哈爾濱工業大學

人工智能实验报告

题　　目　贝叶斯网络

专　　业　计算机科学与技术学院

学　　号　1160300814

学　　生　姜思琪

指 导 教 师　李钦策

同 组 人 员　谢将凤、冯传恒、陈曦、丁明泽

一. 简介/问题描述

1.1 待解决问题的解释

实验 3 要求根据输入的文件,构建一个贝叶斯网络,然后根据 query 文件中的每一个条目求出条件概率的值

1.2 问题的形式化描述

参照课程第五部分讲授的贝叶斯网络完成，给定事件和事件之间的关系，并且给出每个事件的 CPT 图，根据贝叶斯公式根据上述条件求出目标概率，编写程序实现基于贝叶斯网络的推理。在这里用到的贝叶斯算法是建立在有向无环图和 CPT 表的技术上实现的。

首先，给定的输入文件格式为：

```
N
rv0 rv1 ... rvN-1
0   0   1   ... 0
1   0   0   ... 1
...
0   1   1   ... 0
mat0
mat1
...
matN-1
```

在这里:

- N 是贝叶斯网络中随机事件的数目
- rv 是随机事件的名字（字符串形式表示）
- mat 是一个二维数组，分别表示从他的父亲到其本身的可能性概率。第一个元素表示发生的概率，第二个元素表示不发生的概率，显然两个元素相加为 1

在上述中 mat 即为 CPT 表（Conditional Probability Table），其被设计为如下格式：

对于每个节点，如果他有 N 个父节点，则其 CPT 表中有 2^N 列，

我们记为标号 0 - （2^N-1），其行序号的定义方法如下，利用二进制分别表示对应的父亲为是否发生，1 为发生，0 位不发生，将得到的二进制数转化为十进制代表其对应的行号。举例如下:

A 有两个父节点 C，F，则 CPT 表如下表所示：

CPT entry	
P(A=*true*|C=*false*,F=*false*)	P(A=*false*|C=*false*,F=*false*)
P(A=*true*|C=*false*,F=*true*)	P(A=*false*|C=*false*,F=*true*)
P(A=*true*|C=*true*,F=*false*)	P(A=*false*|C=*true*,F=*false*)
P(A=*true*|C=*true*,F=*true*)	P(A=*false*|C=*true*,F=*true*)

其次，编写程序对应的查询格式为：P(rvQ | rvE1=val, rvE2=val, ...)

rvQ 表示查询的条件的名字，即在 rvE1=val, rvE2=val, ..发生的条件下，rvQ 发生的概率。

RvEx 表示条件的名称，而后面的 val 为 true/false，分别表示发生和不发生。

最后，输出格式为两个数据分别表示 P(QueryVar=true|...) 和 P(QueryVar=false|...))。

例如：

0.872 0.128

1.3 解决方案介绍（原理）

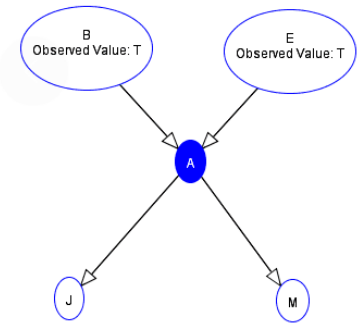
根据贝叶斯网络的原理,首先确定每个结点的父节点,然后根据各个节点之间的关系计算出每一个原子事件发生的概率,当需要计算某个事件发生的概率时,就从所有的原子事件中找出符合条件的,加在一起,这样就能某个特定事件的概率,从而求出条件概率

二. 算法介绍

2.1 所用方法的一般介绍

首先,根据输入的文件的第一行确定 N,接下来输入 N 个单词,每个单词对应了贝叶斯网络中的一个节点,所以我们建立一个 2^N 个元素的列表来存储每个原子事件发生的概率,刚开始时初始化为全 0,然后把第一个元素的值置为 1,代表 0 个结点的情况.

