模型-预测主题-离散型预测-贝叶斯网络【czy】

1. 贝叶斯网概览

1.1 模型名称

贝叶斯网络（Bayesian network），又称信念网络（belief network）或是有向无环图模型（directed acyclic graphical model），是一种概率图型模型。

1.2 贝叶斯派理论背景(小故事 可忽略)

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，即要么发生，要么不发生，不去考虑某件事情发生的概率有多大，不发生的概率又是多大。而且概率虽然未知，但最起码是一个确定的值。比如问那时的人们一个问题：“有一个袋子，里面装着若干个白球和黑球，请问从袋子中取得白球的概率是多少？”他们认为取出白球的概率就是，要么取到白球，要么取不到白球，即θ只能有一个值，不是$\frac{1}{2}$，就是0，而且不论你取了多少次，取得白球的概率θ始终都是$\frac{1}{2}$，即不随观察结果X的变化而变化。 这种频率派的观点长期统治着人们的观念，直到后来一个名叫托马斯·贝叶斯（Thomas Bayes，1702-1763）的人物出现。 托马斯·贝叶斯在世时，并不为当时的人们所熟知，很少发表论文或出版著作，与当时学术界的人沟通交流也很少,可他最终发表了一篇名为“An essay towards solving a problem in the doctrine of chances”，即机遇理论中一个问题的解。这篇论文的发表随机产生轰动效应，从而奠定贝叶斯在学术史上的地位。 事实上，上篇论文发表后，在当时并未产生多少影响，在20世纪后，这篇论文才逐渐被人们所重视。回到上面的例子：“有一个袋子，里面装着若干个白球和黑球，请问从袋子中取得白球的概率θ\thetaθ是多少？”贝叶斯认为取得白球的概率是个不确定的值，因为其中含有机遇的成分。 总结可得频率派与贝叶斯派各自不同的思考方式：

- 频率派把需要推断的参数θ看做是固定的未知常数，即概率虽然是未知的，但最起码是确定的一个值，同时，样本X是随机的，所以频率派重点研究样本空间，大部分的概率计算都是针对样本X的分布。

- 贝叶斯派的观点则截然相反，他们认为参数θ是随机变量，而样本X是固定的，由于样本是固定的，所以他们重点研究的是参数的分布。

贝叶斯派既然把θ看做是一个随机变量，所以要计算θ的分布，便得事先知道θ的无条件分布，即在有样本之前（或观察到X之前），θ有着怎样的分布呢？ 比如往台球桌上扔一个球，这个球落会落在何处呢？如果是不偏不倚的把球抛出去，那么此球落在台球桌上的任一位置都有着相同的机会，即球落在台球桌上某一位置的概率服从均匀分布。这种在实验之前定下的属于基本前提性质的分布称为先验分布，或者无条件分布。 至此，贝叶斯及贝叶斯派提出了一个思考问题的固定模式：

先验分布$\pi(\theta)$**+ 样本信息**$\chi \Rightarrow$**后验分布**$\pi(\theta|x)$

上述思考模式意味着，新观察到的样本信息将修正人们以前对事物的认知。换言之，在得到新的样本信息之前，人们对θ的认知是先验分布$\pi(\theta)$，在得到新的样本信息χ后，人们对θ的认知为$\pi(\theta|x)$。

2. 贝叶斯定理（条件概率）

2.1 理论部分

在引出贝叶斯定理之前，先考虑$P(A|B)$，即在B发生的情况下A发生的可能性。

1. 首先，事件B发生之前，我们对事件A的发生有一个基本的概率判断，称为A的先验概率，用$P(A)$表示；
2. 其次，事件B发生之后，我们对事件A的发生概率重新评估，称为A的后验概率，用$P(A|B)$表示；
3. 类似的，事件A发生之前，我们对事件B的发生有一个基本的概率判断，称为B的先验概率，用$P(B)$表示；
4. 同样，事件A发生之后，我们对事件B的发生概率重新评估，称为B的后验概率，用$P(B|A)$表示；

