi \quad P(\mathcal{D}\mid\mathcal{G}_\varnothing)=\int_{\Theta_X\times\Theta_Y}P(\theta_X,\theta_Y\mid\mathcal{G}_\varnothing)P(\mathcal{D}\mid\theta_X,\theta_Y,\mathcal{G}_\varnothing)\mathrm{d}[\theta_X,\theta_Y]$$

$$P(\mathcal{D}\mid\mathcal{G}_\varnothing)=\left(\int_{\Theta_X}P(\theta_X\mid\mathcal{G}_\varnothing)\prod_m P(x[m]\mid\theta_X,\mathcal{G}_\varnothing)\mathrm{d}\theta_X\right)$$

$$=\left(\int_{\Theta_Y}P(\theta_Y\mid\mathcal{G}_\varnothing)\prod_m P(y[m]\mid\theta_Y,\mathcal{G}_\varnothing)\mathrm{d}\theta_Y\right)$$

相同

$$\mathcal{G}_{X\to Y} \quad P(\mathcal{D}\mid\mathcal{G}_{X\to Y})=\left(\int_{\Theta_X}P(\theta_X\mid\mathcal{G}_{X\to Y})\prod_m P(x[m]\mid\theta_X,\mathcal{G}_{X\to Y})\mathrm{d}\theta_X\right)$$

$$\left(\int_{\Theta_{Y\mid x^0}}P(\theta_{Y\mid x^0}\mid\mathcal{G}_{X\to Y})\prod_{m:x[m]=x^0}P(y[m]\mid\theta_{Y\mid x^0},\mathcal{G}_{X\to Y})\mathrm{d}\theta_{Y\mid x^0}\right)$$

$$\left(\int_{\Theta_{Y\mid x^1}}P(\theta_{Y\mid x^1}\mid\mathcal{G}_{X\to Y})\prod_{m:x[m]=x^1}P(y[m]\mid\theta_{Y\mid x^1},\mathcal{G}_{X\to Y})\mathrm{d}\theta_{Y\mid x^1}\right)$$

其中***X***与***Y***都为二值变量。

结构得分(贝叶斯得分)

比较差异项：

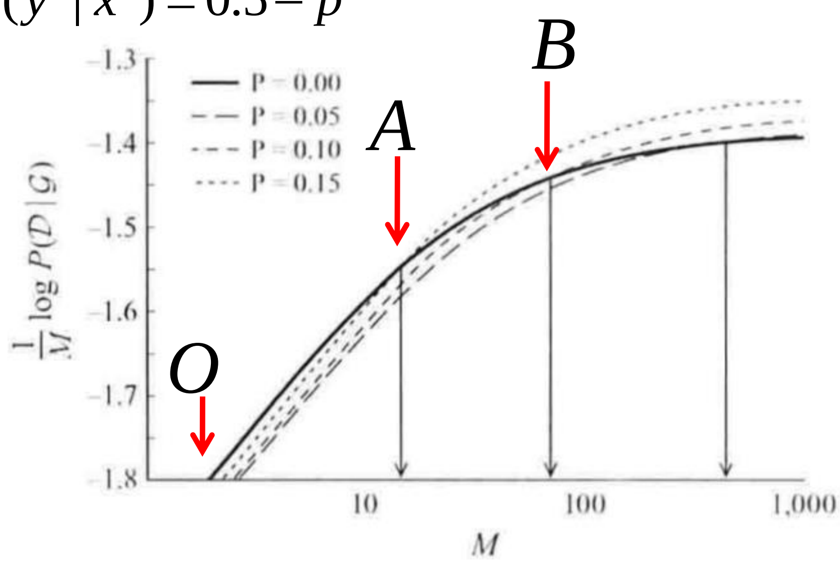
$$\left(\int_{\Theta_Y} P(\theta_Y \mid \mathcal{G}_\varnothing)\prod_m P(y[m]\mid\theta_Y,\mathcal{G}_\varnothing)\mathrm{d}\theta_Y\right)$$

$$\left(\int_{\Theta_{Y|x^0}} P(\theta_{Y|x^0} \mid \mathcal{G}_{X\to Y})\prod_{m:x[m]=x^0} P(y[m]\mid\theta_{Y|x^0},\mathcal{G}_{X\to Y})\mathrm{d}\theta_{Y|x^0}\right)$$
$$\left(\int_{\Theta_{Y|x^1}} P(\theta_{Y|x^1} \mid \mathcal{G}_{X\to Y})\prod_{m:x[m]=x^1} P(y[m]\mid\theta_{Y|x^1},\mathcal{G}_{X\to Y})\mathrm{d}\theta_{Y|x^1}\right)$$

