苏州大学实验报告

院、系	计算机学院	年级专业	19计科图灵	姓名	张昊	学号	1927405160
课程名称	人工智能与知识工程实验					成绩	
指导教师	陈文亮	同组实验者	无		实验日期	2021/11/4	

实 验 名 称 产生式系统

一. 实验题目

设计并编程实验一个小型产生式系统（不要和课本例子一样，请自行设计）

要求：语言不限，领域自选

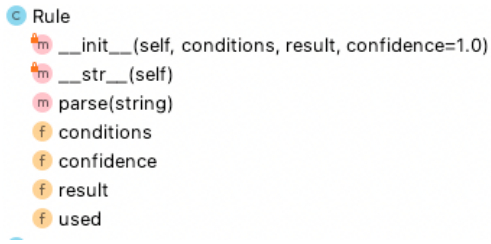
二. 实验过程

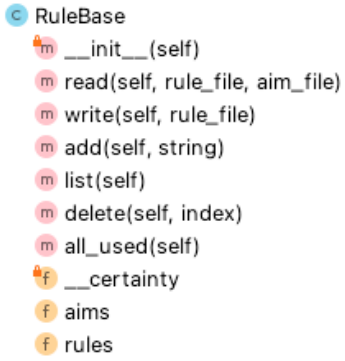
主要采用正向推理的方法，基于可信度方法实现了不确定性推理，冲突消解使用较大置信度优先、同置信度先推出优先的策略。这样一来，确定性推理就退化为不确定性推理的一种特例（置信度为 1.0 的情况）。

1. 框架实现

1) 规则（知识，产生式）：前提，结论，置信度。这里规定产生式中只有 AND，不出现 OR（将 OR 改为多个合取式的析取式，再分裂为多个产生式），格式：IF A AND B AND C THEN D [置信度] 其中置信度可以省略，默认为 1.0（即为确定性规则）

```
class Rule:
    def __init__(self, conditions: tuple, result: str, confidence: float = 1.0):
        assert -1.0 <= confidence <= 1.0, f'置信度 {confidence} 不在 [-1, 1] 的范围内'
        self.conditions: tuple = tuple(sorted(set(conditions)))
        self.result: str = result
        self.confidence: float = confidence
        self.used = False
```





2) 规则库：由规则集合以及其他辅助信息构成

```
class RuleBase:
    def __init__(self):
        self.rules: list = []
        self.__certainty: bool = True  # 判断是否全为确定性规则
        self.aims: set = set()  # 保存目标结论
```

3) 事实：内容，置信度。这里对输入进行了处理，只保留 'A 是 B' 的 B。

```
class Fact:
    def __init__(self, name: str, confidence: float = 1.0):
        assert -1.0 <= confidence <= 1.0, f'置信度 {confidence} 不在 [-1, 1] 的范围内'
        self.name: str = name.strip().split('是')[-1]  # 只保留 'A 是 B' 的 B
        self.confidence: float = confidence
```

Fact
__init__(self, name, confidence=1.0)
__str__(self)
parse(string)
confidence
name

4) 综合数据库：事实。保存了起始的输入事实，避免推理得到输入的事实。同时也实现了添加、删除的方法

```
class Database:
    def __init__(self):
        self.data: list = []
        self.starts: set = set()  # 避免推理得到输入的事实
```

Database
__init__(self)
read(self, basic_file)
write(self, fact_file)
get(self, fact)
fuzzy_name_match(name1, name2)
add(self, fact)
synthesis(cf1, cf2)
delete(self, index)
data
starts

5) 推理机：采用正向推理，建立综合数据库和规则库之间的联系。也实现了添加、删除的方法

Reasoner
__init__(self, rules, facts)
reasoning(self)
get_result(self, need_mid=False)
facts_changed(self)
__old_facts
facts
rules

6) 控制系统：菜单驱动的控制器，根据用户输入控制推理机的运行。

Controller
__init__(self, rule_file, basic_file, aim_file, file_generate_folder)
run(self)
__exec(self)
__add_rule(self)
__delete_rule(self)
__add_fact(self)
__delete_fact(self)
__show(self, option)
__save(self, option)
CHOICE
MENU
database
file_generate_folder
finished
reasoner
result
rule_base