根据条件概率的定义，在事件B发生的条件下事件A发生的概率为

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

同理，在事件A发生的条件下事件B发生的概率

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

整理可得到贝叶斯定理的公式表达式

$$P(A|B) = \frac{P(B \cap A)P(A)}{P(B)}$$

2.2 举例：拼写检查

当在搜索引擎不小心输入一个不存在的单词时，搜索引擎会提示你是不是要输入某一个正确的单词，比如当你在Google中输入“Julw”时，系统会提示你是不是要搜索“July”， 这叫做拼写检查。根据谷歌一员工写的文章显示，Google的拼写检查基于贝叶斯方法。

用户输入一个单词时，可能拼写正确，也可能拼写错误。如果把拼写正确的情况记做c（代表correct），拼写错误的情况记做w（代表wrong），那么"拼写检查"要做的事情就是：在发生w的情况下，试图推断出c。换言之：已知w，然后在若干个备选方案中，找出可能性最大的那个c，也就是求$P(c|w)$的最大值。

而根据贝叶斯定理，有：

$$P(c|w) = \frac{P(w|c)P(c)}{P(w)}.$$

由于对于所有备选的c来说，对应的都是同一个w，所以它们的$P(w)$是相同的，因此我们只要最大化$P(w|c)P(c)$即可。其中：

- $P(c)$表示某个正确的词的出现"概率"，它可以用"频率"代替。如果我们有一个足够大的文本库，那么这个文本库中每个单词的出现频率，就相当于它的发生概率。某个词的出现频率越高，$P(c)$就越大。
- $P(w|c)$表示在试图拼写c的情况下，出现拼写错误w的概率。为了简化问题，假定两个单词在字形上越接近，就有越可能拼错，$P(w|c)$就越大。举例来说，相差一个字母的拼法，就比相差两个字母的拼法，发生概率更高。你想拼写单词July，那么错误拼成Julw（相差一个字母）的可能性，就比拼成Jullw（相差两个字母）高。

所以，我们只要找到与输入单词在字形上最相近的那些词，再在其中挑出出现频率最高的一个，就能实现$P(w|c)P(c)$的最大值。

3. 朴素贝叶斯分类器

3.1 理论部分

朴素贝叶斯分类器是根据贝叶斯公式，在假设样本的各个特征相互独立的情况下进行分类的方法。朴素贝叶斯分类的正式定义如下：

1. 设$x=\{a_1,a_2,\cdots,a_m\}$为一个待分类的样本，其中每个a为x这个样本的一个特征属性；
2. 设$C=\{y_1,y_2,\cdots,y_n\}$为类别集合，即任意样本x属于这n个类别中的一个；
3. 计算后验概率$P(y_1|x),P(y_2|x),\cdots,P(y_n|x)$
4. 若$P(y_k|x)=\max\{P(y_1|x),P(y_2|x),\cdots,P(y_n|x)\}$，判定$x\in y_k$。

其中，第3步计算条件概率是整个方法的核心，其原理是我们熟知的贝叶斯公式，具体如下：

1. 找到一个已知分类的待分类项集合$x=\{b_1,b_2,\cdots,b_m\}$，作为训练样本集。
2. 统计各类别下各个特征的条件概率，即：

$$P(b_1|y_1),P(b_2|y_1),\cdots,P(b_m|y_1);P(b_1|y_2),P(b_2|y_2),\cdots,P(b_m|y_2);\cdots;P(b_1|y_n),P(b_2|y_n),\cdots,P(b_m|y_n).$$

3. 根据贝叶斯定理$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{P(x)}$， 因为分母对于所有类别为常数，故只要将分子最大化皆可。又因为各特征条件独立，故：$P(x|y_i)P(y_i)=P(y_i)\times\prod_{k=1}^{m}P(b_k|y_i)$.

