《人工智能》

实验报告

班级：计算机 86

姓名：常傲

学号：2182212573

日期：2020 年 11 月 24 日

利用主观贝叶斯方法进行不确定性推理

计算机 86 班　常傲　2182212573

摘要（Abstract）：不确定性可以理解为在缺少足够信息的情况下做出判断，是智能问题的本质特征；推理是人类的思维过程，它是从已知事实出发，通过运用相关的知识逐步推出某个结论的过程。所谓不确定性推理原理就是从不确定性初始证据出发，通过运用不确定性的知识，最终推出具有一定程度的不确定性但却是合理或者近乎合理的结论的方法或理论。本次实验使用 matlab 编程实现关于不确定性关系及其推论的研究。

关键词（Keywords）：贝叶斯、不确定性关系。

实验原理：

贝叶斯推断是一种统计学方法，用来估计统计量的某种性质。

贝叶斯推断与其他统计学推断方法截然不同。它建立在主观判断的基础上，也就是说，你可以不需要客观证据，先估计一个值，然后根据实际结果不断修正。正是因为它的主观性太强，曾经遭到许多统计学家的诟病。

贝叶斯推断需要大量的计算，因此历史上很长一段时间，无法得到广泛应用。只有计算机诞生以后，它才获得真正的重视。人们发现，许多统计量是无法事先进行客观判断的，而互联网时代出现的大型数据集，再加上高速运算能力，为验证这些统计量提供了方便，也为应用贝叶斯推断创造了条件，它的威力正在日益显现。

我们把 P(A)称为"先验概率"，即在 B 事件发生之前，我们对 A 事件概率的一个判断。P(A|B)称为"后验概率"，即在 B 事件发生之后，我们对 A 事件概率的重新评估。P(B|A)/P(B)称为"可能性函数"，这是一个调整因子，使得预估概率更接近真实概率。

所以，条件概率可以理解成下面的式子：

后验概率＝先验概率 x 调整因子

实验过程：

输入 P(E),P(H),LS,LN,分一下几种情况：

1. 几率函数

$$O(x)=\frac{P(x)}{1-P(x)}$$

2.

$P(E)=P\left(\frac{E}{S}\right)=1$，由贝叶斯公式可知：

$$P\left(\frac{H}{E}\right)=P\left(\frac{E}{H}\right)\times P(H)/P(E)$$

同理有：

$$P\left(\frac{\neg H}{E}\right)=P(\frac{E}{\neg H})\times P(H)/P(E)$$

将以上两式相除可得

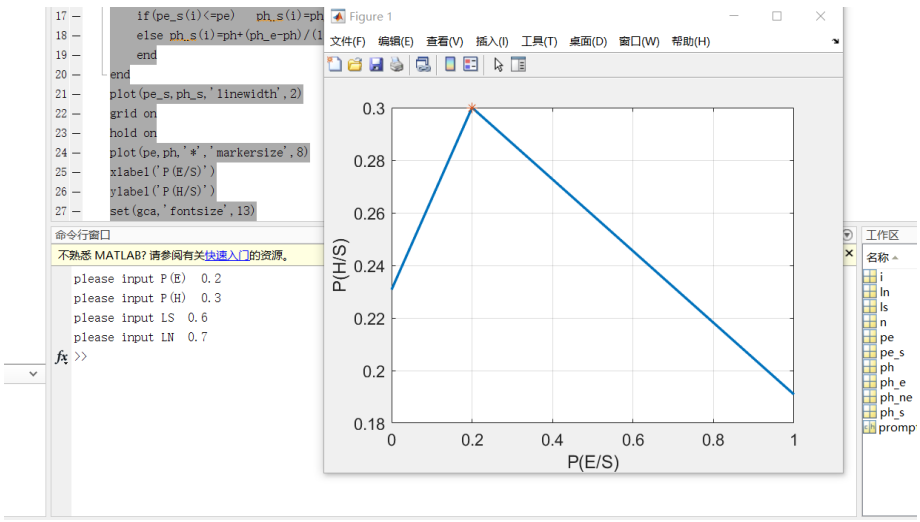
$$P\left(\frac{H}{E}\right)=\frac{LS\times P(H)}{(LS-1)\times P(H)+1}$$