2. 实现推理机中的推理函数

```
class Reasoner:
    def __init__(self, rules: RuleBase, facts: Database):
        self.rules: RuleBase = rules
        self.facts: Database = facts
        self.__old_facts: Database = Database()
    def reasoning(self):
        while True:
            if self.rules.all_used():  # 所有产生式都使用了
                break
            self.__old_facts = deepcopy(self.facts)
            result = self.get_result()
            if result is not None:  # 数据库中已有结论
                print(f'数据库中已有结论：{result}，匹配成功')
                return result
            for i, rule in enumerate(self.rules.rules):
                if rule.used:
                    continue
                head = [self.facts.get(name) for name in rule.conditions]
                if any(map(lambda x: x is None, head)):
                    # print('no', i, rule, head)  # DEBUG
                    continue
                print(f'数据库中存在如下结论：', ', '.join(map(str, head)))
                print(f'使用产生式：{rule}')
                # 计算不确定度
                e_confidence = min(map(lambda x: x.confidence, head))
                cf_h = rule.confidence * max(0, e_confidence)
                new_fact = Fact(rule.result, cf_h)
                print(f'将新的证据 {new_fact} 加入综合数据库')
                self.facts.add(new_fact)
                rule.used = True
            if not self.facts_changed():  # 数据库没有变化
                break
        print('循环结束')
        result = self.get_result(need_mid=True)
        if result is not None:
            print(f'数据库中已有结论：{result}，匹配成功')
            return result
        else:
            print('未能匹配到任何一个结论')
            return None
    def get_result(self, need_mid=False):  # 拿到结论
        now_list = self.facts.data[:]
        now_list.sort(key=lambda x: x.confidence)
        print('当前综合数据库:', ', '.join(map(str, now_list)))
        for fact in reversed(now_list):
```

```
                # 综合数据库中存在事实
                if fact.name in self.rules.aims:
                    return fact
            if need_mid:
                for fact in reversed(now_list):
                    # 不存在事实，但是有中间结论
                    if fact.name not in self.facts.starts:
                        return fact
            return None
        def facts_changed(self) -> bool:  # 判断数据库是否变化
            now_list = self.facts.data[:]
            old_list = self.__old_facts.data[:]
            if len(now_list) != len(old_list):
                print('综合数据库已改变')
                return True
            now_list.sort(key=lambda x: (x.name, x.confidence))
            old_list.sort(key=lambda x: (x.name, x.confidence))
            for now, old in zip(now_list, old_list):
                if not (Database.fuzzy_name_match(now.name, old.name) and
                        abs(now.confidence - old.confidence) < 1e-9):
                    print('综合数据库已改变')
                    return True
            print('综合数据库未改变')
            return False
```

3. 实现控制模块和主函数循环（没什么特别要说明的内容，详见附件）

三. 实验结果

实验环境为 Python 3.9，依赖项保存在 requirements.txt 中。

必要的输入文件位于 data 目录下，有三个：

rule.txt：规则库文件，格式为（每条规则一行，不区分关键字 if then 的大小写）：

IF E1 AND E2 AND E3 THEN D [置信度]

其中 [置信度] 可以省略，默认为 1.0

knowledge.txt：综合数据库文件，格式为（每条事实一行，于置信度用空格隔开）：

事实 置信度

aim.txt：目标结论集合，每条可能的目标结论一行。

根据输入规则和事实的不同，程序会自动识别并提供确定性和非确定性的推理。

产生式系统使用菜单驱动，有如下功能：

[0] 开始运行（进行推理，会输出运行过程）

[1] 添加规则

[2] 删除规则

[3] 添加事实

[4] 删除事实

[5] 打印规则库

[6] 打印综合数据库

[7] 保存规则库（到文件）

[8] 保存综合数据库（到文件）

下面举两个例子，输入数据存放在 data 文件夹，运行过程存放在 output 文件夹下。

1. 确定性推理 certain_demo.py：（菜单的输出省略）

/Users/holger/codes/ai_e2/venv/bin/python /Users/holger/codes/ai_e2/certain_demo.py