总体来说，朴素贝叶斯分类器分三个阶段进行工作：

1. **准备工作**:

   这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征，并对每个特征进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征、特征划分及训练样本质量决定。

2. **分类器训练**

   这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征划分对每个类别的条件概率估计，并将结果记录。其输入是特征和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

3. **应用**

   这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

3.2 举例：预测柯南中凶手和被害人

本示例介绍在朴素贝叶斯模型的基础上，通过角色特征（性格、行为、与他人关系等）预测柯南中人物身份（凶手/被害人）。此处使用长春版漫画单行本1-70卷中共60个事件，以下称``训练数据"。模型先计算训练数据中角色拥有各种特征组合时是凶手或被害人的概率，再以此预测新数据（1-70卷中训练数据后面的21个事件）中角色的身份。

首先收集数据，即1-70卷的事件中的凶手、被害人都有些什么特征。我们感兴趣的是杀人事件，排除掉非杀人事件后共81 个事件被统计，皆有且只有一名凶手和一至两名被害人。涉案角色为3 至9 不等，平均5 人，共404 名角色的20个特征被统计。

这些特征的选择基于个人经验和一些大家熟悉的对剧情或人物的调侃这类先验信息，比如凶手一开始大多慈眉善目甚至案发后有不在场证明；被害人一般都凶神恶煞让人讨厌，或者在大家说最好待在一起时非要自己待着；还有事后被证明无辜的人中有部分会被毛利大叔错误指认。

这些特征首先被以000、111编码（表现出某个特征则编码为111，否则为000）。于是每个角色就有了202020个代表他们特征的值（由0或1组成），并且也有两个值代表他们是否为凶手或被害人（比如某个人是凶手，那么他的这两个值就是1,0；是被害人就是0,1；都不是就是0,0）。

此后，我们就可通过回归分析看凶手和被害人这两个身份可能被哪些特征预测了，这里我使用了**Logistic回归**。这一步是因为这二十个特征并不一定都有很好的预测能力，预先筛选一下可以让贝叶斯模型更精简。这一步后对"凶手"或"被害人"在0.05水平上显著的特征被留下并进入贝叶斯模型。对凶手有预测能力（包括正相关和负相关）的特征包括"对除主角以外的周围人不友善"、"对周围人友善"、"有不在场证明等有利证据"、"遭遇攻击但未死"、"与死者是恋人或婚姻关系"。之后在此基础上我们又添加了"被害人死后表现出悲伤"、"被小五郎指为凶手"、"有对自己的不利证据"等三个特征。对被害人有预测能力的特征包括"对除主角以外的周围人不友善"、"对主角不友善"、"表现出紧张或惊恐"、"要自己待着"，我之后又添加了性别和年龄（50岁以上或以下）几个特征。此外，被害人相关特征在有人被杀前统计，因为对被害人的预测需要在事件发生前做出；相反，凶手相关特征则根据凶手被正确指出前所有角色的表现来统计。

接下来的一步是分别对凶手和被害人建立朴素贝叶斯模型，算出各个可能的特征组合有多大概率对应"凶手"或"被害人"身份。

"凶手"的贝叶斯模型为

$$P(offender|x_1,x_2,\cdots,x_n)=\frac{P(x_1,x_2,\cdots,x_n|offender)P(offender)}{P(x_1,x_2,\cdots,x_n)}$$

"被害人"的贝叶斯模型为

$$P(victim|x_1,x_2,\cdots,x_n)=\frac{P(x_1,x_2,\cdots,x_n|victim)P(victim)}{P(x_1,x_2,\cdots,x_n)}$$

其中，$x_1,\cdots,x_n$为各个特征取值的组合，具有第i个特征则$x_i=1$，否则$x_i=0$. $P(offender)$和$P(victim)$为先验概率，可忽略。

