

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：可信评估体系及应用	学号：10235101526	作业日期：2024/11/13
指导老师：陈仪香	组号：	

目录

一 作业	1	3	软件可信度量值计算	3
1 度量元量化	1	4	分级计算	4
2 子属性分组	2	5	总结	5

一 作业

软件编号	软件类型	1. 总体策划与执行情况				2. 分析与设计情况				3. 测试验证情况					4. 可靠性和安全性情况				5. 软件技术状态更改情况				6. 质量问题归零及举一反三情况		7. 配置管理情况				8. 软件开发环境情况		9. 第三方评测情况	
		开发计划与执行情况	功能性能与任务符合情况	软件开发文档完整性	输入文件控制	需求分析情况	软件设计情况	测试计划与执行情况	代码审查与静态分析情况	测试内容全面性	专项测试情况	测试覆盖率情况	测试环境、方法情况	可靠性全分析情况	可靠性全设计情况	可靠性全验证情况	技术状态更改依据、论证及审批情况	更改测试、验证情况	更改落实情况	质量问题归零工作完成情况	举一反三工作落实情况	配置管理组织、要求与工具情况	变更控制情况	配置审计、配套软件情况	开发环境、配套硬件	开发环境、配套软件	评测输入和计划	评测执行情况	问题解决情况			
10	III	C	B	C	D	B	B	C	A	C	D	C	A	C	C	C	C	C	A	D	C	A	A	A	B	B	C	C	D			
11	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A			
12	II	C	B	B	B	D	D	A	C	C	A	C	B	B	B	A	B	A	C	A	A	A	A	B	B	A	A	A				
13	I	C	B	B	B	D	D	A	C	C	C	B	C	B	C	A	B	A	A	C	A	A	A	B	B	A	A	A				
14	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A				
15	II	C	B	B	B	D	D	A	B	C	C	B	C	B	B	B	B	A	A	C	A	A	A	B	B	A	A	A				
16	I	C	B	B	B	D	D	B	C	C	C	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A				
17	I	C	R	B	R	D	D	B	C	C	C	R	C	R	C	C	A	B	A	A	C	A	A	A	R	B	A	A	A			

1 度量元量化

首先按照度量元的等级评价来进行数据量化, 即:

- 若度量元等级评价为 A, 则该度量元值 $index = 1$; 类似地, 若为 B, 则 $index = 0.9$;
- 若为 C, 则 $index = 0.7$; 若为 D, 则 $index = 0.2$.

10 - 17 的量化后表格如下所示:

```

1 scores = np.array([
2     [7, 9, 7, 2, 9, 9, 7, 10, 7, 2, 7, 10, 7, 7, 7, 7, 7, 10, 2, 7, 10, 10, 10, 9, 9, 7, 7, 2],
3     [7, 9, 9, 9, 2, 2, 9, 7, 7, 7, 9, 7, 9, 7, 7, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
4     [7, 9, 9, 9, 2, 2, 10, 10, 7, 7, 10, 7, 9, 9, 9, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
5     [7, 9, 9, 9, 2, 2, 9, 7, 7, 7, 9, 7, 9, 7, 7, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
6     [7, 9, 9, 9, 2, 2, 9, 7, 7, 7, 9, 7, 9, 7, 7, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
7     [7, 9, 9, 9, 2, 2, 10, 9, 7, 7, 9, 7, 9, 9, 9, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
8     [7, 9, 9, 9, 2, 2, 9, 7, 7, 7, 9, 7, 9, 7, 7, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 10],
9     [7, 9, 9, 9, 2, 2, 9, 7, 7, 7, 9, 7, 9, 7, 7, 10, 9, 10, 10, 7, 10, 10, 10, 9, 9, 10, 10, 9]
10 ])

