

# 人工智能

姓名：孟衍璋 学号：16337183

## 一、实验目的

利用贝叶斯网络来接解决蒙提霍尔三门问题，（出自美国的电视游戏节目。问题的名字来自该节目的主持人蒙提•霍尔）。贝叶斯网络工具包采用 eBay 的 Bayesian-belief-networks，其为一个贝叶斯网络的 python 工具包。安装方式：`$ git clone https://github.com/eBay/bayesian-belief-networks`。

## 二、问题定义

参赛者会看见三扇关闭了的门，其中一扇的后面有一辆汽车，选中后面有车的那扇门可赢得该汽车，另外两扇门后面则各藏有一只山羊。

当参赛者选定了一扇门，但未去开启它的时候，节目主持人开启剩下两扇门的其中一扇，露出其中一只山羊（主持人不会打开有车的那扇门）。主持人其后会问参赛者要不要换另一扇仍然关上的门。问题是：换另一扇门会否增加参赛者赢得汽车的机率？

## 三、实验要求

利用贝叶斯网络工具包，编写程序，计算换门、不关门两种不同情况下参赛者赢得汽车的机率。要求程序附有详细注释。

## 四、实验步骤

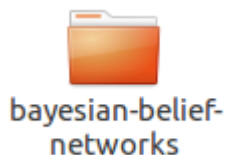
### 1. 实验环境准备

做这个实验之前，首要的事情就是，安装好实验要求的包。刚开始就遇到了问题，实验给出的 Bayesian-belief-networks 工具包是基于 python2 的，而自己的电脑是安装的 python3，并不兼容。于是便准备在虚拟机里完成实验。

先在命令行输入：

```
git clone https://github.com/eBay/bayesian-belief-networks
```

便可以得到文件：



在命令行中进入该文件夹目录，根据其中的 README.txt 文件的指示，使用以下命令安装：

```
$ python setup.py install
```

安装好之后按照指示进入教程：

```
$ pip install sphinx  
$ cd docs/tutorial  
$ make clean  
$ make html
```

在执行到这个地方的时候埋下了一个坑，用 pip 安装好 sphinx 包之后，系统提醒需不需要更新 pip 的版本，更新之后，再用 pip 安装包的时候，便报出了如下错误：

```
Traceback (most recent call last):
  File "/usr/bin/pip", line 9, in <module>
    from pip import main
ImportError: cannot import name main
```

经过查询之后，找到了解决办法：

解决：pip文件在usr/bin目录下，cd进去，进行以下修改

把下面的三行

```
1 | from pip import main
2 | if __name__ == '__main__':
3 |     sys.exit(main())
```

换成下面的三行

```
1 | from pip import __main__
2 | if __name__ == '__main__':
3 |     sys.exit(__main__._main())
```

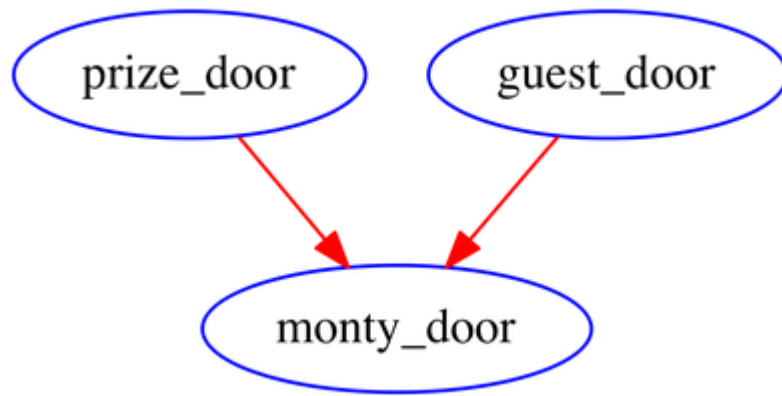
在进入相应目录修改 pip 文件内容时，又遇到了一个问题，pip 文件是只读模式的，还要根据以下命令进行修改：

```
myz@ubuntu:/usr/bin$ sudo chmod 777 pip
```