在对全部训练数据中的404个角色进行计算之后，我们就得到了各个特征组合对应的凶手概率和被害人概率。然后就可以把这两个概率应用在新数据（70卷之后的单行本）上了。具体来说，就是先把新数据中每个涉案角色的特征组合统计出来，然后分别计算他们是凶手或被害人的概率。在每个事件中，"凶手"概率最高的人被预测为凶手，"被害人"概率最高的人被预测为被害人。如果出现多于一个概率最高值，则拥有这个值的人都被预测为凶手/被害人。

之后就是计算模型预测准确率并将其与机遇水平（瞎蒙正确率）相比较。那么预测准确率怎么算呢？如果是只预测其中一个人为凶手，那么在每个事件中预测对了就记为100%，错了就记为0；如果预测多于一人（M人，M>1）为凶手，且其中一个正确，则记为(100/M)%，如果没有一个正确则记为0。预测被害人准确率的计算与预测凶手准确率类似，只是被害人有可能多于一个。这种情况下，完全预测正确记为100%，只预测正确其中一人记为50%50\%50%。预测凶手的机遇水平为(100/A)%(100/A)\%

(100/A)%，此处A为还活着的人的人数，因为被害人已经被排除了；预测被害人的机遇水平为(100/N)%(100/N)\%(100/N)%，N为事件涉案人数。总的来说预测被害人的机遇水平更低一些。

总得来说，通过这些特征预测被害人的准确率高于预测凶手的准确率，这说明作者对被害人的塑造更为脸谱化，而凶手特征则比较多变。

4. 贝叶斯网络

4.1 理论部分

贝叶斯网络（Bayesian Network），又称信念网络（Belief Network），或有向无环图（Directed Acylic Graphical Model，DAG），或一种概率图模型，于1985年由Judea Pearl首先提出。它是一种模拟人类推理过程中因果关系的不确定性处理模型，其网络拓扑结构是一个DAG。

贝叶斯网络的有向无环图中的节点表示随机变量$X_1, X_2, \cdots, X_n$，它们可以是可观察的变量，或隐变量、未知参数等。认为有因果关系（或非条件独立）的变量或命题则用箭头来连接。若两个节点间一个单箭头连接在一起，表示其中一个节点是“因”（parent），另一个是“果”（children），两节点就会产生一个条件概率值。

例如，假设节点E直接影响到节点H，即$E \rightarrow H$，则用从E指向H的箭头建立两节点的有向弧(E,H)，边的权值用条件概率P(H|E)来表示，如图1所示。

图 1: 两变量依赖性的例子

简言之，把某个研究系统中涉及的随机变量，根据是否条件独立绘制在一个有向图中，就形成了贝叶斯网络。其主要用来描述随机变量之间的条件依赖，用圈表示随机变量，用有向弧表示条件依赖。

1. 定义 （贝叶斯网络）

令G=(I,E)表示一个DAG，其中I代表图形中所有节点的集合，而E代表有向连接线段的集合，且令$X = (X_i), i \in I$为有向无环图中的某一节点i所代表的随机变量，若节点X的联合概率可以表示成：