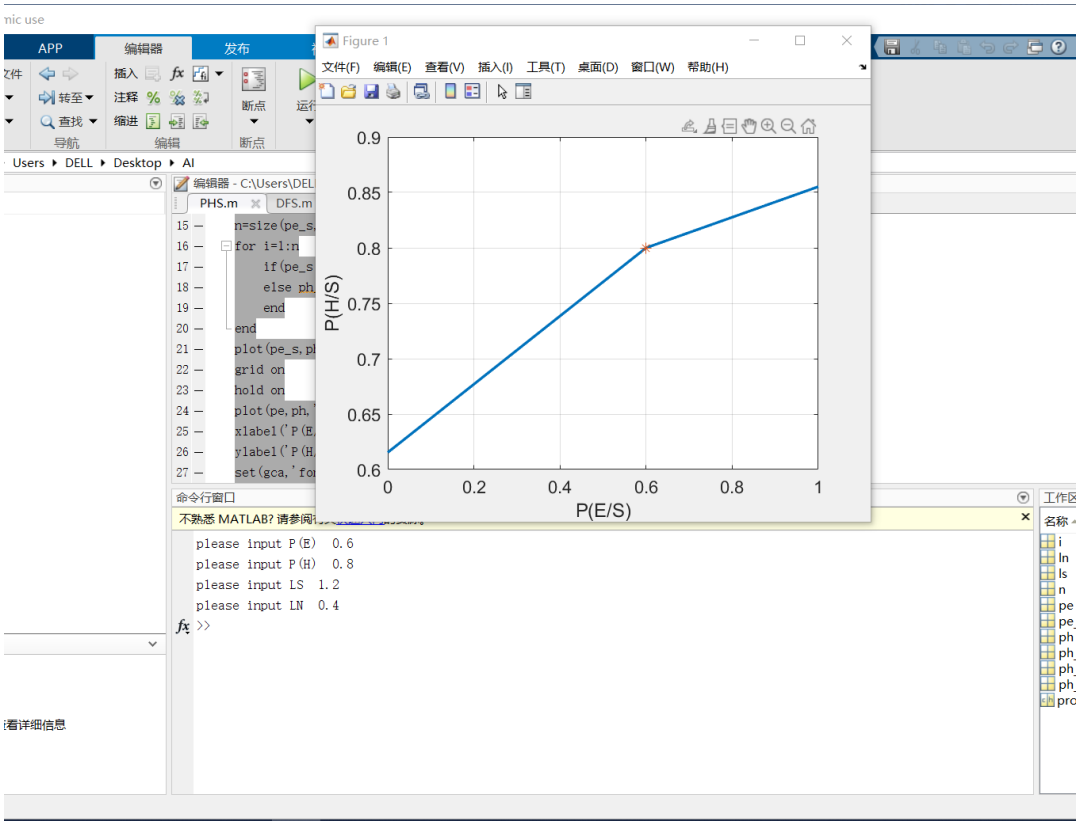
其余情况下同理

实验代码：

```
clear
clc
close
prompt='please input P(E)  ';
pe=input(prompt);
prompt='please input P(H)  ';
ph=input(prompt);
prompt='please input LS  ';
ls=input(prompt);
prompt='please input LN  ';
ln=input(prompt);
ph_ne=ph*ln/(1-ph+ph*ln);
ph_e=ph*ls/(1+ph*(ls-1));
pe_s=0:0.001:1;
n=size(pe_s,2);
for i=1:n
    if(pe_s(i)<=pe)   ph_s(i)=ph_ne+(ph-
ph_ne)/pe*pe_s(i);
    else ph_s(i)=ph+(ph_e-ph)/(1-ph)*(pe_s(i)-pe);
    end
end
plot(pe_s,ph_s,'linewidth',2)
grid on
hold on
plot(pe,ph,'*','markersize',8)
xlabel('P(E/S)')
ylabel('P(H/S)')
set(gca,'fontsize',13)
```

实验结果：





[1] 鲍军鹏,张选平.人工智能导论[M].机械工业出版社:北京,2019:93.