接下来输入贝叶斯网络的图关系,我们建立一个根据每行的输入,建立相互之间的关联,roots[i]中的每一个元素代表第 i 个结点的拓扑排序的前面的结点



比如这个贝叶斯网络,结点顺序为 B,E,A,J,M,用 0~4 依次表示.其中

roots[0]=[]

roots[1]=[]

roots[2]=[0, 1]

roots[3]=[0, 1, 2]

roots[4]=[0, 1, 2, 3]

然后输入 N 个概率表,当输入第 i 个表的时候,计算出前 i 个属性的联合分布概率,共需要存 2^i 个数,当输入第 i+1 个表时,根据贝叶斯网络中的公式,需要从新输入的概率表中找出要乘的条件概率数值的下标,我们可以构造一个函数 getChartIndex(seq, uppernodes, n),其中 seq 是某个事件发生概率的下标,根据其父节点的关系,以及 seq 中每个事件的真值,计算出下一个要乘的条件概率值再新输入的概率表中的下标,其具体计算过程是,将 seq 视为一个二进制数(seq<2^n),每一位分别表示前 n 个事件属性是否发生,然后根据前面的父子关系,对 seq 进行位运算,最后即可得到要成的条件概率的下标并且返回之.这样就可以求出所有原子事件的概率,得到一个概率表.

接下来就是通过解析查询语句来计算条件概率,首先使用 split()方法把输入字符串分为单个查询语句,然后分析查询语句,最后遍历所有概率,找出其中符合条件的概率,加在一起就是要求的概率,最后使用条件概率公式就可以得出答案.

2.2 算法伪代码

```
f ← 打开的文件;
输入 N;
totalChart ← 2^N 大小的数组;
totalChart[0] ← 1;
{    nameList ← f 中下一行字符串的分割   ;
     roots ← [1,2,3…N];
for i←0:N:
{          str_in ← f 中下一行字符串   ;
           row ← str_in 转化成的一维数组;
           for j in range(N):
           {
                if row[j]  等于  1:
                     roots[j].append(i)   ;
            }
     }
     n ← 0;
for i←0:N:
{          rowLen ← 1 << len(roots[i]);
           chart ← rowLen 个空列表组成的列表;
           for j←0:rowLen:
           {    str_in ← f 中下一行字符串;
                chart[j] ←  将 str_in 转化为一维数组;
           }
           tempChart ← totalChart;
           for i ←0:2^(n+1)
                totalChart[i] ← tempChart[i // 2] * chart[getChartIndex(i // 2, roots[n], n)][1 - i % 2];
           n ← n +1;
     }
sum ← 0;
```

```
关闭文件;
    f ← 打开文件 file2;
inputString ← f 中全部字符串;
# 得到其中的每个式子
    expressionList ← getExpression(inputString);
for expressionList 中的每个元素 entry:
{ # 此函数根据表达式计算标志位
        flagbitsAll, flagbitsCon, valuebits ← parseExpression(entry);
        pCon, pT, pF← 0, 0, 0;
        for i ←0:2^N
        {     if i & flagbitsCon 等于 valuebits:
            {   pCon ← totalChart[i] + pCon
                if i & flagbitsCon 等于 i & flagbitsAll:
                    pF ← totalChart[i] + pF
                else:
                    pT ← totalChart[i] + pF
            }
        }
        输出 pF/pCon,   pT/pCon
}
关闭文件 f
```

三. 算法实现

3.1 实验环境与问题规模

操作系统: Windows 10 专业版

处理器: Intel(R) Core(TM) i5-6200U

语言: Python2.7

编辑器: Sublime Text

问题规模: 不超过 15 个结点

3.2 数据结构

一个 2^n 个元素的列表 totalChart,里面存的是只考虑前 n 个属性时,每个原子事件的概率,将其下标转换为二进制即可得到每个时间的真值,当 n=N 时,该贝叶斯网络的联合概率分布表就形成了