$$p(X) = \prod_{i \in I} p(X_i | X_{pa(i)})$$

则称X为相对于一有向无环图G的贝叶斯网络，其中pa(i)表示节点X_i之“因”。

根据节点间依赖关系的不同，贝叶斯网络中的节点可分为三种基本结构。

2. 形式 1：head-to-head

贝叶斯网络的第一种结构形式如图2所示。

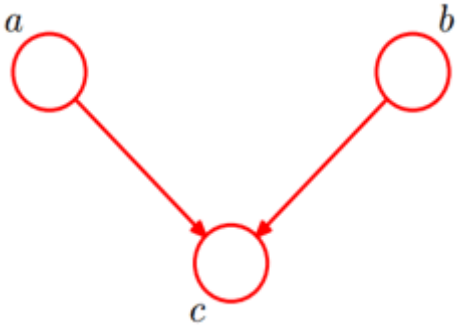

图 2: head-to-head 结构

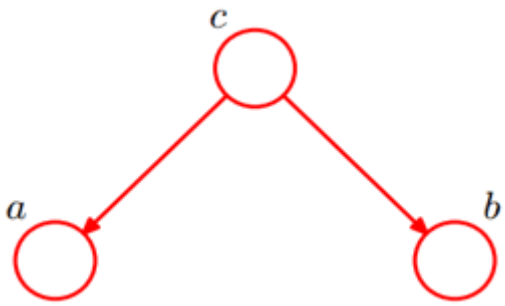

图 3: tail-to-tail 结构

命题 1在head-to-head结构中，$P(a,b) = P(a) \cdot P(b)$

证明 根据定义1，有

$P(a,b,c) = P(a) \cdot P(b) \cdot P(c|a,b),$

再对等式两边遍历变量c的所有取值，

$\sum_c P(a,b,c) = \sum_c P(a) \cdot P(b) \cdot P(c|a,b)$

化简可得，

$P(a,b) = P(a) \cdot P(b).$

由命题1知，在c的未知的条件下，a、b被阻断（blocked），是独立的，称之为head-to-head条件独立。

3. 形式 2：tail-to-tail

贝叶斯网络的第二种基本结构是图4。

命题 2 在tail-to-tail网络结构中，$P(a,b|c) = P(a|c) \cdot P(b|c).$

证明 根据定义1，有

$P(a,b,c) = P(c) \cdot P(a|c) \cdot P(b|c),$

又因为

$P(a,b|c) = \frac{P(a,b,c)}{P(c)}$

所以

$P(a,b|c) = P(a|c) \cdot P(b|c).$

由命题2知，在c的给定的情况下，a、b被阻断（blocked），是独立的，称之为tail-to-tail条件独立。

4. 形式3：head-to-tail

贝叶斯网络的第三种结构形式如图4所示。

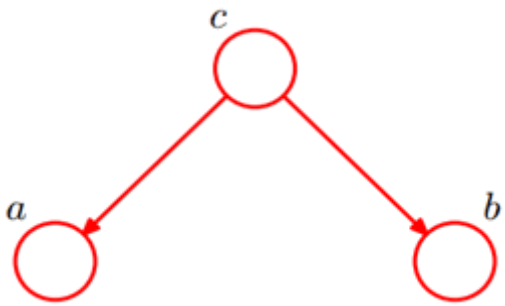

图 4: tail-to-tail 结构

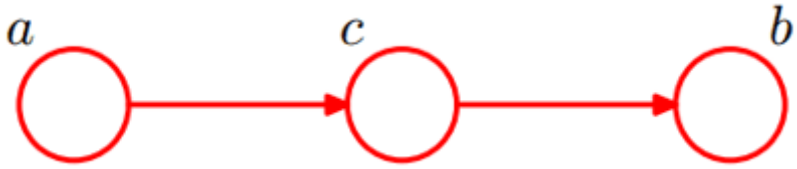

图 5: head-to-tail 结构

命题 3 在head-to-tail结构中，$P(a,b|c) = P(a|c) \cdot P(b|c)$。

根据定义1，有

$P(a,b,c) = P(a) \cdot P(c|a) \cdot P(b|c),$

因此有，

$P(a,b|c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a) \cdot P(c|a) \cdot P(b|c)}{P(c)} = \frac{P(a,c) \cdot P(b|c)}{P(c)} = P(a|c) \cdot P(b|c).$

由命题3知，在c给定的条件下，a、b被阻断（blocked），是独立的，称之为head-to-tail条件独立。

这个head-to-tail其实就是一个链式网络，如图5所示。在x_i给定的情况下，x_{i+1}的分布和$x_1, x_2, \cdots, x_{i-1}$条件独立。也就是说，$x_{i+1}$的分布状态只和$x_i$有关，和其他变量条件独立，这种顺次演变的随机过程，就叫做马尔科夫链）。

