

## 实验三 推理技术

### 一. 不确定性推理

#### 1. 简介

不确定性推理是指从不确定性的初始证据出发，通过运用不确定性的知识，推出具有一定程度的不确定性但却合理或近乎合理的结论的思维过程。目前常用的不确定性推理的数学方法主要有基于概率的似然推理、基于模糊数学的模糊推理、可信度方法，以及使用人工神经网络算法、遗传算法的计算推理等。

在不确定性推理中，除了解决在确定性推理过程中所提到的推理方向、推理方法、控制策略等基本问题外，一般还需要解决不确定性的表示与度量、不确定性的匹配、不确定性的传递算法以及不确定性的合成等问题。选择不确定性表示方法时应考虑的因素：充分考虑领域问题的特征；恰当地描述具体问题的不确定性；满足问题求解的实际需求；便于推理过程中对不确定性的推算。

简而言之，表示问题、计算问题、语义问题。表示问题指的是采用什么方法描述不确定性，这是解决不确定推理的关键一步。通常有数值表示和非数值的语义表示方法，两者都不完善。数值表示便于计算、比较，再考虑到定性的非数值描述才能较好的解决不确定问题。计算问题主要指不确定性的传播和更新，也即获得新的信息的过程。

概率方法有完善的理论，被最早用于不确定性知识的表示和处理。但因条件概率不易给出、计算量大等原因，应用受了限制。（本次实验主要用到概率方法）。

主要应用数学中条件概率、全概率公式、贝叶斯公式等来进行不确定性推理。如果前提条件用  $E$  表示，用  $H_i$  表示结论，用贝叶斯公式就可得到：

$$P(H_i|E) = \frac{P(H_i) P(E|H_i)}{\sum_i [P(H_i)P(E|H_i)]}$$

若有多个证据  $E_1, E_2, \dots, E_m$  和多个结论  $H_1, H_2, \dots, H_n$ ，则可以进一步扩充为：

$$P(H_i|E_1, E_2, \dots, E_m) = \frac{P(H_i)P(E_1|H_i)P(E_2|H_i)\dots P(E_m|H_i)}{\sum [P(H_i)P(E_1|H_i)P(E_2|H_i)\dots P(E_m|H_i)]}$$

## 2. 实验内容

参照贝叶斯网络完成，给定事件和事件之间的关系，并且给出每个事件的条件概率，根据贝叶斯公式根据上述条件求出目标概率，编写程序实现基于贝叶斯网络的推理。在这里用到的贝叶斯算法是建立在有向无环图和条件概率表的技术上实现的。参照贝叶斯网络完成，给定事件和事件之间的关系，并且给出每个事件的条件概率，根据贝叶斯公式根据上述条件求出目标概率，编写程序实现基于贝叶斯网络的推理。在这里用到的贝叶斯算法是建立在有向无环图和条件概率表的技术上实现的。

首先，给定的输入文件格式为：

```
N

rv0 rv1 ... rvN-1
0  0  1  ... 0
1  0  0  ... 1
...
0  1  1  ... 0

mat0

mat1

...

matN-1
```

在这里：

- $N$  是贝叶斯网络中随机事件的数目
- $rv$  是随机事件的名字（字符串形式表示）
- $mat$  是一个二维数组，分别表示从他的父亲到其本身的可能性概率。第一个元素表示发生的概率，第二个元素表示不发生的概率，显然两个元素相加为 1

在上述中  $mat$  即为条件概率表(Conditional Probability Table 简称 CPT)，其被设计为如下格式：

对于每个节点，如果他有  $N$  个父节点，则其 CPT 表中有  $2^N$  列，我们记为标号  $0 - (2^N - 1)$ ，其行序号的定义方法如下，利用二进制分别表示对应的父亲为是否发生，1 为发生，0 位不发生，将得到的二进制数转化为十进制代表其对应的行号。举例如下：

A 有两个父节点 C, F, 则 CPT 表如下表所示:

CPT entry	
$P(A=true C=false,F=false)$	$P(A=false C=false,F=false)$
$P(A=true C=false,F=true)$	$P(A=false C=false,F=true)$
$P(A=true C=true,F=false)$	$P(A=false C=true,F=false)$
$P(A=true C=true,F=true)$	$P(A=false C=true,F=true)$

其次, 编写程序对应的查询格式为:  $P(rvQ \mid rvE1=val, rvE2=val, \dots)$   
 $rvQ$  表示查询的条件名字, 即在  $rvE1=val, rvE2=val, \dots$  发生的条件下,  
 $rvQ$  发生的概率。