若***Y***在***X***取值下为不同的分布，则 $\mathcal{G}_{X\to Y}$ 更好的刻画边缘似然。

给个实验：$P(x^1)=0.5$ $P(y^1\mid x^1)=0.5+p$ $P(y^1\mid x^0)=0.5-p$

p 是***X***与***Y***之间的依赖程度。



说明：

实线为 $\mathcal{G}_\phi$ ，即 $p=0$ 。

$O\to A$：选择实线（**简单模型**）

$A\to B$：选择虚线（**拟合数据**）

结构得分(贝叶斯得分)

如何理解贝叶斯得分

定义：**BIC得分**（贝叶斯近似）

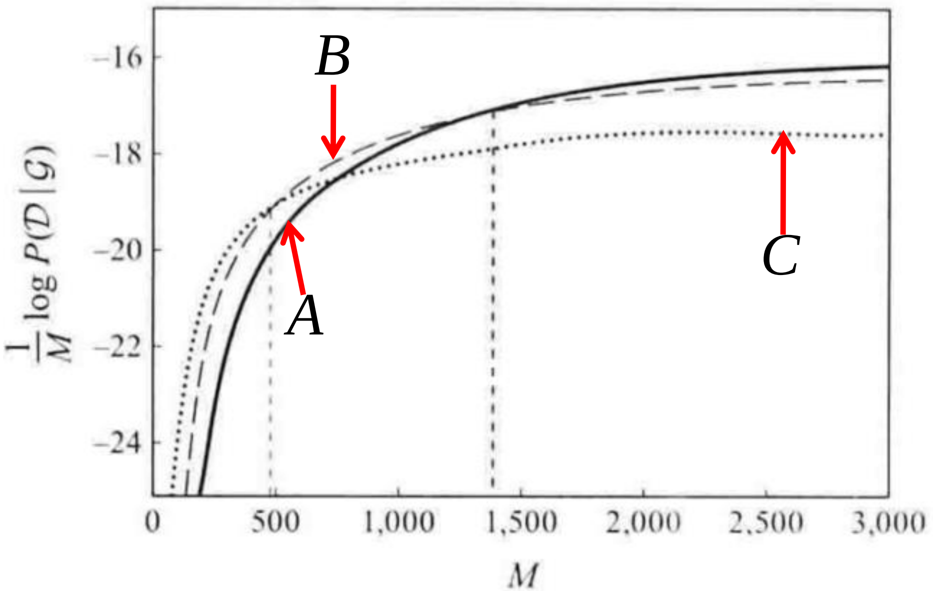
$$\mathrm{score}_{BIC}(\mathcal{G}:\mathcal{D}) = \ell(\hat{\theta}_{\mathcal{G}}:\mathcal{D}) - \frac{\log M}{2}\mathrm{Dim}[\mathcal{G}]$$

$Dim[G]$为模型的维数，或模型独立参数的数量。

$$\mathrm{score}_{BIC}(\mathcal{G}:\mathcal{D}) = M\sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i;\mathrm{Pa}_{X_i}) - M\sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i) - \frac{\log M}{2}\mathrm{Dim}[\mathcal{G}]$$

结论：

- 熵不依赖于图，对结构没影响。
- 变量对父节点依赖越强，得分越高。
- 网络越复杂，得分越低。
- M 越大，数据拟合重要性越大。



A：原始结构(含509个参数)；B：简化结构(359个参数)；C：树结构(含214个参数)

结构得分(贝叶斯得分)

$$P(G \mid D) \propto P(D \mid G) P(G)$$

对先验没有要求？

结构得分(先验性)

- 结构先验
- 参数先验

结构先验：经常对结构使用均匀先验（然并卵）。

参数先验：

什么样的先验是我们想要的呢？

结构得分(先验性)

数据挖掘实验室 Data Mining Lab

推广的似然得分：$\text{score}_L(\mathcal{G}:\mathcal{D}) = M\sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i;\text{Pa}_{X_i}^{\mathcal{G}}) - M\sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i)$

分解

$$\sum \text{FamScore}(X_i \mid \text{Pa}_i^{\mathcal{G}} : \mathcal{D})$$

$$\text{FamScore}_L(X \mid \boldsymbol{U} : \mathcal{D}) = M \cdot [\boldsymbol{I}_{\hat{P}}(X;\boldsymbol{U}) - \boldsymbol{H}_{\hat{P}}(X)]$$

