人工智能实验报告

主观贝叶斯方法

课程名称：	人工智能
姓名：	计算机 001 班曾锦程
学院：	电信学部
专业：	计算机科学与技术
学号：	2203613040
指导老师：	相明

2023 年 1 月 27 日

西安交通大学实验报告

专业：计算机科学与技术
姓名：计算机 001 班曾锦程
学号：2203613040
日期：2023 年 1 月 27 日
地点：

课程名称：	人工智能	指导老师：	相明	成绩：	
实验名称：	主观贝叶斯方法	实验类型：	设计实验	同组学生姓名：	无

一、 问题描述

运用贝叶斯公式进行不确定性推理，必然受到贝叶斯公式运用条件的限制。事实上，事件之间彼此独立的要求很苛刻的，在现实中往往不能保证这个条件被严格满足。而且在贝叶斯公式中还要求事先知道已知结论时前件的条件概率和结论的先验概率。要获得这些概率，就必须做一些统计工作。然而，在实践中未必能进行足够的重复实验来获得充分的观察数据。再者，用贝叶斯公式得到的后验概率实际上是对先验概率的修正。

假如先验概率偏差比较大，那么必然会对后验概率造成不良影响。所以在人工智能实践中，为了应用简便和省事，往往用主观决定代替客观观察，用主观指定的数值来代替统计概率。主观贝叶斯方法就是这种思想的一种体现。主观贝叶斯方法是由杜达等人于 1976 年在贝叶斯公式基础上进行改进而提出的一种不确定性推理模型。

在证据不确定的情况下，根据充分性量度 LS、必要性量度 LN、E 的先验概率 $P(E)$ 和 H 的先验概率 $P(H)$ 作为前提条件，分析 $P(H|S)$ 和 $P(E|S)$ 的关系。

二、 算法描述

1. 实验原理

(1) 证据不确定性的表示

(a) 在主观 Bayes 方法中，证据的不确定性用概率表示。对于证据 E，由用户根据观察 S 给出 $P(E|S)$，即动态强度。用 $P(E|S)$ 描述证据的不确定性（证据 E 不是可以直接观测的）。

(b) 证据肯定存在时，$P(E|S)=1$

(c) 证据肯定不存在时，$P(E|S)=0$

(d) 证据具有不确定性时，$0<P(E|S)<1$

(2) LS 与 LN 的性质

(a) 充分性度量 (LS) 的定义:

$$LS=\frac{P(E|H)}{P(E|\neg H)}$$

(b) 必要性度量 (LS) 的定义:

$$LN=\frac{P(\neg E|H)}{P(\neg E|\neg H)}$$

(c) LS>1: 表明证据 E 是对 H 有利的证据。

LN>1：表明证据 ¬E 是对 H 有利的证据。

所以：不能出现 LS>1 且 LN>1 的取值。

(d) LS<1: 表明证据 E 是对 H 不利的证据。

LN<1：表明证据 ¬E 是对 H 不利的证据。

所以：不能出现 LS<1 且 LN<1 的取值。

(e) 一般情况下，取 LS>1，LN<1，LS 和 LN 的值由领域专家给出，代表知识的静态强度。

(f) 本实验中 LS 与 LN 主要是影响 $P(H|E)$ 和 $P(H|\neg E)$，可以不考虑 LS 与 LN，直接使用两个概率计算。

(g) 值得注意的是，由于 LS 和 LN 性质的原因，$P(H)$ 应该在 $P(H|E)$ 与 $P(H|\neg E)$ 之间。

(3) 证据不确定的情况

(a) 当 $0 < P(E|S) < 1$ 时，可以证明只有在确定 P(E) 的情况下 [1]:

$$P(H|S) = P(H|E) \times P(E|S) + P(H|\neg E) \times P(\neg E|S)$$

(b) 当 $P(E|S) = 1$ 时，证据肯定存在，此时:

$$P(H|S) = P(H|E)$$

(c) 当 $P(E|S) = 0$ 时，证据肯定不存在，此时:

$$P(H|S) = P(H|\neg E)$$

(d) 当 $P(E|S) = P(E)$ 时，证据 E 与观察 S 无关。由全概率公式得：

$$P(H|S) = P(H|E) \times P(E) + P(H|\neg E) \times P(\neg E) = P(H)$$

(e) 当 $P(E|S)$ 为其它值时，由 (a) 分析可知 $P(H|S)$ 与 $P(E|S)$ 成线性关系，可通过分段线性插值计算 $P(H|S)$（原本使用 (a) 中公式即可计算，但是因为专家给出的 $P(E)$ 并不是准确值，$P(H)$ 与 $P(E)$ 不能一致满足公式 (d)，真实情况如图 1 所示，所以使用分段线性插值强行一致，这样的近似估计方法还有很多）[1] 即:

$$P(H|S) = \begin{cases} P(H|\neg E) + \dfrac{P(H) - P(H|\neg E)}{P(E)} \times P(E|S) & 0 \le P(E|S) < P(E) \\ P(H) + \dfrac{P(H|E) - P(H)}{1 - P(E)} \times [P(E|S) - P(E)]) & P(E) \le P(E|S) \le 1 \end{cases}$$

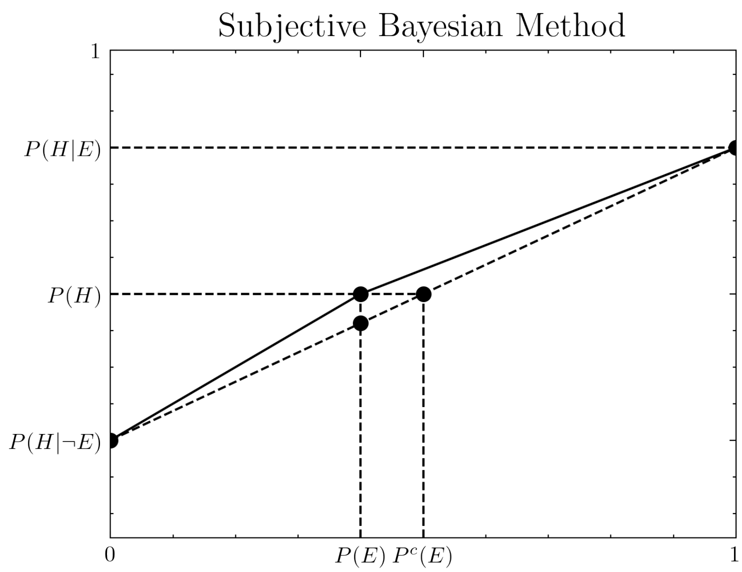


图 1: 分段插值

(4) 应用

对于知识：

$$IF \quad E \quad THEN(LS, LN) \quad H(P(H))$$

给定证据 E 的不确定性度量 P(E|S)，则结论的可信度可以表示为：

$$P(H|S) = \begin{cases} P(H|\neg E) + \dfrac{P(H) - P(H|\neg E)}{P(E)} \times P(E|S) & 0 \leq P(E|S) < P(E) \\ P(H) + \dfrac{P(H|E) - P(H)}{1 - P(E)} \times [P(E|S) - P(E)]) & P(E) \leq P(E|S) \leq 1 \end{cases}$$

该公式称为 EH 公式。用 C(E/S) 代替 P(E/S)，可得到等价的 CP 公式。

2. 实验步骤

(1) 编写程序，设置好参数，完成分段插值过程

(2) 画出图形

(3) 观察结果。

三、 实验结果

1. 分段函数图像

设置 $P(H) = 0.5, P(E) = 0.6, P(H|E) = 0.8, P(H|\neg E) = 0.4$, 分段函数如图 2 所示。

2. 多组参数测试图像

在此基础上，使用 matplotlib 中的 `subplot` 函数进一步画出在不同概率取值（使用 `random`）情况下分段函数图，如图 2 所示，多组参数标于子图标题。