$RvEx$  表示条件的名称, 而后面的  $val$  为  $true/false$ , 分别表示发生和不发生。

最后, 输出格式为两个数据分别表示  $P(QueryVar=true|\dots)$  和

$P(QueryVar=false|\dots)$ 。例如: 0.872 0.128

例如输入如下

5

Burglar Earthquake Alarm John Mary

0 0 1 0 0

0 0 1 0 0

0 0 0 1 1

0 0 0 0 0

0 0 0 0 0

0.001 0.999

0.4 0.6

0.001 0.999

0.29 0.71

0.94 0.06

0.95 0.05

0.05 0.95

0.9 0.1

0.01 0.99

0.7 0.3

要求根据上述数据条件推理输出以下目标概率

$P(\text{Burglar} \mid \text{Alarm}=true)$

$P(\text{Alarm} \mid \text{Earthquake}=true, \text{Burglar}=true)$

$P(\text{Burglar} \mid \text{John}=\text{true}, \text{Mary}=\text{false})$

$P(\text{Burglar} \mid \text{John}=\text{true}, \text{Mary}=\text{false})$

$P(\text{Burglar} \mid \text{Alarm}=\text{true}, \text{Earthquake}=\text{true})$

输出结果：

$P(\text{Burglar} \mid \text{Alarm}=\text{true}) =$

: 0.001 : 0.999

$P(\text{Alarm} \mid \text{Earthquake}=\text{true}, \text{Burglar}=\text{true}) =$

: 0.95 : 0.05

$P(\text{Burglar} \mid \text{John}=\text{true}, \text{Mary}=\text{false}) =$

: 0.001 : 0.999

$P(\text{Burglar} \mid \text{John}=\text{true}, \text{Mary}=\text{false}) =$

: 0.001 : 0.999

$P(\text{Burglar} \mid \text{Alarm}=\text{true}, \text{Earthquake}=\text{true}) =$

: 0.001 : 0.999

**注：**

1. 本项目可用任意语言（C, Java, Python）实现，需要提交的内容要包含：  
完整的运行代码 + 实现效果截图。

## 二. 消解原理

### 1. 简介

消解是一种可用于一定的子句公式的重要推理规则。子句定义为由文字的析取组成的公式，当消解可以使用时，消解过程被应用于子母体自居对，以产生一个导出子句。消解是一种可用于一定的子句公式的重要推理规则。子句定义为由文字的析取组成的公式。一个原子公式和原子公式的否定都叫做文字。由子句构成的集合被叫做子句集。在说明消解过程之前首先说明任一谓词演算公式可以化成一个子句集。

### 2. 步骤

1. 消去蕴含符号
2. 少否定符号的辖域

3. 对变量标准化
4. 消去存在量词
5. 化为前束形
6. 把母式化为合取范式
7. 消去全称量词
8. 消去连词符号
9. 更换变量名称

### 3. 常用规则

已知两子句  $L1 \vee \alpha$  和  $\sim L2 \vee \beta$ ，如果  $L1$  和  $L2$  具有最一般合一者  $\sigma$ ，那么通过消解可以从这两个父辈子句推导出一个新子句  $(\alpha \vee \beta) \sigma$ 。这个新子句叫做消解式，它是由取这两个子句的析取，然后消去互补对而得到的。

消解推理常用的规则如下：

父辈子句	消解式
$p$ 和 $\sim p \vee q$ (即 $p \rightarrow q$ )	$q$
$p \vee q$ 和 $\sim p \vee q$	$q$
$p \vee q$ 和 $p \vee \sim q$	$q \vee \sim q$ 或 $p \vee \sim p$
$\sim p \vee p$	NIL
$\sim p \vee q$ (即 $p \rightarrow q$ ) 和 $\sim q \vee r$ (即 $q \rightarrow r$ )	$\sim p \vee r$ (即 $p \rightarrow r$ )
$B(x)$ 和 $\sim B(x) \vee C(x)$	$C(x)$
$P(x, f(y)) \vee Q(x) \vee R(f(y))$	$P(f(y)), \sigma = (f(y)/x)$
$P(x, f(y)) \vee Q(x) \vee R(f(y))$ 和 $\sim P(f(f(a)), z) \vee R(z, w)$	$Q(f(f(a))) \vee R(f(a), y) \vee R(f(y), w), \sigma = f(f(a))/x, f(y)/z$