可分解 ➔ 局部变化不影响其他部分 ➔ 采用局部搜索算法

我们希望

参数模块性：令$\{P(\theta_G \mid G): G \in G\}$ 是满足全局参数独立性的参数先验集合。如果对满足$Pa_{X_i}^{G} = Pa_{X_i}^{G'} = U$的每对$G$和$G'$，均有 $P(\theta_{X_i|U} \mid G) = P(\theta_{X_i|U} \mid G')$成立，则称先验满足参数模块性。

➔X_i 的CPD上的先验**仅依赖与网络的局部结构**（即X_i的父节点集），而不依赖于网络的其他部分。则意味着得分可分解。

➔若$P(\theta_G \mid G)$是满足全局参数独立性和参数模块性的一个**参数先验**，则网络结构上的贝叶斯得分可分解。

我们发现

结构得分(先验性)

数据挖掘实验室
Data Mining Lab

先验的表示：

这里用一种单纯的形式：Dirichlet分布

$$Dirichlet(\alpha,\alpha,\alpha,...,\alpha) \text{ 其中 } \alpha=1$$

结构搜索

明确优化问题：

输入：

- 训练集D
- 得分函数（如需要，包括先验）
- 可能的网络结构的集合G（融入任意先验知识）；

输出：

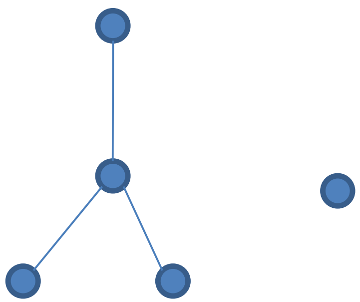
- 最大化的得分网络结构（可能来自结构的集合）

网络结构G的得分：

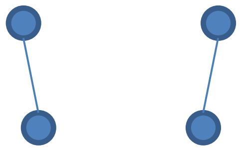
$$\text{score}(\mathcal{G}:\mathcal{D}) = \sum_i \text{FamScore}(X_i \mid \text{Pa}_i^{\mathcal{G}}:\mathcal{D})$$

结构搜索(树结构)

定义：如果每个变量X在G中至多有一个父节点，即 $\mathrm{Pa}_X^G |\leq 1$
则网络结构G称为树结构。



树



森林

关键性质：

- 可分解性
- 父节点的限制

结构搜索(树结构)

网络得分：

$$\Delta(\mathcal{G}) = \text{score}(\mathcal{G}:\mathcal{D}) - \text{score}(\mathcal{G}_\phi:\mathcal{D})$$

$$\Delta(\mathcal{G}) = \sum_{i,\text{Pa}_i^{\mathcal{G}} \neq \varnothing} (\text{FamScore}(X_i \mid \text{Pa}_i^{\mathcal{G}}:\mathcal{D}) - \text{FamScore}(X_i:\mathcal{D}))$$

定义

$$w_{j\to i} = \text{FamScore}(X_i \mid X_j:\mathcal{D}) - \text{FamScore}(X_i:\mathcal{D})$$

$$\Delta(\mathcal{G}) = \sum_{X_j \to X_i \in \mathcal{G}} w_{j\to i}$$

有向图加权图中寻找一个最大权值生成森林(困难)

结构搜索(树结构)

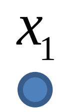
满足

$$w_{i \to j} = w_{j \to i}$$

有向图 → 无向图

结构搜索(给定顺序)

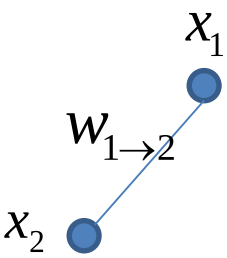
给定变量顺序：$\{x_1, x_2, x_3, x_4, x_5\}$



$$\arg_i\{w_{1\to i} \mid i = 2, 3, \ldots, 5\}$$

结构搜索(给定顺序)

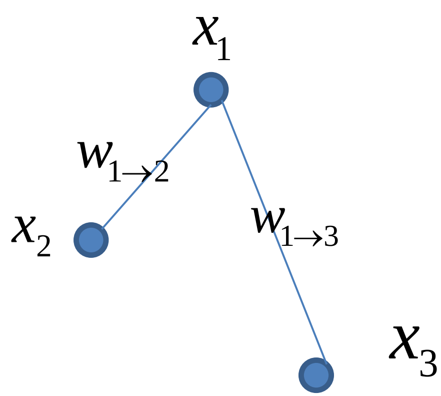
给定变量顺序： $\{x_1, x_2, x_3, x_4, x_5\}$



$$\arg_i\{w_{1\to i} \mid i = 2, 3, \ldots, 5\}$$

结构搜索(给定顺序)

