

主观贝叶斯推理网络实验报告

班级:	软工 2203
学号:	221310332
姓名:	周立成
老师:	肖黎丽

问题重述:

已知下列规则:

R_1 :	IF	E_1	THEN	$(2, 0.000001)$	H_1
R_2 :	IF	E_2	THEN	$(100, 0.000001)$	H_1
R_3 :	IF	H_1	THEN	$(65, 0.01)$	H_2
R_4 :	IF	E_3	THEN	$(300, 0.0001)$	H_2

且先验几率 $O(H_1)=0.1, O(H_2)=0.01$, 通过用户得到 $C(E_1|S_1)=3, C(E_2|S_2)=1$, $C(E_3|S_3)=-2$. 求出 $O(H_2|S_1, S_2, S_3)$

要求: 通过题目的样例, 且保持语句主体不变的情况下, 改变 LS_i, LN_i , 以及 $C(E_i|S_i)$ 也能得到正确的结果。

分析和设计:

1. 在储存规则上, 使用结构体 **Rules**, 来储存 4 条规则, 其中包括 LS, LN , 以及证据和结论的字符串。
2. 再储存额外的信息时, 使用 `vector<long double> OH`, 以及 `vector<long double> C_ES` 来分别保存先验几率 $O(H_i)$ 和用户观察 $C(E_i|S_i)$
3. 同时应该支持以下贝叶斯运算(尽管我们只用了一部分)

a) 根据 LS, LN 求出 $\frac{P(E|H)}{P(E|\neg H)}$:
$$\begin{cases} P(E|H) = LS \times \frac{LN-1}{LN-LS} \\ P(E|\neg H) = \frac{LN-1}{LN-LS} \end{cases}$$

b) 从 $O(X)$ 得到 $P(X)$:
$$P(X) = \frac{O(X)}{1 + O(X)}$$

c) 从 $P(X)$ 得到 $O(X)$:
$$O(X) = \frac{P(X)}{1 - P(X)}$$

d) 从贝叶斯基本公式得到 $P(E)$:
$$P(E) = \frac{P(E|H) \times P(H)}{P(H|E)}$$

e) 从更新的贝叶斯公式得到 $O(H|E), O(H|\neg E)$:
$$\begin{cases} O(H|E) = LS \times O(H) \\ O(H|\neg E) = LN \times O(H) \end{cases}$$

f) 得到 $P(E|S)$:
$$P(E|S) = \begin{cases} \frac{C(E|S) + P(E) \times (5 - C(E|S))}{5}, & 0 \leq C(E|S) \leq 5 \\ \frac{P(E) \times (C(E|S) + 5)}{5}, & -5 \leq C(E|S) \leq 0 \end{cases}$$

g) 得到 $C(E|S)$:
$$C(E|S) = \begin{cases} 5 \times \frac{P(E|S) - P(E)}{1 - P(E)}, & P(E) \leq P(E|S) \leq 1 \\ 5 \times \frac{P(E|S) - P(E)}{P(E)}, & 0 \leq P(E|S) \leq P(E) \end{cases}$$

h) EH 公式:

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

i) CP 公式:

$$P(H|S)=\begin{cases} P(H|\neg E)+[P(H)-P(H|\neg E)]\times\left[\frac{1}{5}C(E|S)+1\right], C(E|S)\leq 0 \\ P(H)+[P(H|E)-P(H)]\times\frac{1}{5}C(E|S), C(E|S)>0 \end{cases}$$

j) 所有观察下 H 的后验概率:

$$O(H|S_1S_2,\dots S_n)=\frac{O(H|S_1)}{O(H)}\times\frac{O(H|S_2)}{O(H)}\times\dots\times\frac{O(H|S_n)}{O(H)}\times O(H)$$