5. 代码实现

由于Matlab自带的贝叶斯网工具箱（BNT）具有完备的函数系统和可视化工具，因而建议使用 Matlab 实现应用场景中的贝叶斯网问题；BNT工具箱自带的可视化工具对贝叶斯网的结构训练 很重要。模板代码实现平台为Matlab（2014版以后），BNT工具包可能需要另行安装，安装包（提取码：mspx）。

安装方法：

1. 解压FullBNT-1.0.4.zip，将整个目录FullBNT-1.0.4复制到MATLAB的安装目录的TOOLBOX目录下
2. 打开Matlab，在MATLAB命令窗口中输入以下命令：

```
\>> addpath(genpath('D:\MATLAB7\toolbox\FullBNT-1.0.4'))
```

3. %上述路径为范例，请酌情修改
4. 永久保存路径

> savepath

5. 检验是否安装成功

> which test_BNT.m

代码说明： 贝叶斯网分为参数学习和结构学习两种应用场景，模板代码中使用五个因子节点作为示例，模拟 过劳死现象与多种诱发因素的可能概率关系。比赛时可根据具体应用场景增减因子，并修改相应 的因子关系（由矩阵表示）和概率表（由tabular_CPD(·)实现）。学习过程通过函数 learn_params(BNT, DATA); %参数学习 或者 learn_struct_K2(·); %结构学习 实现。 BNT提供可视化功能，通过函数 draw_graph(dag)实现，其中dag（有向无环图）为节点的关系矩阵。

其余实现逻辑在代码注释中体现。由于版本问题可能出现注释乱码，请参考http://blog.csdn.net/soliddream66/article/details/61414565更改配置，或者参考如下注释。

5.1 结构学习

```
N=5;%四个节点分别是国家政策C,学校政策U,工作压力大W,身体状况差B，过劳死D

dag=zeros(N,N);%网络连接矩阵初始化

C=1;U=2;W=3;B=4;D=5;%初始化节点顺序

dag(C,U)=1;%定义节点之间的连接关系
dag(U,[W B])=1;
dag(W,D)=1;
dag(B,D)=1;

discrete_nodes=1:N;%离散节点

node_sizes=2*ones(1,N);%节点状态数

%建立网络架构

bnet=mk_bnet(dag,node_sizes,'names',{'国家政策（C）','学校政策（U）','工作压力大
（W）','身体状况差（B）','过劳死（D）'},'discrete',discrete_nodes);

%手工构造条件概率CPT表

bnet.CPD{C} = tabular_CPD(bnet,C,[0.5 0.5]);
bnet.CPD{U} = tabular_CPD(bnet,U,[0.95 0.01 0.05 0.99]);
bnet.CPD{W} = tabular_CPD(bnet,W,[0.9 0.05 0.1 0.95]);
bnet.CPD{B} = tabular_CPD(bnet,B,[0.3 0.01 0.7 0.99]);
bnet.CPD{D} = tabular_CPD(bnet,D,[0.335 0.3 0.05 0 0.665 0.7 0.95 1]);

%画出建立好的贝叶斯网络

 figure
 draw_graph(dag)

%手动构造样本数据samples：
nsamples=2000;
samples=cell(N,nsamples);

for i=1:nsamples
    samples(:,i)=sample_bnet(bnet);
end

data=cell2num(samples);

%结构学习

order=[1 2 3 4 5];   % 节点次序
ns=[2 2 2 2 2];        % 节点属性值的个数
max_fan_in=2;          % 最大父节点数目

dag2 = learn_struct_K2(data,ns,order,'max_fan_in',max_fan_in);
bnet2=mk_bnet(dag2,node_sizes,'names',{'国家政策（C）','学校政策（U）','工作压力大
（W）','身体状况差（B）','过劳死（D）'},'discrete',discrete_nodes);

%手工构造条件概率CPT表
```

```
bnet2.CPD{C} = tabular_CPD(bnet2,C,[0.5 0.5]);
bnet2.CPD{U} = tabular_CPD(bnet2,U,[0.95 0.01 0.05 0.99]);
bnet2.CPD{W} = tabular_CPD(bnet2,W,[0.9 0.05 0.1 0.95]);
bnet2.CPD{B} = tabular_CPD(bnet2,B,[0.3 0.01 0.7 0.99]);
bnet2.CPD{D} = tabular_CPD(bnet2,D,[0.335 0.3 0.05 0 0.665 0.7 0.95 1]);
figure

draw_graph(dag2);  %画出建立好的贝叶斯网络
CPT2=cell(1,N);

for i=1:N
    s=struct(bnet2.CPD{i});
    CPT2{i}=s.CPT;
end

fprintf('输出结构学习之后过劳死节点参数：\n');

dispcpt(CPT2{5});
```