请选择下一步动作：

[0] 开始运行

[1] 添加规则

[2] 删除规则

[3] 添加事实

[4] 删除事实

[5] 打印规则库

[6] 打印综合数据库

[7] 保存规则库

[8] 保存综合数据库

[9] 退出程序

>>> 5

[0] IF 种子有果皮包被 THEN 被子植物 [1.0]

[1] IF 种子无果皮包被 THEN 裸子植物 [1.0]

[2] IF 没有叶 AND 没有根 AND 没有茎 THEN 藻类植物 [1.0]

[3] IF 生长在水中 THEN 水生植物 [1.0]

[4] IF 有托叶 AND 被子植物 THEN 蔷薇科 [1.0]

[5] IF 十字形花冠 AND 被子植物 THEN 十字花科 [1.0]

[6] IF 缺水环境 AND 被子植物 THEN 仙人掌科 [1.0]

[7] IF 有刺 AND 蔷薇科 AND 被子植物 THEN 玫瑰 [1.0]

[8] IF 可食用 AND 水生植物 AND 结果实 AND 被子植物 THEN 荷花 [1.0]

[9] IF 仙人掌科 AND 喜阳 AND 有刺 AND 被子植物 THEN 仙人球 [1.0]

[10] IF 水生植物 AND 药用 AND 藻类植物 THEN 水棉 [1.0]

[11] IF 可食用 AND 木本植物 AND 结红色果 AND 蔷薇科 AND 被子植物 THEN 苹果 [1.0]

[12] IF 十字花科 AND 可食用 AND 被子植物 AND 黄色花 THEN 油菜 [1.0]

[13] IF 可食用 AND 水生植物 AND 藻类植物 THEN 海带 [1.0]

[14] IF 叶片针状 AND 木本植物 AND 结果实 AND 裸子植物 THEN 松树 [1.0]

请选择下一步动作：

>>> 6

[0] 种子无果皮包被 (1.0)

[1] 没有茎 (1.0)

[2] 没有叶 (1.0)

[3] 没有根 (1.0)

[4] 生长在水中 (1.0)

[5] 可食用 (1.0)

请选择下一步动作：

>>> 0

当前综合数据库: 种子无果皮包被 (1.0), 没有茎 (1.0), 没有叶 (1.0), 没有根 (1.0), 生长在水中 (1.0), 可食用 (1.0)

数据库中存在如下结论： 种子无果皮包被 (1.0)

使用产生式：IF 种子无果皮包被 THEN 裸子植物 [1.0]
将新的证据 裸子植物 (1.0) 加入综合数据库
数据库中存在如下结论： 没有叶 (1.0), 没有根 (1.0), 没有茎 (1.0)
使用产生式：IF 没有叶 AND 没有根 AND 没有茎 THEN 藻类植物 [1.0]
将新的证据 藻类植物 (1.0) 加入综合数据库
数据库中存在如下结论： 生长在水中 (1.0)
使用产生式：IF 生长在水中 THEN 水生植物 [1.0]
将新的证据 水生植物 (1.0) 加入综合数据库
数据库中存在如下结论： 可食用 (1.0), 水生植物 (1.0), 藻类植物 (1.0)
使用产生式：IF 可食用 AND 水生植物 AND 藻类植物 THEN 海带 [1.0]
将新的证据 海带 (1.0) 加入综合数据库
综合数据库已改变
当前综合数据库: 种子无果皮包被 (1.0), 没有茎 (1.0), 没有叶 (1.0), 没有根 (1.0), 生长在水中 (1.0), 可食用 (1.0), 裸子植物 (1.0), 藻类植物 (1.0), 水生植物 (1.0), 海带 (1.0)
数据库中已有结论：海带 (1.0)，匹配成功
结论为 海带 (1.0)
运行结束，耗时：0.0049440860748291016 秒
请选择下一步动作：
>>> 6
[0] 种子无果皮包被 (1.0)
[1] 没有茎 (1.0)
[2] 没有叶 (1.0)
[3] 没有根 (1.0)
[4] 生长在水中 (1.0)
[5] 可食用 (1.0)
[6] 裸子植物 (1.0)
[7] 藻类植物 (1.0)
[8] 水生植物 (1.0)
[9] 海带 (1.0)
请选择下一步动作：
>>> 8
文件 data/knowledge-20211111011057.txt 写入成功
请选择下一步动作：
>>> 9
Bye~