4. 根据 IF-ELSE 条件判断——实现以上的函数即可

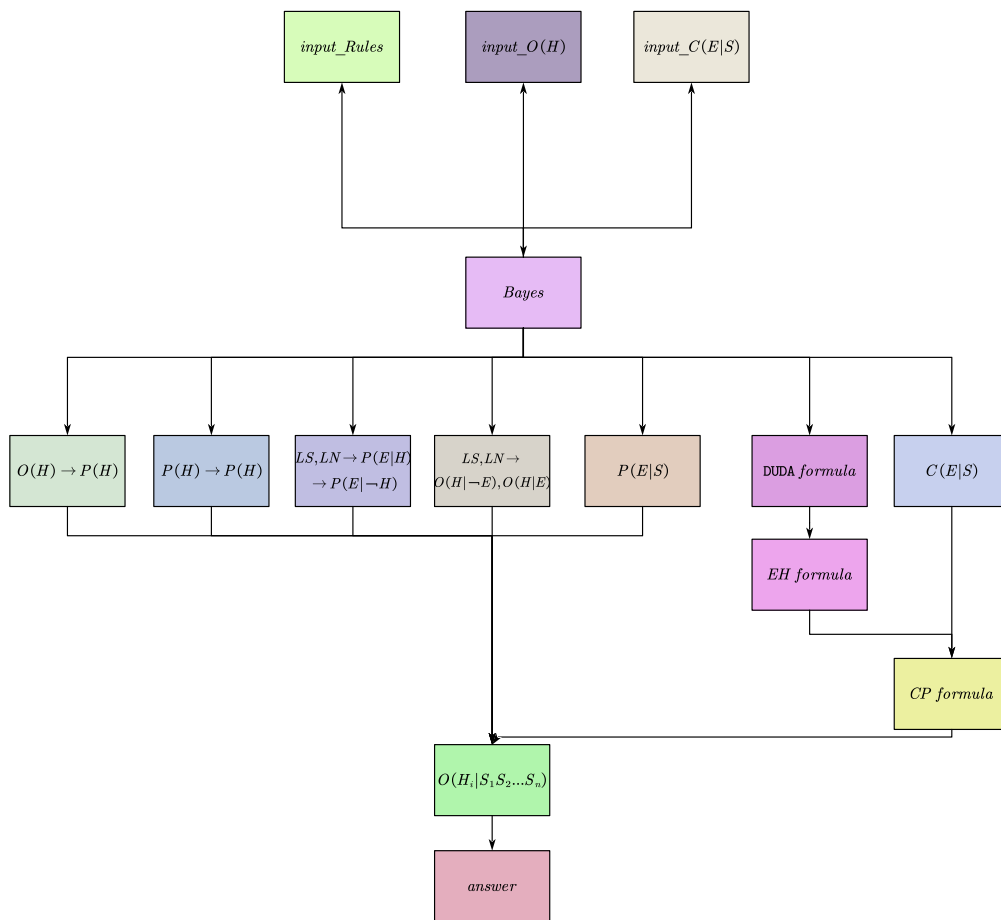
5. 运算部分根据书上例题部分计算,即:

$$O(H_1|S_1)\rightarrow O(H_1|S_2)\rightarrow O(H_1|S_1S_2)\rightarrow O(H_2|S_1S_2)\rightarrow O(H_2|S_3)\rightarrow O(H_2|S_1S_2S_3)$$

6. 各函数头以及部分如下:

```
1.  pair<long double, long double> Get_P_E_H_and_P_E_not_H(long double LS, long
double LN)
2.  long double get_Not(long double P)
3.  long double From_P_get_O(long double P)
4.  long double get_P_E_from_Bayes_basic_formula(long double P_EH, long double P_H,
long double P_HE)
5.  long double get_P_HE(long double P_EH, long double P_H, long double P_E)
6.  long double get_P_NotH_E(long double P_E_NotH, long double P_NotH, long double
P_E)
7.  pair<long double, long double> Updated_Bayes_O_HE_O_H_Not_E(pair<long double,
long double> LS_LN, long double O_H)
8.  long double get_P_ES(long double C_ES, long double P_E)
9.  long double get_C_ES(long double P_ES, long double P_E)
10. long double get_P_HS_from_Duda_Original(long double P_HE, long double P_ES, long
double P_H_NotE, long double P_NotE_S)
11. long double get_P_HS_from_EH_Linear_Interpolation(long double P_H, long double
P_E, long double P_ES, long double P_HE, long double P_H_not_E)
12. long double get_P_HS_From_CP_formula(long double P_H, long double P_HE, long
double P_H_NotE, long double C_ES)
13. long double Get_Posterior_Probability_From_All_Observation (vector<long double>
O_HS, long double O_H)
14. long double process_O_H_S(int index_H, int index_RULE, int index_C)//采用1,2,3,4
的下标输入
```