```

度量元等级评价量化

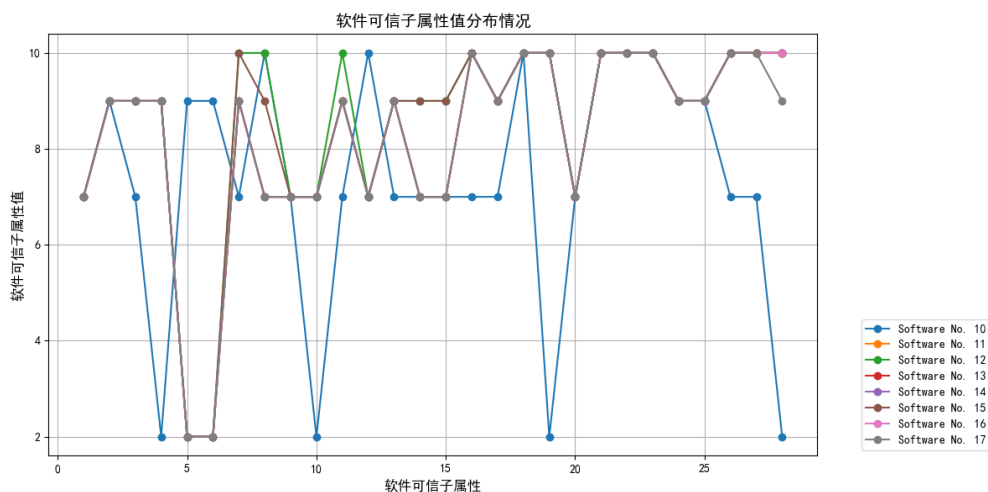


图 1: 度量元等级评价分布情况折线图

2 子属性分组

根据示例，每一个子属性下应该进行幂积运算，我们将子属性进行分组：

```

# 子属性权重
sub_attribute_weights = np.array([
    0.31, 0.36, 0.33, # 总体策划与执行情况
    0.33, 0.33, 0.34, # 分析与设计情况
    0.16, 0.17, 0.17, 0.17, 0.17, 0.16, # 测试验证情况
    0.33, 0.34, 0.33, # 可靠性和安全性情况
    0.34, 0.33, 0.33, # 软件技术状态更改情况
    0.50, 0.50, # 质量问题归零及举一反三情况
    0.33, 0.34, 0.33, # 配置管理情况
    0.50, 0.50, # 软件开发环境情况
    0.33, 0.33, 0.34 # 第三方评测情况
])

lengths = [3, 3, 6, 3, 3, 2, 3, 2, 3]
results = []

```

子属性分组

下面，进行子属性的权重幂积运算：

```

1  for row in scores:
2      idx = 0
3      row_results = []
4      # 遍历每个组
5      for length in lengths:
6          x_i = row[idx:idx+length]
7          w_i = sub_attribute_weights[idx:idx+length]
8          G = np.prod(x_i ** w_i)
9          row_results.append(G)
10         idx += length # 移动到下一个组
11     results.append(row_results)
12
13 results = np.array(results)

```

子属性权重幂积运算

运算结果如下所示:

```
Table of Results:
[[ 7.66284514  5.47876131  6.36391132  7.
    7.87436763  3.74165739
 10.
    9.
    4.57209141]
 [ 8.32545013  3.28541416  7.60528888  7.60528888  9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  8.36660027  9.
    9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  7.60528888  7.60528888  9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  7.60528888  7.60528888  9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  8.07219181  9.
    9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  7.60528888  7.60528888  9.65828526  8.36660027
 10.
    9.
    10.
    ]
 [ 8.32545013  3.28541416  7.60528888  7.60528888  9.65828526  8.36660027
 10.
    9.
    9.6481146 ]]

Process finished with exit code 0
```

Row Index	1	2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
10	7.6628	5.4788	6.3639	7.0000	7.8744	3.7417	10.0000	9.0000	4.5721
11	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000
12	8.3255	3.2854	8.3666	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000
13	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000
14	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000
15	8.3255	3.2854	8.0722	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000
16	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000
17	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	9.6481

表 1: 结果表

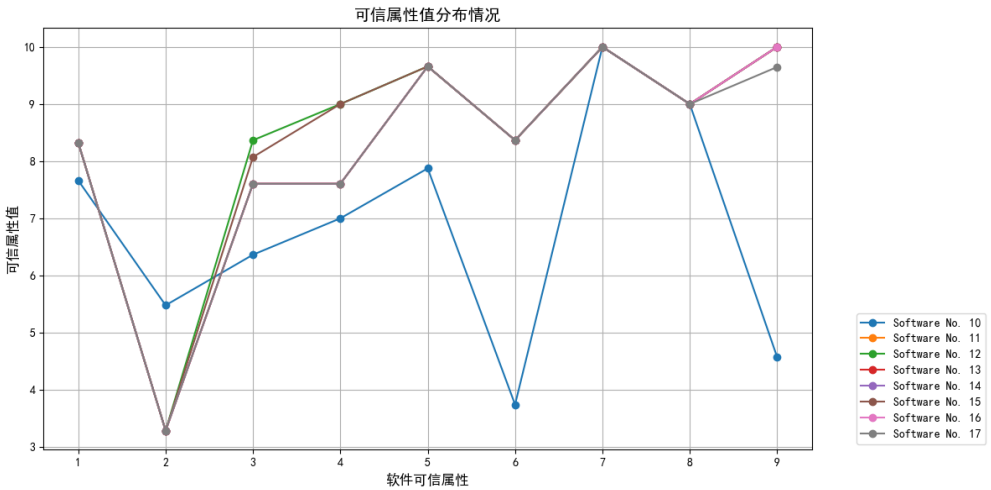


图 2: 可信属性值分布情况折线图

3 软件可信度量值计算

获得了输出的数据以后，接下来我们要对新的数据进行新的加权：

```
1 attribute_weights = np.array([0.05, 0.17, 0.20, 0.15, 0.09, 0.09, 0.11, 0.05, 0.09])
2
3 final_results = []
4 for row in results:
5     # 幂积公式: x1^w1 * x2^w2 * ... * xn^wn
6     final_value = np.prod(row ** attribute_weights)
7     final_results.append(final_value)
8
```

```
final_results = np.array(final_results)
```