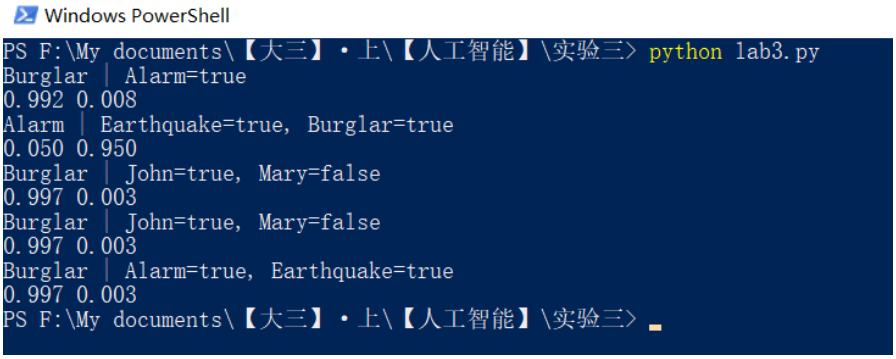
一个二维列表 roots,第 i 个列表里存的是第 i 个节点的在该贝叶斯网络拓扑排序前面的所有结点

3.3 实验结果

程序通过解析 query 文件中的表达式,求出了每个表达式中的条件概率,然后我们根据 beyes.jar 程序验证了我们得到的结果,结果在保留三位小数的情况下完全一致.

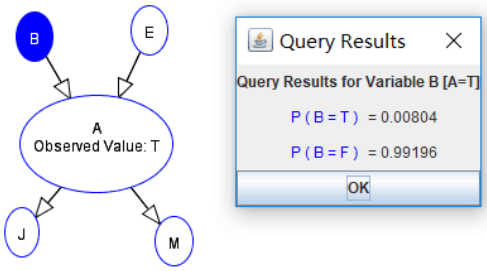
3.4 系统中间及最终输出结果（要求有屏幕显示）

以下是我们程序读取 qurey 文件得到的结果：

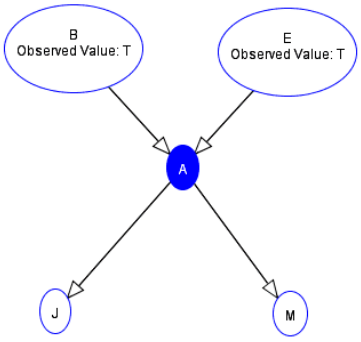


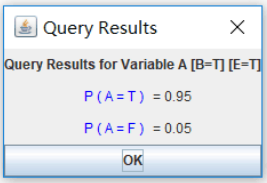
其中第一个是为 false 的概率,第二个是为 true 的概率。然后我们可以用 beyes.jar 来验证:

第一个问题:

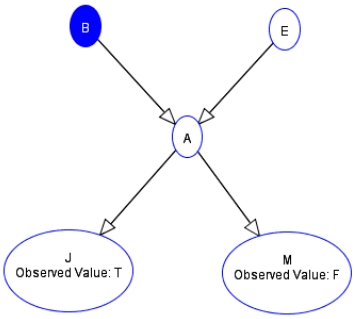


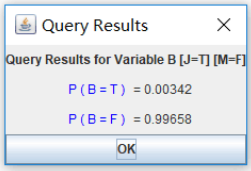
第二个问题:





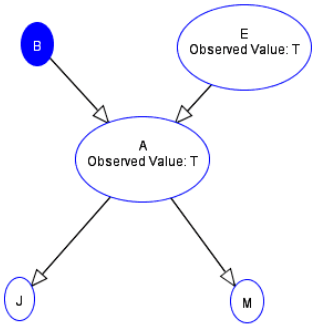
第三个问题：

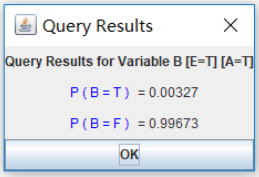




第四个问题在这里和第三个问题一样，不再赘述

第五个问题：





以上数据和我们写的程序求得的数据在保留三位小数的情况下是相等的，因此可以证明我们写的程序是正确的

四. 总结及讨论

通过本次实验，我更深刻了解到了贝叶斯网络及其使用，可以根据 query 文件中的每一个条目求出条件概率的值，可以求出目标概率。

本次实验中，使用到了有向无环图和 CPT 表的技术。这是我第一次使用它们进行编程。在同组队员的帮助下，我克服了理解上的困难，感谢团队的力量。