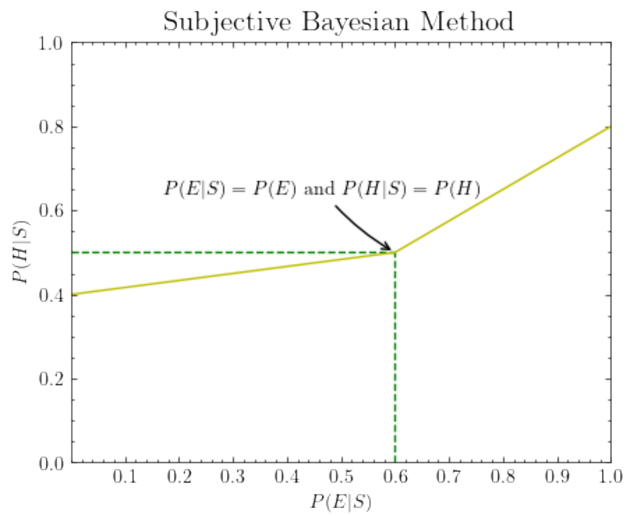


图 2: 主观贝叶斯方法

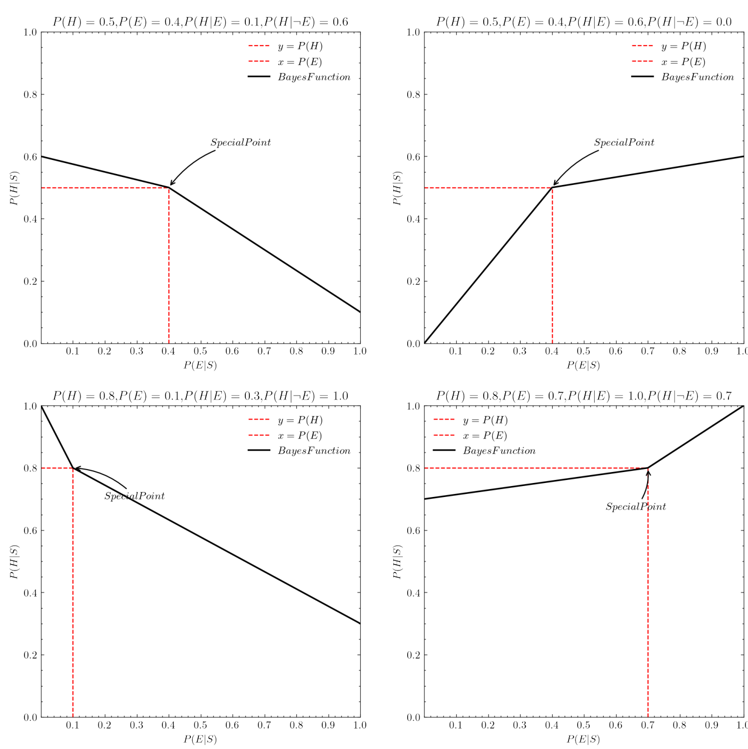


图 3: 多组参数测试

四、 实验结果分析

1. $P(H|E)$ 与 $P(H|\neg E)$ 对函数影响

可以从图 3 中发现，$P(H|E)$ 是分段函数在 $P(E|S) = 1$ 时的取值，$P(H|\neg E)$ 是分段函数在 $P(E|S) = 0$ 时的取值，所以 $P(H|E)$ 与 $P(H|\neg E)$ 主要是影响函数图像左右两点的高低从而影响函数正负相关关系，而且这两点代表了证据不确定情况中的情况 (b) 和 (c)，与 $P(H)$ 和 $P(E)$ 共同决定特殊点左右两段的斜率大小。

2. $P(H)$ 与 $P(E)$ 对函数影响

可以从图 3 中发现 $P(H)$ 和 $P(E)$ 决定的分段点的位置，并且与 $P(H|E)$ 和 $P(H|\neg E)$ 共同决定共同决定特殊点左右两段的斜率，同时特殊点代表了证据不确定情况中的情况 (d)。

五、 源代码

Step0:引入必要python库

numpy、matplotlib、scienceplots、random

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import scienceplots
import random
```

Step1:设置必要参数

设置参数
$P(H), P(H|E), P(E), P(H|\neg E)$

In [2]:

```
PH = 0.5
PHE = 0.8
PE = 0.6
PH_E = 0.4
```

定义自变量
$0 \leqslant P(E|S) \leqslant 1$

In [3]:

```
PES = np.arange(0,1.1,0.1) #自变量P(E/S)
```

Step2:计算$P(H|S)$

根据公式写出计算$P(H|S)$的函数
(1)当$0 \leqslant P(E|S) < P(E)$时:

$$P(H|S) = P(H|\neg E) + \frac{P(H) - P(H|\neg E)}{P(E)} \times P(E|S)$$

(2)当$P(E) \leqslant P(E|S) \leqslant 1$时:

$$P(H|S) = P(H) + \frac{P(H|E) - P(H)}{1 - P(E)} \times [P(E|S) - P(E)]$$

In [4]:

```
def func(PES, PH, PE, PHE, PH_E):
    z1 = PES >= 0
    z2 = PES < PE
    z3 = PES >= PE
    z4 = PES <=1
    result1 = PH_E + (PH-PH_E)/PE * PES
    result1 = result1 * (z1 & z2)
    result2 = PH + (PHE - PH)/(1 - PE) * (PES - PE)
    result2 = result2 * (z3 & z4)
    result = result1 + result2
    return result
```

计算出$P(H|S)$

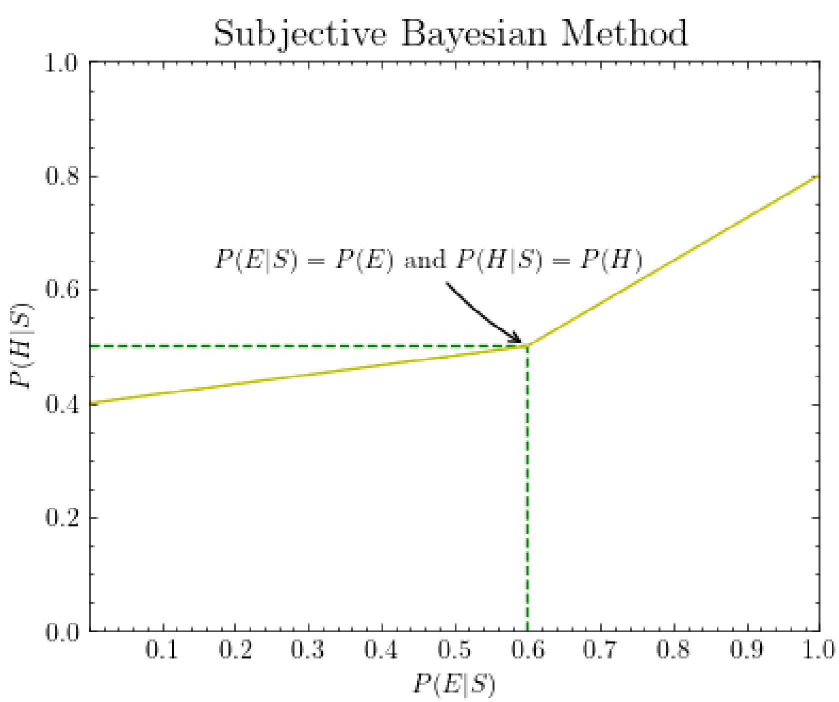
In [5]:

```
PHS = func(PES, PH, PE, PHE, PH_E)
```

Step3:使用matplotlib画图

In [6]:

```
plt.style.use('science')
plt.figure(figsize=(5,4),dpi=100)
plt.ylabel('$P(H|S)$',size=10)
plt.xlabel('$P(E|S)$',size=10)
plt.xlim(0,1)
plt.ylim(0,1)
plt.gca().xaxis.set_major_locator(plt.MaxNLocator(prune='lower'))
plt.title('Subjective Bayesian Method',size=15)
plt.annotate(r"$P(E|S)=P(E)$ and $P(H|S) = P(H)$",xy=(PE,PH),xycoords='data',xytext=(-120,30),
                 textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle="arc3,ra
x1 = np.arange(0,PE+0.1,0.1)
y1 = np.ones(x1.shape)*PH
y2 = np.arange(0,PH+0.1,0.1)
x2 = np.ones(y2.shape)*PE
plt.plot(x1,y1,color='g',linestyle='--')
plt.plot(x2,y2,color='g',linestyle='--')
plt.plot(PES,PHS,color='y')
plt.savefig('C:/Users/Lenovo/Desktop/XJTU-latex-report-master/XJTU-latex-report-master/1.png')
plt.show()
```



Step4:使用多组值测试

In [7]:

