CS24011：实验室3：一台基于模糊逻辑的洗衣机

里萨巴蒂斯塔纳瓦罗
弗朗西斯科洛博

学术课程： 2022-23

Git标签：24011-lab3-S-模糊逻辑

准备工作

在这个练习中，我们将使用Python 3并在frules库进行构建。作为第一步，通过发出以下命令，在您的环境中安装该库

```
$ pip安装规则
```

你可以在https://github上找到更多关于这个图书馆的信息。com/swistakm/frules.更具体地说，我们将使用它的一些辅助功能来完成以下操作。

A定义模糊集的隶属度函数

例如，图书馆允许我们根据生产年份方便地定义汽车年龄的会员功能，如下所示

```
从分子。从分子表达式。表情 进口进口 表达 ltrapezoid ，梯形，梯形 作为E
#汽车年龄表达式

老=E（l梯形（2001,2008），“old“”）
新的=E（梯形（2013,2014），“new“”）
```

下面的图1允许我们可视化上述的成员函数。

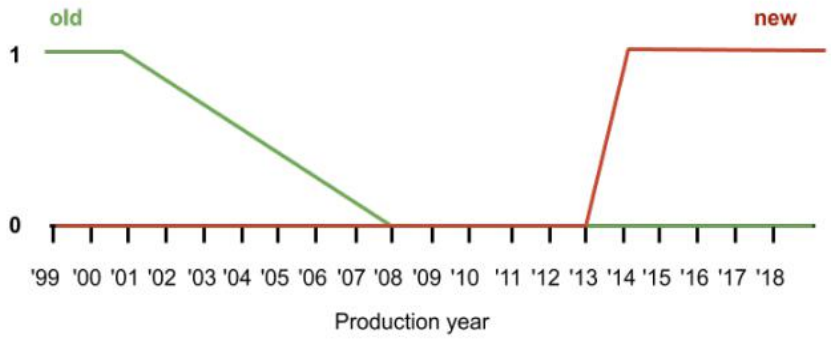

图1：新旧车会员功能可视化

B评估一个清晰的值属于一个模糊集的程度

例如，我们可以计算具有= 2007生产年份的汽车的“多大程度”或“多新程度”，如下所示

```
从分子。规则导入规则作为R
旧套=R（产品为=年）
how _ old = old _ set。版本（新=2007）新集=R（新=）          #结果      在  0
how _ new = new _ set。（关于ar =2007）                                  .14
                                                              #结果      在
```

问题介绍

利用模糊逻辑的家用电器的一个很好的例子就是洗衣机[1].对于这个练习，我们的任务是建立一个简单的基于模糊逻辑的系统，这将帮助确定如何设置机器应该使用的温度，这取决于衣服的肮脏和精致程度。

构建你的模糊逻辑系统

在您的GitLab回购中，您将找到一个名为模糊洗涤机器的Python脚本。它包含一些存根代码。

您将开发您自己的此脚本版本，此后在本文档中称为“您的解决方案”。

例如，在开发期间，您可以运行该命令

```
$蟒蛇的大小，它和_m。py模糊化dirty _ set 0。9
```

它将执行并显示模糊化的结果（dirty_set，0 .9）。同样，你也可以使用

```
$蟒蛇的大小，它和_m。py g e t _ r u l e _ o u t p u t _ v a l u e 3 0 .5
```

它将执行并显示get_rule_lomput_value（3,0.5）的结果。您将从存根代码中获得的输出当然没有意义的。

更一般地说，您可以按如下方式运行该脚本

```
$蟒蛇的大小，它和_m。py函数ARGS。..
```

来测试解决方案中的每个函数。当您完成该练习时，该命令

```
$蟒蛇的大小，它和_m。py \
     当你回来的时候，你的时候。9 ,6 .5)) ″
```

