华东师范大学软件学院课程作业

课程名称：软件质量分析	年级：2023 级本科	姓名：张梓卫
作业主题：软件老化分析模型	学号：10235101526	作业日期：2024/12/24
指导老师：陈仪香	组号：	

目录

一 软件老化的模型实现

1 验证 Python 程序使用

查看示例中的表格，可以发现，在每一个类别中，括号标注了当前类别的度量元的数量：

	MEM（15） 内存管理	STO（4） 存储空间	LOG（7） 逻辑资源	NUM（2） 误差累积	ARU（2） 未知原因	行总计	（权重：总计所占百分比 θ_i）
MEM（15）		15/4	15/7	15/2	15/2	585/28= 20.893	0.539
STO（4）	4/15		4/7	2	2	508/105= 4.838	0.125
LOG（7）	7/15	7/4		7/2	7/2	553/60= 9.217	0.238
NUM（2）	2/15	1/2	2/7		1	403/210= 1.919	0.049
ARU（2）	2/15	1/2	2/7	1		403/210= 1.919	0.049
列总计	1	13/2	23/7	14	14	543/14 38.786	1.000

图 1: Python 代码

右侧的权重即为当前类别的权重。

在例子当中，t = 0 时，该时刻所有不同的度量元最大的风险值均为 1。

故我们按照给出的 Python 代码可以得出以下的输入序列：

```
类别数：5
各个类别的权重：0.539 0.125 0.238 0.049 0.049
各个类别的度量元数量：15 4 7 2 2
第1个类别-各个度量元的权重：0.0506 0.1845 0.0238 0.0238 0.0774 0.0774 0.0774 0.0238 0.0238 0.0506 0.0506 0.0238 0.0238 0.1042 0.1845
第1个类别-各个度量元的该时刻最大风险值：1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
第2个类别-各个度量元的权重：0.25 0.25 0.25 0.25
第2个类别-各个度量元的该时刻最大风险值：1 1 1 1
第3个类别-各个度量元的权重：0.2340 0.1064 0.1064 0.2340 0.1064 0.1064 0.1064
第3个类别-各个度量元的该时刻最大风险值：1 1 1 1 1 1 1
第4个类别-各个度量元的权重：0.2 0.8
第4个类别-各个度量元的该时刻最大风险值：1 1
第5个类别-各个度量元的权重：0.1 0.9
第5个类别-各个度量元的该时刻最大风险值：1 1
各类别的熵：[0.0, 0.0, 0.0, 0.0, 0.0]
各类别的可信值：[10.0, 10.0, 10.0, 10.0, 10.0]
可信值：10.0
```

图 2: 输入序列

此时得到的可信值为 10，与 PPT 中的结果一致。我们可以更换至 $t = 10$ 时重新输入一个序列。
注意此时在每一个度量元处，需要我们判别哪一个才是大哥前时刻最大的风险值。

```
类别数：5
各个类别的权重：0.539 0.125 0.238 0.049 0.049
各个类别的度量元数量：15 4 7 2 2
第1个类别-各个度量元的权重：0.0506 0.1845 0.0238 0.0238 0.0774 0.0774 0.0774 0.0238 0.0238 0.0506 0.0506 0.0238 0.0238 0.1042 0.1845
第1个类别-各个度量元的该时刻最大风险值：7 7 9 7 7 4 7 9 7 1 7 9 1 4 7
第2个类别-各个度量元的权重：0.25 0.25 0.25 0.25
第2个类别-各个度量元的该时刻最大风险值：7 7 7 9
第3个类别-各个度量元的权重：0.2340 0.1064 0.1064 0.2340 0.1064 0.1064 0.1064
第3个类别-各个度量元的该时刻最大风险值：7 10 7 9 4 10 4
第4个类别-各个度量元的权重：0.2 0.8
第4个类别-各个度量元的该时刻最大风险值：9 7
第5个类别-各个度量元的权重：0.1 0.9
第5个类别-各个度量元的该时刻最大风险值：7 9
各类别的熵：[0.7458799513127149, 0.8723841573705238, 0.8518824861842456, 0.8669269338992704, 0.9433280624968181]
各类别的可信值：[4.743167406123722, 4.1795389258142865, 4.266110860961348, 4.20240995313211, 3.893299624085329]
可信值：4.481945667961918
```