属性计算

```
最终计算结果:
[6.40711733  7.2713282  7.60097716  7.2713282  7.2713282  7.54671446
 7.2713282  7.24792296]
```

结果表格如下:

Row Index	Final Result
10	6.4071
11	7.2713
12	7.6010
13	7.2713
14	7.2713
15	7.5467
16	7.2713
17	7.2479

表 2: 最终计算结果表

4 分级计算

Table: 软件可信性分级模型

软件可信度量值要求（黄金分割）	可信属性要求（多数原则）	可信等级
$9.5 \leq T$	1. 低于9.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$ 个 2. 没有低于8.5分的可信属性	V
$8.5 \leq T < 9.5$ 或者 $T > 9.5$ 且不能评为V级别	1. 低于8.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$ 个 2. 没有低于7.0分的可信属性	IV
$7.0 \leq T < 8.5$ 或者 $T > 8.5$ 且不能评为IV级别及以上者	1. 低于7.0分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$ 个 2. 没有低于4.5分的可信属性	III
$4.5 \leq T < 7.0$ 或者 $T > 7.0$ 且不能评为III级别及以上者	1. 低于4.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$ 个	II
$T < 4.5$ 或者 $T > 4.5$ 且不能评为II级别及以上者	1. 无要求	I

图 3: 软件可信度等级判定方法

根据上述标准，我们可以写出相关的代码：

```
def classify_trust_levels(final_results, attribute_scores):
    trust_levels = []
    for T, scores in zip(final_results, attribute_scores):
        n = len(scores)
        print(f"n的个数为{n}")
        threshold_2_3 = n - int(np.ceil(2 * n / 3)) # 使用 ceil 函数向上取整
        print(f"不能超过{threshold_2_3}个")

        # 统计低于某些阈值的属性个数
        low_9_5 = np.sum(scores < 9.5)
        low_8_5 = np.sum(scores < 8.5)
        low_7_0 = np.sum(scores < 7.0)
        low_4_5 = np.sum(scores < 4.5)

        if T >= 9.5 and low_9_5 <= threshold_2_3 and low_8_5 == 0:
```

```
16         trust_levels.append("V")
17         continue
18
19     if (8.5 <= T < 9.5 and low_8_5 <= threshold_2_3 and low_7_0 == 0) or (T > 9.5 and "V" not in
trust_levels):
20         trust_levels.append("IV")
21         continue
22
23     if (7.0 <= T < 8.5 and low_7_0 <= threshold_2_3 and low_4_5 == 0) or (T > 8.5 and "IV" not in
trust_levels):
24         trust_levels.append("III")
25         continue
26
27     if (4.5 <= T < 7.0 and low_4_5 <= threshold_2_3) or (T > 7.0 and "III" not in trust_levels):
28         trust_levels.append("II")
29         continue
30
31     if T < 4.5 or (T > 4.5 and "II" not in trust_levels):
32         trust_levels.append("I")
33         continue
34
35     return trust_levels
36
37 # 计算可信等级
38 trust_levels = classify_trust_levels(final_results, results)
39 print(trust_levels)
```

分级计算

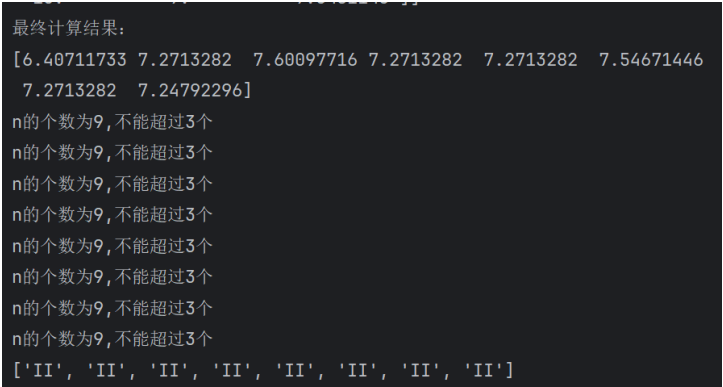


图 4: 软件可信度等级判定结果

5 总结

最终的计算结果表格应如下所示：

Index	1	2	3	4	5	6	7	8	9	Value	Level
10	7.6628	5.4788	6.3639	7.0000	7.8744	3.7417	10.0000	9.0000	4.5721	6.4071	II
11	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
12	8.3255	3.2854	8.3666	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000	7.6010	II
13	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
14	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
15	8.3255	3.2854	8.0722	9.0000	9.6583	8.3666	10.0000	9.0000	10.0000	7.5467	II
16	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	10.0000	7.2713	II
17	8.3255	3.2854	7.6053	7.6053	9.6583	8.3666	10.0000	9.0000	9.6481	7.2479	II

表 3: 结果表