将获得机器的温度设置，例如，污垢量为0。9汤匙，织物的重量是每平方码6.5盎司。

重要提示：请注意，对于下面任何要求您使用指定的函数名编写一个函数的任务，您都将找到相应的函数签名(i。e., 参数，返回类型)在存根代码中。

1https://ao.洗衣机-90月1日-20日

C Initialisation

C.1.语言变量的定义

我们有两个输入变量，肮脏和精致，以及以下语言术语

```
T（脏）={几乎干净，脏}
T（精致）={非常精致，精致，不精致}
```

我们有一个输出变量，温度

```
T（温度）={低、中、高}
```

C.2.模糊集的定义

为了定义我们的模糊集，我们将对每个输入变量使用隶属度函数
下面图2和图3中的梯形形状描绘了肮脏和精致，
各自地

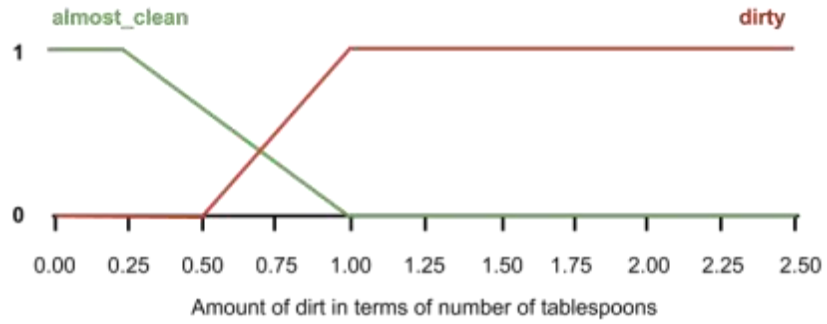

图2：描述污垢的隶属度函数的曲线

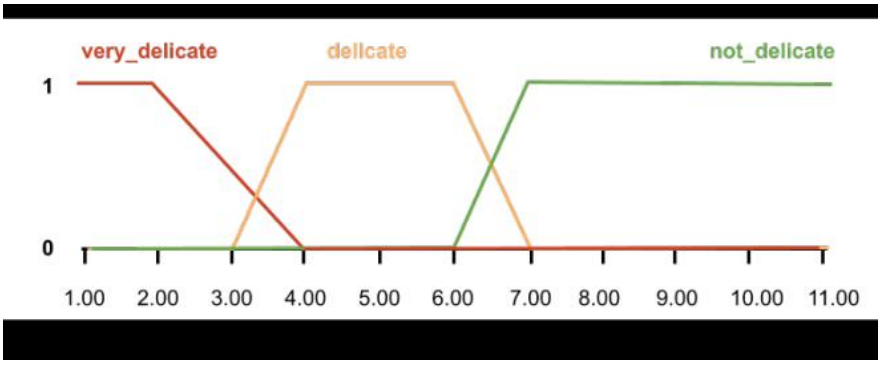

图3：描述精致度的隶属度函数的曲线

基于上述曲线定义成员关系函数的代码行已经作为存根代码的一部分提供了。确保你理解了这些台词。

.3C定义规则

以下是模糊逻辑系统应该遵循的规则。

```
如果衣服很精致，
      然后温度应该                    低点
如果衣服很精致                        或几乎是干净的，
      然后温度应该                    低点
如果衣服很精致                        和肮脏，
      然后温度应该                    中度
如果衣服不精致、不脏，
      那么温度应该很高
```

每个规则都遵循IF<前验>，然后<后>　　模式

D Fuzzification

现在，我们将基于模糊化，对上述规则进行编码。

D.1.模糊化输入的计算

任务1：在你的解决方案中，写一个名为模糊的函数，它将计算模糊的程度
　　一个清晰的输入值(i。e.，污垢的数量，织物的重量)属于一个模糊的集。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py模糊化dirty _ set 0。9调试运行：模糊化(dirty _
set ，0。9)
ret值：0。8
```

.2D前因的评价

如果一个规则包含多个前因，那么它们的组合值可以通过连接或分离来计算。

任务2：在您的解决方案中，编写一个叫做get_连接值的函数来计算连接值
　　（和）在给定规则中的前因子，以及另一个称为get_分离的函数，用于计算它们的分离（OR）
　　。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。我很名字。8　0 .5调试运行： ge t _ co nj unc ti在(0。
8 ,0 .5)
ret值：0。5
$蟒蛇的大小，它和_m。我很我。8　0 .5调试运行： ge t _ di sj unc ti在(0。8
,0 .5)
ret值：0。8
```

任务3：在您的解决方案中，编写一个名为get_rule_先行值的函数来计算
一个规则的前因的组合值（在我们的例子中，它最多可以包含两个前因）。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py g e t _ r u l e _ a n t e c e d e n t _ v a l u e \
    没有0。0  v e r y _ d e l i c a t e _ s e t  6 .5  ″ ’ ’ ″
调试运行：获取_rule_antecedent_value(无，0。0，非常精致的设置，6。5 , ’  ’ )
ret值：0。0

$蟒蛇的大小，它和_m。py g e t _ r u l e _ a n t e c e d e n t _ v a l u e \
    dirty _ set 0。9精致的套装6。5””和‘“
调试运行：get_rule_antee__set，0。9号、精致的套装，6号。5，”和‘)
ret值：0。5
```

E推理

我们将使用苏吉诺式的模糊推理，特别是，零阶的苏吉诺模糊模型，其中每个规则的输出水平都是一个常数k。加权输出水平计算为规则先行词值与对应规则的输出水平的乘积。

输出水平是加权的，因为并非所有的规则都具有同等的重要性。下面是一个表，显示了与上面相同的四个规则，但这次带有相应的输出级别。

规则编号	规则	输出电平
r1	如果衣服很精致， 那么温度应该很低	10
r2	如果衣服很精致或几乎很干净，那么温度应该很低	40
r3	如果衣服又精致又脏，那么温度应该是中等温度的	60
r4	如果衣服不精致、不脏，那么温度应该很高	100

请注意，这些输出级别与成员关系函数一样，都是根据用户的知识/经验任意选择的。

存根代码已经以a的形式提供了上述四个规则的输出级别
字典称为rule_weights_dict，其中键（从1到4的整数）对应于规则编号。

任务4：在您的解决方案中，编写一个名为get_rule_ouput_value的函数，返回该值
根据上面的输出水平，给定一个先行值的规则的加权输出水平。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py g e t _ r u l e _ o u t p u t _ v a l u e 3 0 .5调试运行
: g e t _ r u l e _ o u t p u t _ v a l u e (3 , 0。5)
ret值：30。0
```