图 3: $t = 10$ 时的输入序列

此时得到的结果完全符合 PPT 上的内容：

时间	软件可信度
t=0	10
t=10	4.482

图 4: PPT 上的示例输出

二 使用作业中的数据

由于此处的度量元已经固定下来了，因此我暂不编写和度量元权重有关的程序。
我们可以将此次作业需要用到的 Python 程序优化以下，使其更加符合规范，增强可读性：

```python
import math
def input_data():
    n = int(input("类别数："))
    theta = list(map(float, input("各个类别的权重（用空格分隔）：").split()))
    m = list(map(int, input("各个类别的度量元数量（用空格分隔）：").split()))

    R = []  # 各类别度量元的最大风险值
    BETA = []  # 各类别度量元的权重
    for i in range(n):
```

```
        print(f"输入第 {i + 1} 个类别的数据：")
        beta = list(map(float, input(f"  各度量元的权重（用空格分隔）：").split()))
        r = list(map(float, input(f"  各度量元的最大风险值（用空格分隔）：").split()))
        BETA.append(beta)
        R.append(r)

    return n, theta, m, R, BETA

def calculate_entropy_and_trust(n, m, R, BETA):
    Hs = []  # 熵
    Us = []  # 可信值
    for i in range(n):
        # 计算熵 H
        H = sum(BETA[i][j] * math.log10(R[i][j]) for j in range(m[i]))

        # 计算可信值 U
        U = max(10 * math.exp(-H), 1)

        Hs.append(H)
        Us.append(U)

    return Hs, Us

def calculate_total_trust(n, theta, Us):
    T = 1
    for i in range(n):
        T *= math.pow(Us[i], theta[i])
    return T

def main():
    n, theta, m, R, BETA = input_data()

    # 计算熵和可信值
    Hs, Us = calculate_entropy_and_trust(n, m, R, BETA)

    print("各类别的熵：", Hs)
    print("各类别的可信值：", Us)

    T = calculate_total_trust(n, theta, Us)
    print("总可信值：", T)

if __name__ == "__main__":
    main()
```

```
类别数：5
各个类别的权重（用空格分隔）：0.539 0.125 0.238 0.049 0.049
各个类别的度量元数量（用空格分隔）：15 4 7 2 2
输入第 1 个类别的数据：
  各度量元的权重（用空格分隔）：0.0506 0.1845 0.0238 0.0238 0.0774 0.0774 0.0774 0.0238 0.0238 0.0506 0.0506 0.0238 0.0238 0.1042 0.1845
  各度量元的最大风险值（用空格分隔）：4 4 1 4 1 1 4 1 1 1 4 1 4 1 4
输入第 2 个类别的数据：0.25 0.25 0.25 0.25
  各度量元的最大风险值（用空格分隔）：4 1 4 4
输入第 3 个类别的数据：
  各度量元的权重（用空格分隔）：0.2340 0.1064 0.1064 0.2340 0.1064 0.1064 0.1064
  各度量元的最大风险值（用空格分隔）：1 1 4 1 1 1 7
输入第 4 个类别的数据：
  各度量元的权重（用空格分隔）：0.2 0.8
  各度量元的最大风险值（用空格分隔）：4 1
输入第 5 个类别的数据：
  各度量元的权重（用空格分隔）：0.1 0.9
  各度量元的最大风险值（用空格分隔）：1 4
各类别的熵： [0.3583461068384032, 0.45154499349597177, 0.15397761453481212, 0.12041199826559248, 0.5418539921951662]
各类别的可信值： [6.988311629008543, 6.366437808936636, 8.572912116592352, 8.865551022992525, 5.816688425801949]
总可信值： 7.271013653215722
```

图 5: $t = 0$ 时的数据

Python 运行程序如上所示，整理成表格如下所示：

	时间	MEM	STO	LOG	NUM	ARU
熵	t=0	0.3583	0.4515	0.1540	0.1204	0.5419
	t=10	-	-	-	-	-
可信度	t=0	6.9883	6.3664	8.5729	8.8656	5.8167
	t=10	-	-	-	-	-

表 1: 熵和可信度随时间的变化

接下来填写入 $t = 10$ 时的数据，注意也要把此时刻的风险最大值给填了：

时间	软件可信度
t=0	7.2710
t=10	-

表 2: 软件可信度随时间的变化

```
类别数：5
各个类别的权重（用空格分隔）：0.539 0.125 0.238 0.049 0.049
各个类别的度量元数量（用空格分隔）：15 4 7 2 2
输入第 1 个类别的数据：
  各度量元的权重（用空格分隔）：0.0506 0.1845 0.0238 0.0238 0.0774 0.0774 0.0774 0.0238 0.0238 0.0506 0.0506 0.0238 0.0238 0.1042 0.1845
  各度量元的最大风险值（用空格分隔）：4 7 7 4 9 4 7 7 7 4 7 7 7 9 7
输入第 2 个类别的数据：
  各度量元的权重（用空格分隔）：0.25 0.25 0.25 0.25
  各度量元的最大风险值（用空格分隔）：7 9 7 7
输入第 3 个类别的数据：
  各度量元的权重（用空格分隔）：0.2340 0.1064 0.1064 0.2340 0.1064 0.1064 0.1064
  各度量元的最大风险值（用空格分隔）：7 10 7 4 7 10 9
输入第 4 个类别的数据：
  各度量元的权重（用空格分隔）：0.2 0.8
  各度量元的最大风险值（用空格分隔）：7 4
输入第 5 个类别的数据：
  各度量元的权重（用空格分隔）：0.1 0.9
  各度量元的最大风险值（用空格分隔）：7 7
各类别的熵： [0.8157277746077432, 0.8723841573705238, 0.8328032452534573, 0.6506676010652213, 0.8450980400142568]
各类别的可信值： [4.4231730289333795, 4.1795389258142865, 4.348286449073418, 5.216973747540717, 4.295152464655678]
总可信值： 4.403325777923536
```

图 6: $t = 10$ 时的数据

	时间	MEM	STO	LOG	NUM	ARU
熵	t=0	0.3583	0.4515	0.1540	0.1204	0.5419
	t=10	0.8157	0.8724	0.8328	0.6507	0.8451
可信度	t=0	6.9883	6.3664	8.5729	8.8656	5.8167
	t=10	4.4232	4.1795	4.3483	5.2170	4.2952

表 3: 熵和可信度随时间的变化

时间	软件可信度
t=0	7.2710
t=10	4.4033

表 4: 软件可信度随时间的变化