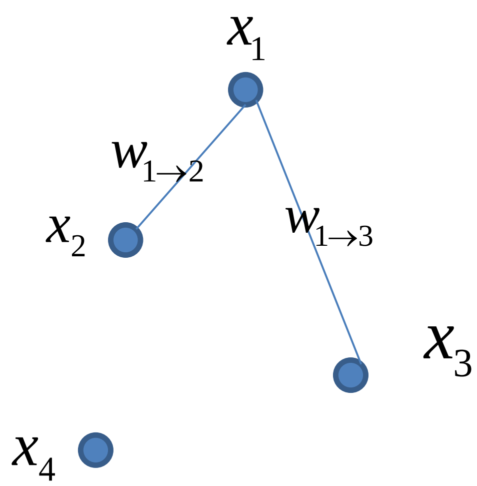
给定变量顺序： $\{x_1, x_2, x_3, x_4, x_5\}$



$$\underset{i}{\arg}\{w_{1\to i} \mid i = 2,3,\ldots,5\}$$

结构搜索(给定顺序)

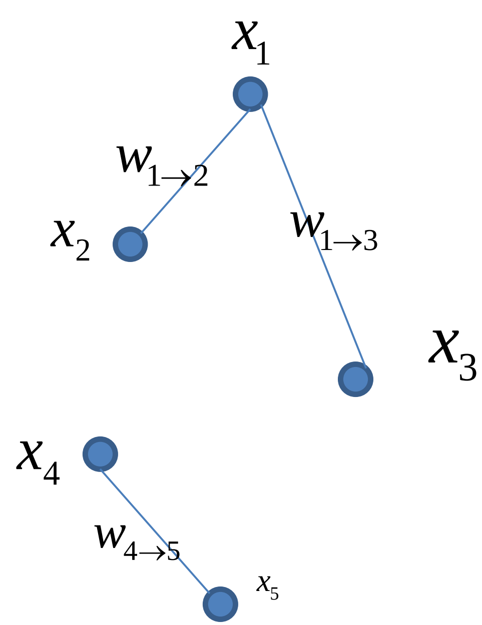
给定变量顺序： $\{x_1, x_2, x_3, x_4, x_5\}$



$$\arg_{i}\{w_{1\to i} \mid i = 2, 3, \ldots, 5\}$$

结构搜索(给定顺序)

给定变量顺序： $\{x_1, x_2, x_3, x_4, x_5\}$



$$\underset{i}{\arg}\{w_{1\to i} \mid i = 2, 3, \ldots, 5\}$$

节点数目较多时，比较的次数很多，学习更多的父节点，代价太高

结构搜索(一般图)

限制度：

$$G_d = \{G : \forall i, |Pa_{X_i}^G| \le d\}$$

如何搜索图：

定义操作：

- 添加边
- 删除边
- 反转边

结构搜索(一般图)

局部搜索算法：

idea：从当前“**状态**”出发，**寻找**与当前状态“**相近**”**的状态**，及邻近状态，其中邻近状态是通过搜索算子（边的操作）得到的。从邻近状态中**选取最优**的。

算法 A.5　带有搜索算子的贪婪局部搜索算法

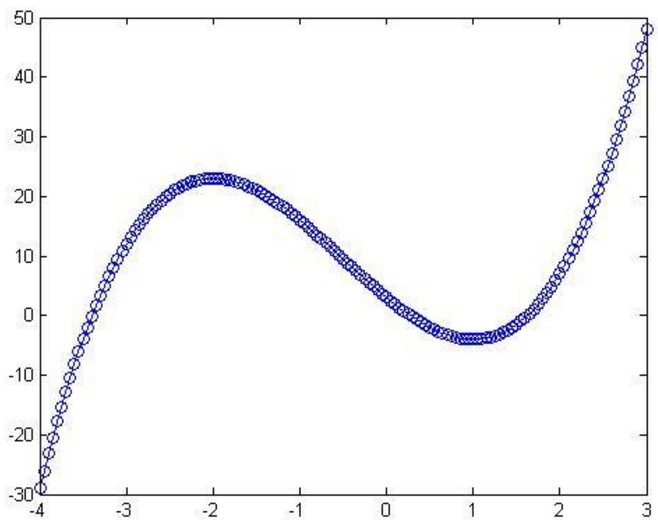
Procedure Greedy-Local-Search (
　　σ_0，　// 初始候选解
　　score，　//得分函数
　　$\mathcal{O}$，//搜索算子集
)