2. 不确定性推理 uncertain_demo.py：（包含事实和知识的插入和删除，菜单的输出省略）

/Users/holger/codes/ai_e2/venv/bin/python /Users/holger/codes/ai_e2/uncertain_demo.py
请选择下一步动作：
[0] 开始运行
[1] 添加规则
[2] 删除规则
[3] 添加事实
[4] 删除事实
[5] 打印规则库
[6] 打印综合数据库

[7] 保存规则库
[8] 保存综合数据库
[9] 退出程序
>>> 5
[0] IF 种子有果皮包被 THEN 被子植物 [1.0]
[1] IF 种子无果皮包被 THEN 裸子植物 [1.0]
[2] IF 没有叶 AND 没有根 AND 没有茎 THEN 藻类植物 [0.9]
[3] IF 生长在水中 THEN 水生植物 [0.6]
[4] IF 有托叶 AND 被子植物 THEN 蔷薇科 [1.0]
[5] IF 吸引菜粉蝶 AND 被子植物 THEN 十字花科 [0.7]
[6] IF 十字形花冠 AND 被子植物 THEN 十字花科 [0.8]
[7] IF 缺水环境 AND 被子植物 THEN 仙人掌科 [0.6]
[8] IF 有刺 AND 蔷薇科 AND 被子植物 THEN 玫瑰 [0.8]
[9] IF 有刺 AND 蔷薇科 AND 被子植物 THEN 虎刺梅 [0.6]
[10] IF 可食用 AND 水生植物 AND 结果实 AND 被子植物 THEN 荷花 [0.6]
[11] IF 可食用 AND 水生植物 AND 被子植物 THEN 海白菜 [0.4]
[12] IF 仙人掌科 AND 喜阳 AND 有刺 AND 被子植物 THEN 仙人球 [1.0]
[13] IF 水生植物 AND 药用 AND 藻类植物 THEN 水棉 [0.7]
[14] IF 可食用 AND 木本植物 AND 结红色果 AND 蔷薇科 AND 被子植物 THEN 苹果 [0.7]
[15] IF 可食用 AND 木本植物 AND 花为白色 AND 蔷薇科 AND 被子植物 THEN 梨子 [0.8]
[16] IF 十字花科 AND 可食用 AND 被子植物 AND 黄色花 THEN 油菜 [1.0]
[17] IF 可食用 AND 水生植物 AND 藻类植物 THEN 海带 [0.8]
[18] IF 可食用 AND 水生植物 AND 藻类植物 THEN 发菜 [0.2]
[19] IF 叶片针状 AND 木本植物 AND 结果实 AND 裸子植物 THEN 松树 [0.9]

请选择下一步动作：
>>> 6
[0] 种子有果皮包被 (1.0)
[1] 生长在水中 (0.8)
[2] 结果实 (0.6)
[3] 可食用 (0.7)
请选择下一步动作：
>>> 0
当前综合数据库: 结果实 (0.6), 可食用 (0.7), 生长在水中 (0.8), 种子有果皮包被 (1.0)
数据库中存在如下结论： 种子有果皮包被 (1.0)
使用产生式：IF 种子有果皮包被 THEN 被子植物 [1.0]
将新的证据 被子植物 (1.0) 加入综合数据库
数据库中存在如下结论： 生长在水中 (0.8)
使用产生式：IF 生长在水中 THEN 水生植物 [0.6]
将新的证据 水生植物 (0.48) 加入综合数据库
数据库中存在如下结论： 可食用 (0.7), 水生植物 (0.48), 结果实 (0.6), 被子植物 (1.0)
使用产生式：IF 可食用 AND 水生植物 AND 结果实 AND 被子植物 THEN 荷花 [0.6]
将新的证据 荷花 (0.288) 加入综合数据库
数据库中存在如下结论： 可食用 (0.7), 水生植物 (0.48), 被子植物 (1.0)
使用产生式：IF 可食用 AND 水生植物 AND 被子植物 THEN 海白菜 [0.4]