7. 整体的结构图如下:



实验测试:

实验环境:

- CPU: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz
- 内存:16GB
- 操作系统:WINDOWS 11 22H2
- 编译工具:MVSC 17.0
- 编程语言:C++
- 编程环境:CLION 2023.2

测试:

输出结果:

D: \Bayes.exe

Input LS1 and LN1 for E1 and H1

2 0.000001

```

IF E1 THEN (2.000000,0.000001) H1
Input LS2 and LN2 for E2 and H1
100 0.000001
IF E2 THEN (100.000000,0.000001) H1
Input LS3 and LN3 for H1 and H2
65 0.01
IF H1 THEN (65.000000,0.010000) H2
Input LS4 and LN4 for E3 and H2
300 0.0001
IF E3 THEN (300.000000,0.000100) H2
Now input O(H1)
0.1
Now input O(H2)
0.01
Now input C(E1|S1)
3
Now input C(E2|S2)
1
Now input C(E3|S3)
-2

```

```

P_H1=0.0909091
O_H1_S1=0.157895
P_H1=0.0909091
O_H1_S2=0.341463
O_H1_S1S2=0.539153
P_H2=0.00990099
P_H1_S1S2=0.350292
O_H2_H1 =0.65
P_H2_H1=0.393939
P_H2_S1S2=0.119475
O_H2_S1S2=0.135686
P_H1=0.00990099
O_H2_S3=0.0059765

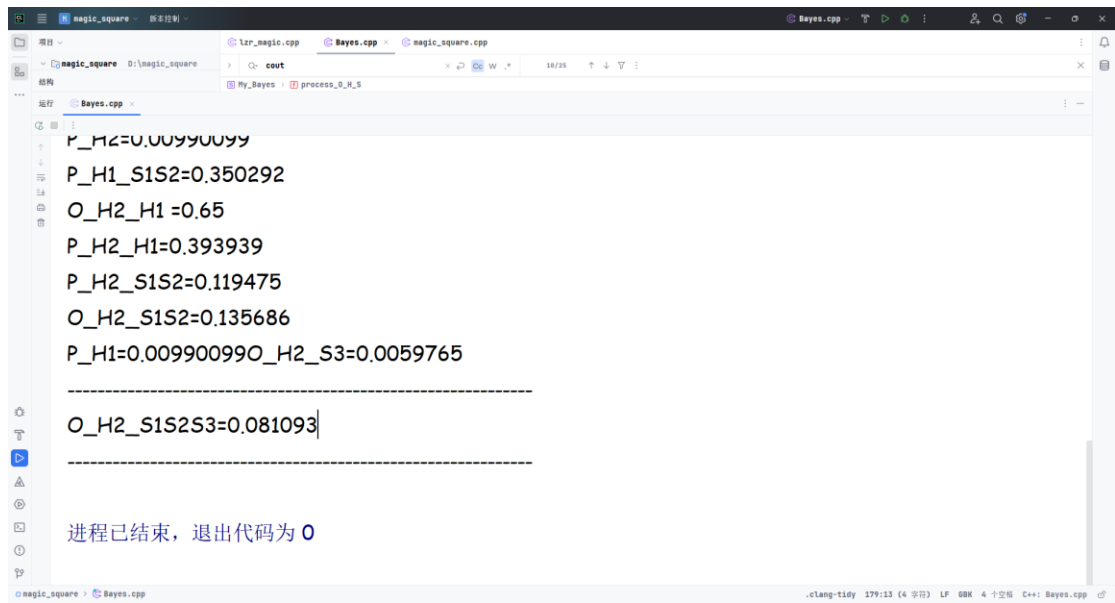
```

```

O_H2_S1S2S3=0.081093

```

进程已结束，退出代码为 0
与书上结果一致,运行正确



```
P_H2=0.00990099
P_H1_S1S2=0.350292
O_H2_H1=0.65
P_H2_H1=0.393939
P_H2_S1S2=0.119475
O_H2_S1S2=0.135686
P_H1=0.00990099O_H2_S3=0.0059765

-----
O_H2_S1S2S3=0.081093|

-----

进程已结束，退出代码为 0
```

改进和不足：

- 希望能自己通过 Rules 生成推理网络自己推理