实验二：重排九宫

摘要（Abstract）：重排九宫就是重新排列九宫图的意思。这是根据当时盛行研究的数学游戏——纵横图（也叫幻方或魔方阵）发展来的。在使用算术的同时，还必须推动方块使其到相对应的位置。其玩法是在 3×3 方格盘上，放有 1-8 八个数，剩下一格为空，每一空格其周围的数字可移至空格。先设定初始排列数字，然后开始思考如何以最少的移动次数来达到目的。

关键词（Keywords）：重排九宫问题、A*路径算法、深度优先、广度优先。

实验原理：

九宫格中乱序填入 1~8 八个数字，相邻两个数字的位置是可以交换的，通过交换这些数字的位置，将空格移动到指定位置。解决的方法不止一种，请你给出移动最少步的那个解法。

实验算法：

使用康托展开（X=a[n]*(n-1)!+a[n-1]*(n-2)!+...+a[i]*(i-1)!+...+a[1]*0! 其中，a[i]为整数，并且 X=a[n]*(n-1)!+a[n-1]*(n-2)!+...+a[i]*(i-1)!+...+a[1]*0!），将给出的式子展开后观察，分别进行深度优先和广度优先进行寻找算法。

实验代码：

1.深度优先：

```
clc
clear
close
s0=textread('start.txt');
target=[1 2 3;8 0 4;7 6 5];
```

```
open={s0};
father={s0};%ÒÑ±éÀú½áµã
treenode={};%½áµã
node=0;%½áµãÖ¸Õë
n=0;%¼ÓÈëclose±íµÄ½áµãÊý
count=1;%±»À©Õ¹³öµÄ½áµãÊý
while(1)
    close=open(1);
    open(1)=[];
    treenode=[treenode close];
    n=n+1;
    if close{1}==target;
         close{1}
         "The search tree has been found, it is
printing now"
        break
    else
        y=extend(close{1});
        yy=cellxor(y,father);
        mem(n)=size(yy,2);
        father=[father,yy];
        if(n==1)
            rootnode=1;
            node(count+1:count+size(yy,2))=1;
        elseif(n==2)
            rootnode=2;
            node(count+1:count+size(yy,2))=rootnode;
        else
            rootnode=rootnode+mem(n-2);
            node(count+1:count+size(yy,2))=rootnode;
        end
        count=count+size(yy,2);
        open=[yy open];
    end
end
cm=zeros(count,count);
for i=2:count
    cm(node(i),i)=1;
end
for i=1:count
temp=father{i};
temp=temp';
num(i,:)=temp(:);
id{i}=[num2str(num(i,1:3)) ' # ' num2str(num(i,4:6))
```

```
' # ' num2str(num(i,7:9))];
end
bg=biograph(cm,id);
view(bg)
```