```
PH = []
PHE = []
PE = []
PH_E = []
while len(PH) <= 3:
    PH1 = random.randint(0,10)/10
    PHE1 = random.randint(0,10)/10
    PE1 = random.randint(1,9)/10
    PH_E1 = random.randint(0,10)/10
    if (PH1 > PHE1 and PH1 < PH_E1) or (PH1 < PHE1 and PH1 > PH_E1):
        PH.append(PH1)
        PHE.append(PHE1)
        PE.append(PE1)
        PH_E.append(PH_E1)
print('PH:')
print(PH)
print('PE:')
print(PE)
print('PHE:')
print(PHE)
print('PH_E:')
print(PH_E)
```

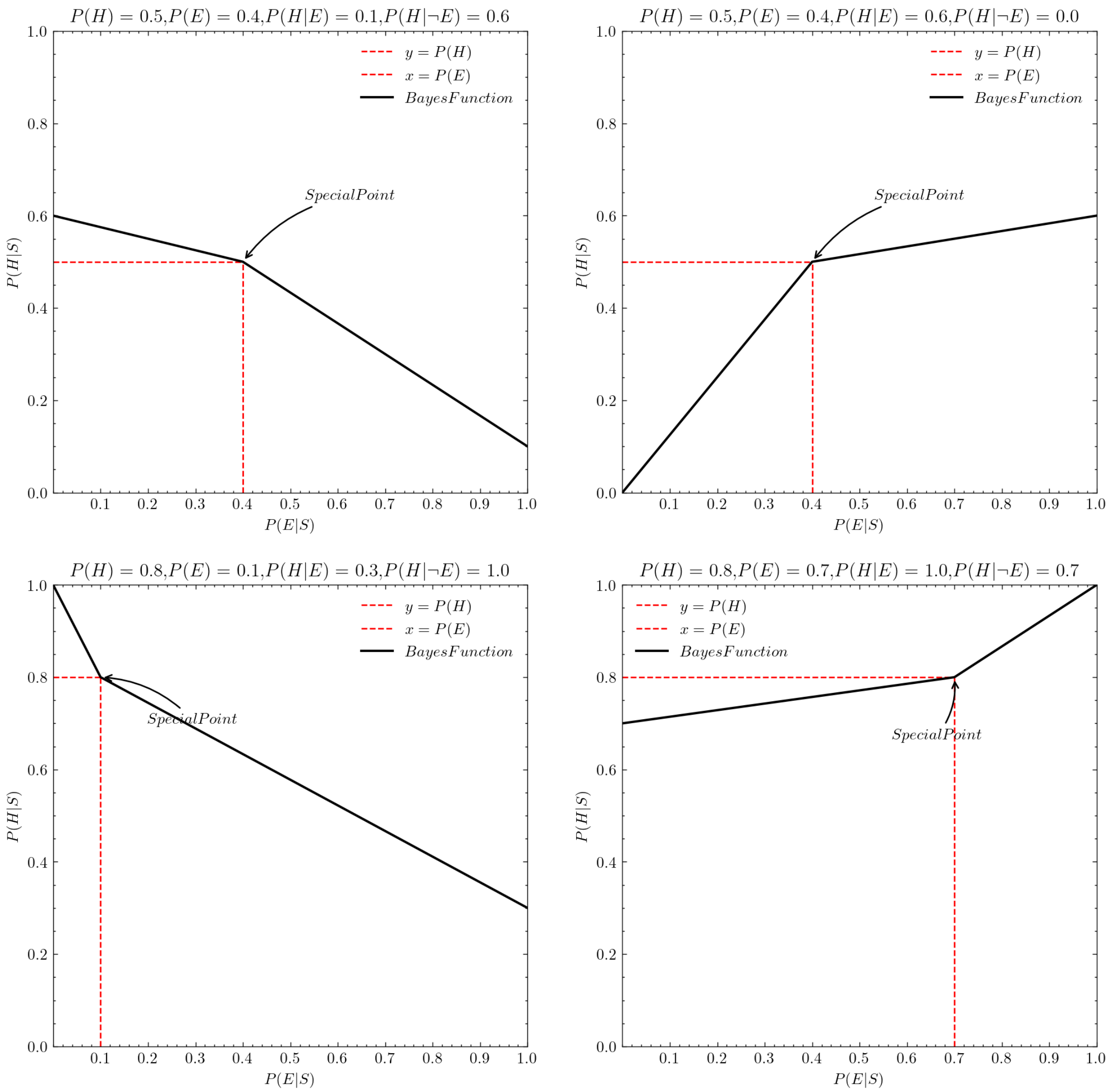
```
PH:
[0.5, 0.5, 0.8, 0.8]
PE:
[0.4, 0.4, 0.1, 0.7]
PHE:
[0.1, 0.6, 0.3, 1.0]
PH_E:
[0.6, 0.0, 1.0, 0.7]
```

In [8]:

```
PHS = []
for i in range(len(PH)):
    PHS.append(func(PES,PH[i],PE[i],PHE[i],PH_E[i]))
```

In [9]:

```
plt.figure(figsize=(12,12),dpi=200)
plt.subplots_adjust(hspace=0.2, wspace=0.2)
for i in range(4):
    plt.style.use('science')
    number = int('22'+ str(i+1))
    plt.subplot(number)
    plt.title('$P(H)=$ '+str(PH[i])+','+'$P(E)=$ '+str(PE[i])+','+'$P(H|E)=$ '+str(PHE[i])+','+'$P(
    plt.ylabel('$P(H|S)$',size=10)
    plt.xlabel('$P(E|S)$',size=10)
    plt.xlim(0,1)
    plt.ylim(0,1)
    plt.gca().xaxis.set_major_locator(plt.MaxNLocator(prune='lower'))
    if PH[i] <=0.5 and PE[i] <=0.5:
        plt.annotate(r"$Special Point$",xy=(PE[i],PH[i]),xycoords='data',xytext=(40,40),
                     textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle=
    elif PH[i] > 0.5 and PE[i] > 0.5:
        plt.annotate(r"$Special Point$",xy=(PE[i],PH[i]),xycoords='data',xytext=(-40,-40),
                     textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle=
    elif PH[i] > 0.5 and PE[i] <= 0.5:
        plt.annotate(r"$Special Point$",xy=(PE[i],PH[i]),xycoords='data',xytext=(30,-30),
                     textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle=
    elif PH[i] <= 0.5 and PE[i] > 0.5:
        plt.annotate(r"$Special Point$",xy=(PE[i],PH[i]),xycoords='data',xytext=(-30,30),
                     textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle=
    x1 = np.arange(0,round(PE[i]+0.1,1),0.1)
    y1 = np.ones(x1.shape)*PH[i]
    y2 = np.arange(0,round(PH[i]+0.1,1),0.1)
    x2 = np.ones(y2.shape)*PE[i]
    plt.plot(x1,y1,linewidth=1.0,color='r',linestyle='--')
    plt.plot(x2,y2,linewidth=1.0,color='r',linestyle='--')
    plt.plot(PES,PHS[i],linewidth=1.5,color='k')
    plt.legend(['$y=P(H)$','$x=P(E)$','$Bayes Function$'])
plt.savefig('C:/Users/Lenovo/Desktop/XJTU-latex-report-master/XJTU-latex-report-master/2.png')
plt.show()
```


$P(H)=0.5, P(E)=0.4, P(H|E)=0.1, P(H|\neg E)=0.6$
$P(H)=0.5, P(E)=0.4, P(H|E)=0.6, P(H|\neg E)=0.0$
$P(H)=0.8, P(E)=0.1, P(H|E)=0.3, P(H|\neg E)=1.0$
$P(H)=0.8, P(E)=0.7, P(H|E)=1.0, P(H|\neg E)=0.7$
$y=P(H)$
$x=P(E)$
$BayesFunction$
$SpecialPoint$
$P(H|S)$
$P(E|S)$

In [ ]:

参考文献

[1] R. O. Duda, P. E. Hart, and N. J. Nilsson, “Subjective bayesian methods for rule-based inference systems,” in *Proceedings of the June 7-10, 1976, national computer conference and exposition*, 1976, pp. 1075–1082.