将新的证据 海白菜 (0.192) 加入综合数据库
综合数据库已改变
当前综合数据库: 海白菜 (0.192), 荷花 (0.288), 水生植物 (0.48), 结果实 (0.6), 可食用 (0.7), 生长在水中 (0.8), 种子有果皮包被 (1.0), 被子植物 (1.0)
数据库中已有结论：荷花 (0.288)，匹配成功
结论为 荷花 (0.288)
运行结束，耗时：0.005290985107421875 秒
请选择下一步动作：
>>> 6
[0] 种子有果皮包被 (1.0)
[1] 生长在水中 (0.8)
[2] 结果实 (0.6)
[3] 可食用 (0.7)
[4] 被子植物 (1.0)
[5] 水生植物 (0.48)
[6] 荷花 (0.288)
[7] 海白菜 (0.192)
请选择下一步动作：
>>> 8
文件 data/knowledge-20211111010030.txt 写入成功
请选择下一步动作：
>>> 1
请输入要新增的规则：IF A AND B THEN C [0.9]
规则新增成功
请选择下一步动作：
>>> 2
[0] IF 种子有果皮包被 THEN 被子植物 [1.0]
[1] IF 种子无果皮包被 THEN 裸子植物 [1.0]
[2] IF 没有叶 AND 没有根 AND 没有茎 THEN 藻类植物 [0.9]
[3] IF 生长在水中 THEN 水生植物 [0.6]
[4] IF 有托叶 AND 被子植物 THEN 蔷薇科 [1.0]
[5] IF 吸引菜粉蝶 AND 被子植物 THEN 十字花科 [0.7]
[6] IF 十字形花冠 AND 被子植物 THEN 十字花科 [0.8]
[7] IF 缺水环境 AND 被子植物 THEN 仙人掌科 [0.6]
[8] IF 有刺 AND 蔷薇科 AND 被子植物 THEN 玫瑰 [0.8]
[9] IF 有刺 AND 蔷薇科 AND 被子植物 THEN 虎刺梅 [0.6]
[10] IF 可食用 AND 水生植物 AND 结果实 AND 被子植物 THEN 荷花 [0.6]
[11] IF 可食用 AND 水生植物 AND 被子植物 THEN 海白菜 [0.4]
[12] IF 仙人掌科 AND 喜阳 AND 有刺 AND 被子植物 THEN 仙人球 [1.0]
[13] IF 水生植物 AND 药用 AND 藻类植物 THEN 水棉 [0.7]
[14] IF 可食用 AND 木本植物 AND 结红色果 AND 蔷薇科 AND 被子植物 THEN 苹果 [0.7]
[15] IF 可食用 AND 木本植物 AND 花为白色 AND 蔷薇科 AND 被子植物 THEN 梨子 [0.8]
[16] IF 十字花科 AND 可食用 AND 被子植物 AND 黄色花 THEN 油菜 [1.0]
[17] IF 可食用 AND 水生植物 AND 藻类植物 THEN 海带 [0.8]
[18] IF 可食用 AND 水生植物 AND 藻类植物 THEN 发菜 [0.2]

[19] IF 叶片针状 AND 木本植物 AND 结果实 AND 裸子植物 THEN 松树 [0.9]
[20] IF A AND B THEN C [0.9]

请选择要删除的规则: 20
规则删除成功
请选择下一步动作：
>>> 3
请输入要新增的事实：FACT1
请输入新增事实的置信度：0.6
事实新增成功
请选择下一步动作：
[>>> 4
[0] 种子有果皮包被 (1.0)
[1] 生长在水中 (0.8)
[2] 结果实 (0.6)
[3] 可食用 (0.7)
[4] 被子植物 (1.0)
[5] 水生植物 (0.48)
[6] 荷花 (0.288)
[7] 海白菜 (0.192)
[8] FACT1 (0.6)
请选择要删除的事实: 8
事实删除成功
请选择下一步动作：
>>> 9
Bye~

四. 实验总结和反思

本次实验加深了我对产生式系统的理解，让我明白了确定性推理、不确定性推理概念，并理解了可信度方法的内容和在产生式系统中的应用。