5.2 参数学习

```
%BNT的参数学习%

N=5;  %四个节点分别是国家政策C,学校政策U,工作压力大W,身体状况差B，过劳死D
dag=zeros(N,N);  %网络连接矩阵初始化
C=1;U=2;W=3;B=4;D=5;  %初始化节点顺序

dag(C,U)=1;  %定义节点之间的连接关系
dag(U,[W B])=1;
dag(W,D)=1;
dag(B,D)=1;

discrete_nodes=1:N;  %离散节点

node_sizes=2*ones(1,N);  %节点状态数

%建立网络架构

bnet=mk_bnet(dag,node_sizes,'names',{'国家政策（C）','学校政策（U）','工作压力大
（W）','身体状况差（B）','过劳死（D）'},'discrete',discrete_nodes);

%手工构造条件概率CPT表

bnet.CPD{C} = tabular_CPD(bnet,C,[0.5 0.5]);
bnet.CPD{U} = tabular_CPD(bnet,U,[0.95 0.01 0.05 0.99]);
bnet.CPD{W} = tabular_CPD(bnet,W,[0.9 0.05 0.1 0.95]);
bnet.CPD{B} = tabular_CPD(bnet,B,[0.3 0.01 0.7 0.99]);
bnet.CPD{D} = tabular_CPD(bnet,D,[0.335 0.3 0.05 0 0.665 0.7 0.95 1]);

%画出建立好的贝叶斯网络

figure
draw_graph(dag)

%手动构造样本数据samples：
```

```
nsamples=20000;
samples=cell(N,nsamples);
for i=1:nsamples
    samples(:,i)=sample_bnet(bnet);
end

data=cell2num(samples);
bnet2 = mk_bnet(dag,node_sizes,'discrete',discrete_nodes);

%手动构造条件概率表cpt

seed=0;
rand('state',seed);

bnet2.CPD{C}=tabular_CPD(bnet2,C);
bnet2.CPD{U}=tabular_CPD(bnet2,U);
bnet2.CPD{W}=tabular_CPD(bnet2,W);
bnet2.CPD{B}=tabular_CPD(bnet2,B);
bnet2.CPD{D}=tabular_CPD(bnet2,D);

%手动构造得到的样本作为训练集代入learn_params()函数进行学习

bnet3=learn_params(bnet2,data);

%查看学习后的参数

CPT3=cell(1,N);
for i=1:N
    s=struct(bnet3.CPD{i});
    CPT3{i}=s.CPT;
end

fprintf('输出学习后的过劳死节点参数：\n');
dispcpt(CPT3{5});

%查看原来节点参数后的参数

CPT=cell(1,N);
for i=1:N
    s=struct(bnet.CPD{i});
    CPT{i}=s.CPT;
end

fprintf('输出真实的过劳死节点参数：\n');

dispcpt(CPT{5});
```