2. 广度优先：

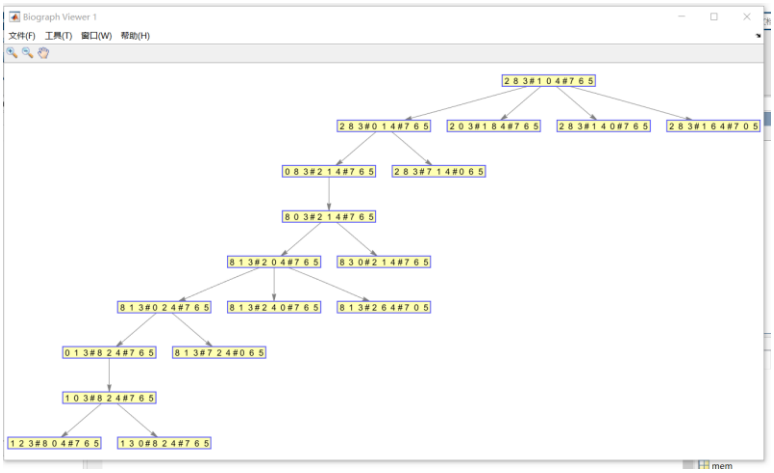
```
clc
clear
close
s0=textread('start.txt');
target=[1 2 3;8 0 4;7 6 5];
open={s0};
father={s0};
node=0;
n=0;
count=1;
while(1)
    close=[];
    close=open(1);
    open(1)=[];
    n=n+1;
    if close{1}==target;
        close{1}
        "The search tree has been found, it is
printing now"
        break
    else
        y=extend(close{1});
        yy=cellxor(y,father);
        father=[father,yy];
        node(count+1:count+size(yy,2))=n;
        count=count+size(yy,2);
        open=[open yy];
    end
end
cm=zeros(n,n);
for i=2:n
    cm(node(i),i)=1;
end
for i=1:n
temp=father{i};
temp=temp';
num(i,:)=temp(:);
id{i}=[num2str(num(i,1:3)) ' # ' num2str(num(i,4:6))
```

```
' # ' num2str(num(i,7:9))];
end
bg=biograph(cm,id);
view(bg)
```

实验结果：

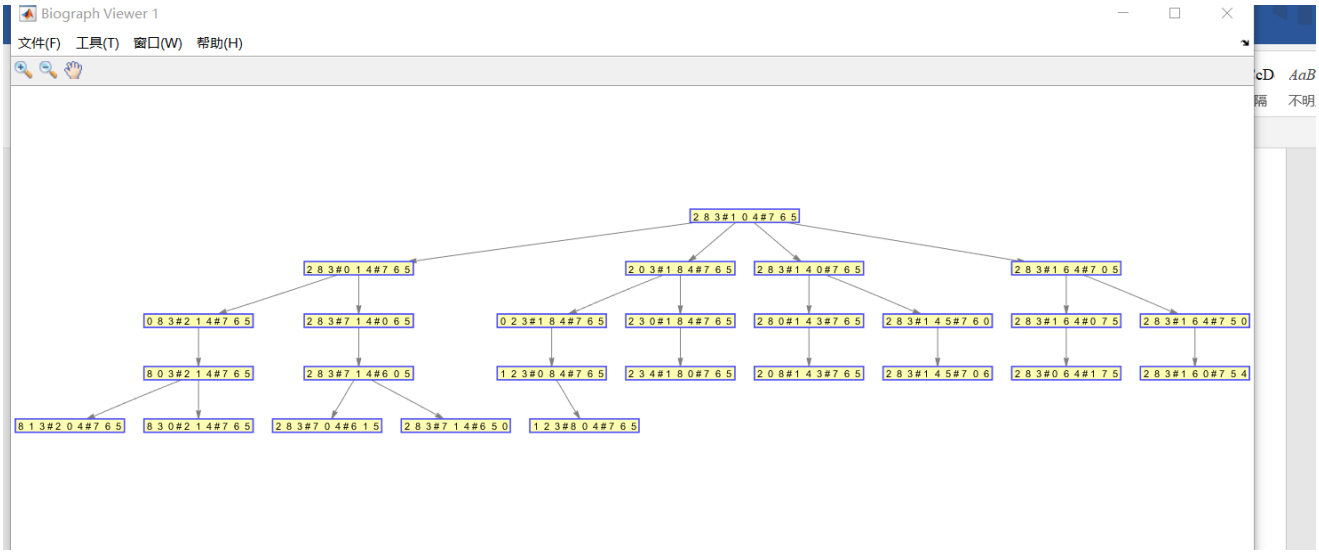
1. 深度优先：

```
ans =

     1     2     3
     8     0     4
     7     6     5


ans =

    "The search tree has been found, it is printing now"

fx >>
```



3. 广度优先：

```
不熟悉 MATLAB? 请参阅有关快速入门的资源。

ans =

     1     2     3
     8     0     4
     7     6     5


ans =

    "The search tree has been found, it is printing now"

fx >>
```



[1] 鲍军鹏,张选平.人工智能导论[M].机械工业出版社:北京,2019:93.