bnlearnR语言中实现贝叶斯网络

本教程旨在介绍使用贝叶斯网络的基础，bnlearn和实际数据贝叶斯网络学习和推断，来寻找典型的图模型数据分析的分析框架。教程主要包括以下5项内容：

- 数据预处理
- 学习贝叶斯网路结构和参数
- 使用学习得到的结构作为预测模型
- 使用网络进行推断
- 对比外部信息验证网络

Bayesian networks

- Definitions
  - 贝叶斯网络结构，是一个有循环图G,图中每个节点$V_i \in \vee$对应一个随机变量;
  - 全局概率分布(X参数为Θ)，可以根绝图中存在的弧分界为更小的局部概率分布。
- 网络结构的主要作用是通过图形分离来表达模型中变量之间的条件独立关系，从而指定全局分布的因式如下:

$$P(\mathbf{X}) = \prod_{i=1}^{N} P(X_i|\Pi_{X_i}) \tag{1}$$

  这里$\Pi_{X_i} = parentsof X_i$。每一个变量都有自己的参数集合$\theta_{X_i}$，$U\theta_{X_i}$远小于$\theta$，因为一些相互独立的变量的参数被固定。如下面的案例所示:
- 贝叶斯神经网络中的第二组成部分是概率分布$P(X)$,BN的选择应该包括如下特征:
  - 可以从数据集中高效的学习;
  - 是灵活的，可以合理的编码各种现象;
  - 易于查询可以进行推理;
- 目前常见的三种选择:
  - 离散贝爷斯网络，其中X和$X_i|\Pi_{X_i}$是θ_{X_i}条件概率
    $\pi_{ik|j} = P(X_i = k|\Pi_{X_i=j})$

  - 高斯BN(GBN),其中X是多元正态，是$X_i|\Pi_{X_i}$通过线性回归模型定义的单变量正态分布
    $X_i = \mu X_i + \prod_{X_i} \beta_{X_i} + \varepsilon X_i$, εX_i $N(0, \sigma^2_{X_I})$
  - 条件线性高斯BN(CLGBN)，其中X是多元正态分布的混合，并且$X_i|\Pi_{X_i}$可以是多项式、单变量正态或正态混合。
    * 离散X_i仅允许有离散父代(表示为Δ_{X_i})，假设遵循用条件概率表参数化的多项分布;
    * 连续X_i允许同时具有离散和连续父代(表示Γ_{X_i},$\Delta_{X_i} \cup \Gamma_{X_i} = \Pi_{X_i}$，并且他们的局部分布
      $X_i|\Pi_{X_i}$ $N(\mu_{X_i,\sigma_{X_i}} + \Gamma_{X_i}\beta_{X_i,\delta_{X_i}}, \sigma^2_{X_i,\delta_{X_i}})$,也可以改写为:
      $X_i = \mu_{X_i,\delta_{X_i}} + \Gamma_{X_i}\beta_{X_i,\delta_{X_i}} + \varepsilon_{X_i,\delta_{X_i}}$，针对连续父代的线性回归的混合，其中离散父代的每个配置都有一个分量δ_{X_i}，如果X_i没有离散父代父母，混合物恢复为单一线性回归。

- **学习结构** BN模型选择和估计统称为学习，通常分为两步过程:
  - 结构学习，从数据中学习网络结构;
  - 参数学习，学习生上一步中学习的结构所隐含的局部分布。

这个工作流程是贝叶斯网络特有的，给定一个数据集$\mathcal{D}$,如果我们将全局分布的参数表示为$\mathbf{X}$,θ.

$$\mathrm{P}(\mathcal{M}|\mathcal{D}) = \mathrm{P}(\mathcal{G},\theta)|\mathcal{D}) = \mathrm{P}(\mathcal{G}|\mathcal{D}) \cdot \mathrm{P}(\theta|\mathcal{G},\mathcal{D}) \tag{2}$$

结构学习是在实践中完成的，

$$\mathrm{P}(\mathcal{G}|\mathcal{D}) \propto \mathrm{P}(\mathcal{G})\mathrm{P}(\mathcal{D}|\mathcal{G}) = \mathrm{P}(\mathcal{G}) \int \mathrm{P}(\mathcal{D}|\mathcal{G},\theta)\mathrm{P}(\theta|\mathcal{G})\mathrm{d}\theta \tag{3}$$

结合局部分布$\mathcal{P}(\mathcal{D}|\mathcal{G},\theta)$分解为仅依赖于$X_i$及其父分布的局部分布的。

$$\mathbf{P}(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G},\theta)P(\theta|\mathcal{G})\mathrm{d}\theta = \prod_{i=1}^{N} \int P(\mathrm{X_i}|\Pi_{X_i},\theta_{X_i})P(\theta_{X_i}|\Pi_{X_i})\mathrm{d}\theta_{X_i} \tag{4}$$

一旦从结构学习中得到了估计$\mathcal{G}$，就可以识别局部分布$X_i|\Pi_{X_i}$，因为已知每个节点的父节点。可以相互独立的估计他们的参数集θ_{X_i}。假设$\mathcal{G}$是稀疏的，每个$X_i|\Pi_{X_i}$都只涉及很少的变量，因此估计其参数在计算上很简单。

- **推理** 贝叶斯网络的推理通常由条件概率或者最大后验(MAP)查询得到，CP查询的总体思路是：
  - 我们有一些证据，也就是我们知道一些变量的值，并相应地修复节点；
  - 我们想根据我们拥有的证据来研究涉及其他变量的某些事件的概率。

  MAP 查询的目标是在给定一些证据的情况下找到网络中最高概率的变量(子集)值的组合。如果证据是部分观察到的新个体，则执行MAP查询相当于经典的预测练习。
- **bnlearn包** bnlearn旨在为方法研究提供灵活的模拟套件，以及与BN一起处理正式世界数据的有效且可拓展的数据分析工具。这是通过模块化架构实现的，其中算法与模型假设分离，从而可以混合和匹配文献中的方法：bnlearn利用 BN 的模块化来实现这一目标如下：
  - 学习网络的结构bn，或者手动创建一个网络结构，会给出一个编码的类对象$\mathcal{G}$；
  - 学习给定结构的参数从一个对象开始，并给出一个bn.fit目标类，其编码为$(\mathcal{G}, \theta)$
  - 推断采用bn.fit类别目标。