任务5：在你的解决方案中，写一个名为配置洗衣机的函数，给定脆度
污垢量和织物重量的输入值，返回一个由2个列表组成的元组，其中分别包含所有规则的先行词值和输出值。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py \
    c o n f i g u r e _ w a s h i n g _m a c h i n e  0 .9  6 .5
调试运行：它有一个和_m(0。9 ,6 .5)
ret值：（[0,0]。5 ,  0 .5 ,  0 .5] ,  [0 ,  20 .0 ,  30 .0 ,  50 .0])
```

F Defuzzification

为了计算一个清晰的输出值（对于温度），我们将对所有规则输出使用加权平均。

任务6：在你的解决方案中，写一个名为get_加权平均的函数来计算
对所有规则的加权平均值。加权平均值是所有规则输出的和除以所有规则先行值的和。

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py \
    g e t _ w e i g h t e d _ a v e r a g e  ″[0 ,0 .5 ,0 .5 ,0 .5]″  ″[0 ,20 .0
,30 .0 ,50 .0]“调试运行： g e t _ w e”([0 , 0。5 ,0 .5 ,0 .5] ,[0 ,20 .0 ,30 .0 ,50
.0])
ret值：6 6 。6 6 6 6 6 6 6 6 6 6 6 6 7
```

任务7：在你的解决方案中，写一个名为get_temor的函数来计算实际的
洗衣机应设置为的温度值。
您可以将上一个任务的脆输出值视为输出变量可能值范围的百分比，在这种情况下是温度。假设洗衣机的可能的温度范围在10° C和90° C之间（如下蓝色所示）。这意味着0%的脆输出值为10° C和一个脆输出
100%的值为90° C。

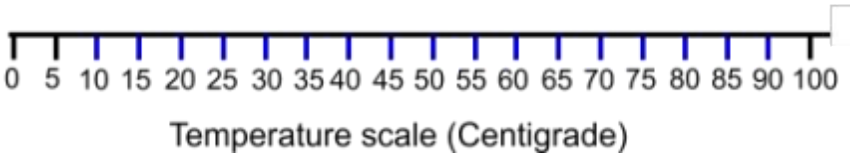

图4：温度范围

您可以在命令行上验证您的函数是否正确。例如，您应该获得以下输出

```
$蟒蛇的大小，它和_m。py \
      当你回来的时候，你的时候。9 ,6 .5))“调试运行：get__m(*(igure_wash在g _m和_m(0
。9 ,6 .5)))
ret值：6 3 。3 3 3 3 3 3 3 3 3 3 3 3 3 4
```

提交

请遵循自述文件。在你的COMP24011_2022 Gitlab repo中刷新你的lab3分支的文件。您将发现一个包含存根/骨架代码的文件，名为模糊洗衣机。在其中为您的解决方案编写Python代码。当你是
准备好推动您的解决方案，记住你需要用命令标记您的提交

```
$ git标签24011-实验室3-S-模糊逻辑
```

必须只有一个叫做模糊洗衣机的文件。在你的承诺中扮演角色。此外，您所包含的任何其他文件都将被忽略。使用Gitlab提交课程提交的一般说明
可以在CS手册上找到。

提交的截止日期是11月28日星期一的18：00。实验室将自动离线。标记程序将从Gitlab下载您的代码，并根据参考实现进行测试。(这将需要一些时间，所以预计标记至少需要2周的时间。**重要的是：骨架模糊的洗衣机。**py已经包含了必要的导入语句和模糊集的定义。您不应该在解决方案中添加任何顶级代码，而只应该完成所需的函数定义。

骨架脚本使用if__名称__==‘__main__’：构造，使源文件
用于调试目的，以及像Python模块一样导入。如果您希望使用其他调试代码，则需要在此调试构造中保护它。其原因是，自动标记程序将通过导入以下代码来测试您的解决方案

```
从_m开始，输入模糊，连接，分离，规则前值，规则输出值，配置洗衣机，加权平均，
      加权平均
```

任何没有被保护的顶级代码都会影响对函数的测试。

标记方案

共计20个标记，其分布范围如下表所示。

任务	功能	标记
1	fuzzify()	2
2a	获得连接（）	2
2b	get_分离（）	2
3	get_rule_entent_value（）	3
4	get_rule_outh_value（）	2
5	配置洗衣机（）	3
6	get_加权平均（）	3
7	get_温度（）	3

每个函数都将在各种测试用例上进行测试，并返回与参考实现相比较的值。完全正确的返回值的比例将决定您的分数（超出表中所示的函数的可能的分数数）。