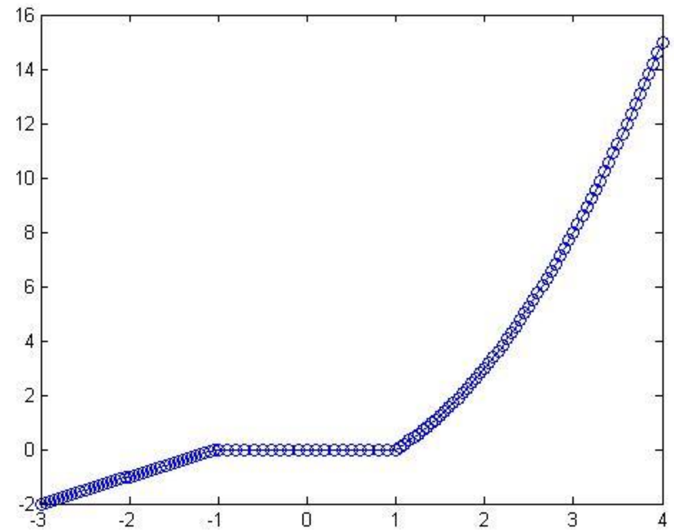
$\sigma_{\text{best}} \leftarrow \sigma_0$
执行
　$\sigma \leftarrow \sigma_{\text{best}}$
　progress $\leftarrow$ *false*
　对每个算子 $o \in \mathcal{O}$
　　$\sigma_o \leftarrow o(\sigma)$　//在 σ 上应用 o 的结果
　　如果 σ_o 为合法的解，那么
　　　如果 $\text{score}(\sigma_o) > \text{score}(\sigma_{\text{best}})$ 那么
　　　　$\sigma_{\text{best}} \leftarrow \sigma_o$
　　　　progress $\leftarrow$ *true*
当 *progress*
返回 σ_{best}

结构搜索(一般图)

局部爬山（贪婪爬山）正如刚才所讲；

问题来了：局部极大解；一个平台期；

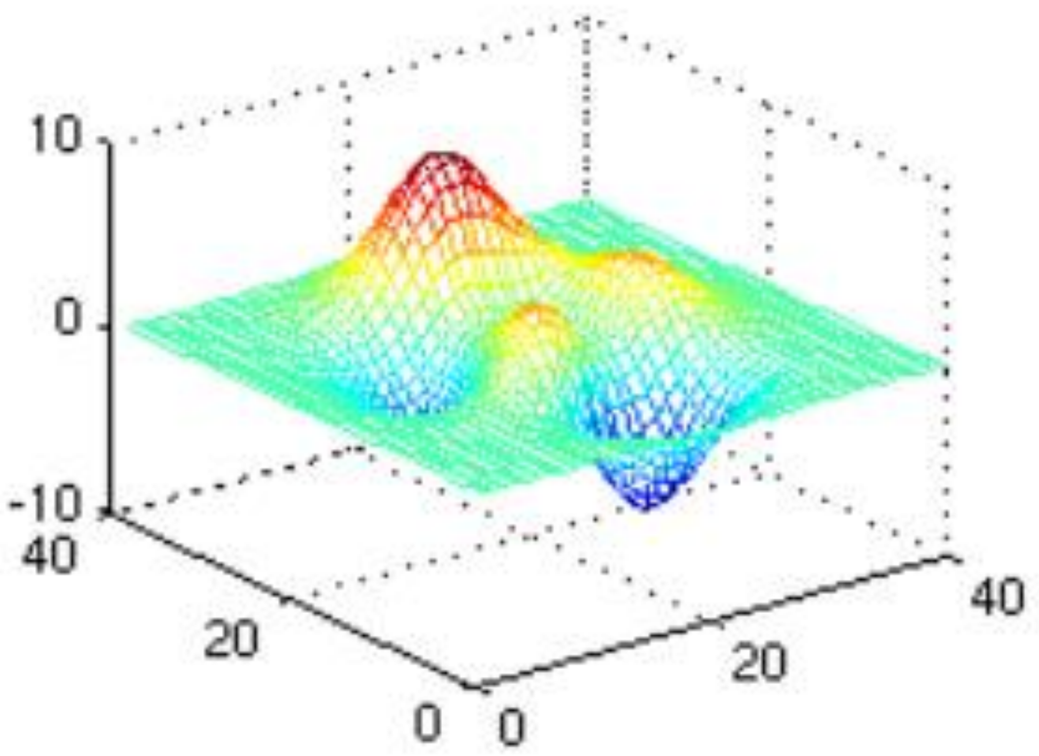




结构搜索(一般图)

数据挖掘实验室
Data Mining Lab

改进：强制新方向



总结

1. 基于约束的方法
2. 结构评分
   - 似然评分
   - 贝叶斯评分
   - 单变量的边缘似然
   - 贝叶斯网络的贝叶斯评分
   - 先验
3. 结构搜索

Thank You