于是终于解决了这个问题。

## 2. 实验问题分析

在该工具包中，将每个事件作为变量来建立模型。对于蒙提霍尔三门问题，需要建立三个变量，他们之间的关系如下：



其中每个变量的域都是三个门，假设分别为 A、B、C。

在程序中，每个变量用一个函数来表示：

```
def f_prize_door(prize_door):  
    return 0.33333333  
  
def f_guest_door(guest_door):  
    return 0.33333333  
  
def f_monty_door(prize_door, guest_door, monty_door):  
    if prize_door == guest_door: # 参赛者选择了藏有汽车的门  
        if prize_door == monty_door:  
            return 0 # Monty永远不可能选到有汽车的门  
        else:  
            return 0.5 # 剩下的两个门Monty任选一个  
    elif prize_door == monty_door:  
        return 0 # Monty永远不会选到有汽车的门  
    elif guest_door == monty_door:  
        return 0 # Monty永远不会选择与参赛者相同的门  
    else:  
        return 1 # 参赛者没有选择到有汽车的门，Monty在剩下的两个门当中只能选剩下的没有汽车的门
```

再从刚安装的工具包中导入文件：

```
from bayesian.bbn import build_bbn
```

主体运行部分：

```
if __name__ == '__main__':  
    g = build_bbn(f_prize_door, f_guest_door, f_monty_door,  
        domains = dict(prize_door = ['A', 'B', 'C'],  
            guest_door = ['A', 'B', 'C'],  
            monty_door = ['A', 'B', 'C']))
```

保存文件为 monty\_hall.py，在命令行中运行：

```
nyz@ubuntu:~/Desktop$ python -i monty_hall.py
```

进入 python 交互式界面，再输入：

```
>>> g.q()
+-----+-----+-----+
| Node          | Value | Marginal |
+-----+-----+-----+
| guest_door    | A     | 0.333333 |
| guest_door    | B     | 0.333333 |
| guest_door    | C     | 0.333333 |
| monty_door    | A     | 0.333333 |
| monty_door    | B     | 0.333333 |
| monty_door    | C     | 0.333333 |
| prize_door    | A     | 0.333333 |
| prize_door    | B     | 0.333333 |
| prize_door    | C     | 0.333333 |
+-----+-----+-----+
```

这是在我们没有向模型提供任何信息的情况下，所以这些概率均为  $1/3$ 。

接下来向模型提供信息：

```
>>> g.q(guest_door = 'A')
+-----+-----+-----+
| Node          | Value | Marginal |
+-----+-----+-----+
| guest_door    | B     | 0.000000 |
| guest_door    | C     | 0.000000 |
| guest_door*   | A*    | 1.000000 |
| monty_door    | A     | 0.000000 |
| monty_door    | B     | 0.500000 |
| monty_door    | C     | 0.500000 |
| prize_door    | A     | 0.333333 |
| prize_door    | B     | 0.333333 |
| prize_door    | C     | 0.333333 |
+-----+-----+-----+
```

这是假设参赛者打开了 A 这道门，那么其他的概率会如何变化。参赛者已经打开了 A，这是我们观察到的，所以他打开 B 和 C 的概率均为 0，而根据游戏规则，monty 不可能会打开 A，他打开 B 和 C 的概率均为 0.5。由于没有提供其他信息，所以每扇门可能有汽车的概率还是  $1/3$ 。

下面再加入 monty 打开某扇门的信息：

```
>>> g.q(guest_door = 'A', monty_door = 'B')
```

Node	Value	Marginal
guest_door	B	0.000000
guest_door	C	0.000000
guest_door*	A*	1.000000
monty_door	A	0.000000
monty_door	C	0.000000
monty_door*	B*	1.000000
prize_door	A	0.333333
prize_door	B	0.000000
prize_door	C	0.666667

可以看出，如果参赛者选择不换门，那么门后有车的概率是 1/3，如果他选择换门，门后有车的概率是 2/3。

## 五、实验心得

其实本次实验的难度不是太高，最主要的就是要能将提供的工具包安装上并导入教程进行学习。

最先遇到的问题就是，该工具包是基于 python2 的，与 python3 并不兼容，于是便选择在虚拟机中安装该工具包。在中间由于 pip 更新出了些小问题，不过通过查询资料得到了解决。

这次实验得出的结果与直觉判断相反，直觉判断会觉得参赛者换门与不换门不会有差异，但实际上换门之后得到奖励汽车的概率是不换门的两倍。

总而言之，这次实验让我知道了如何使用类似的工具包，更加理解了贝叶斯网络的概念，收获良多。