### 4. 实验内容

理解含有变量的子句如何使用消解规则，掌握子句消解的原理和规则，通过消解从两个父辈子句中推导出一个新子句。

输入为任意两个子句，输出最后得到的新子句。

例如：

输入  $P(x, f(y)) \vee Q(x) \vee R(f(a), y)$  和  $\sim P(f(f(a)), z) \vee R(z, w)$

结果为

```

请输入子句：(以#结束)
P(x, f(y))>Q(x)>R(f(a), y)
~P(f(f(a)), z)>R(z, w)
#
输出结果为：
Q(f(f(a)))>R(f(a), y)>R(f(y), w)
请按任意键继续. . .

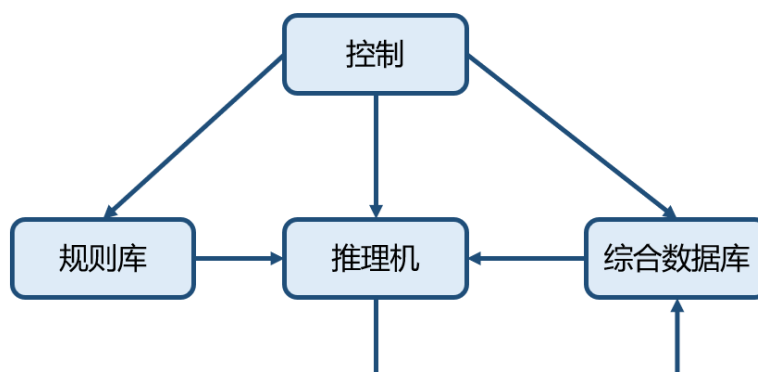
```

注：本项目可用任意语言（C, Java, Python）实现，需要提交的内容要包含：  
完整的运行代码 + 实现效果截图。

### 三. 产生式系统

#### 1. 简介

产生式系统简称产生式。它是指形如  $\rightarrow$  或 IF THEN 或其等价形式的一条规则，其中箭头左边称为产生式的左部或前件；箭头右边称为产生式的右部或后件。如果前件和后件分别代表需要注视的一组条件及其成立时需要采取的行动，那么称为条件-行动型产生式；如果前件和后件分别代表前提及其相应的结论，那么称为前提-结论型产生式。



与常规程序不同之处在于，产生式是否执行并不在事前硬性规定，各产生式之间也不能相互直接调用，而完全决定于该产生式的作用条件能否满足，即能否与全局数据库的数据条款匹配。因此在人工智能中常将产生式称为一种守护神 (demon)，即“伺机而动”之意。另一方面，产生式在执行之后工作环境即发生变化，因而必须对全局数据库的条款作相应修改，以反映新的环境条件。全部工作是在控制程序作用下进行的。现代产生式系统的一个工作循环通常包含匹配、选优、行动三个阶段。匹配通过的产生式组成一个竞争集，必须根据选优策略在

其中选用一条,当选的产生式除了执行规定动作外,还要修改全局数据库的有关条款。因此现代产生式系统的控制程序常按功能划分为若干程序。

人工智能中的推理很多是建立在直观经验基础上的不精确推理,而产生式在表示和运用不精确知识方面具有灵活性,因此许多专家系统采用产生式系统为体系结构,产生式系统被广泛地运用于专家系统等模块。

## 2. 优缺点

### 优点:

1. 模块性,每一产生式可以相对独立地增加、删除和修改;
2. 均匀性,每一产生式表示整体知识的一个片段,易于为用户或系统的其他部分理解;
3. 自然性,能自然地表示直观知识。

### 缺点:

1. 执行效率低;
2. 此外每一条产生式都是一个独立的程序单元,一般相互之间不能直接调用也不彼此包含,控制不便,不宜用来求解理论性强的问题。

## 3. 任务目标：植物识别系统

实现一个植物识别系统的创建,数据库部分包括植物特征,规则库内容,实现根据输入特征判断库中是否存在该植物,并输出查找结果。

**第一步：设计数据库** 中存储植物名及植物特征,规则库实现植物名与特征的结合,控制策略即根据采取的正向或逆向推理选取出此条规则遍历完毕后应该遍历的下一条规则。根据系统特征,采取双向推理,其中双向推理的过程是:将所有名词编号,然后用编号来组织成一条条件(规则库),遍历这些条件,根据用户给出的植物特征,进行比较,同时计算每个条件的符合程度,推理出的名词加入到已知的名词队列中,重新遍历条件,更新符合度,如果没有完全符合的条件,则寻找符合度最高的条件,进行逆向推理,询问可能的且没有在已知名词队列中的名词,进行判断,加入名词队列,重新遍历,更新符合度,直至找到属于结果类的名词,即是结果。

**第二步：创建植物数据库** 这里提供了一些数据供大家参考。

### 基本特征

- **植物个体**
  - 玫瑰、荷花、仙人球、水棉、苹果树、油菜、海带、松树
- **个体特征**
  - **区别个体的特征**：有刺、水生、喜阳、药用、木本、可食用、有白色粉末、叶片针状、结果实、黄色花；
  - **区别种类的特征种子**：有果皮、种子无果皮、无茎叶、无根、有托叶、吸引菜粉蝶、十字形花冠、缺水环境；
  - **种类**：被子植物、裸子植物、藻类植物、蔷薇科、十字花科、仙人掌科；
  - **个体**：玫瑰、荷花、仙人球、水棉、苹果树、油菜、海带、松树

### 规则库

- 种子有果皮 -> 被子植物
- 种子无果皮 -> 裸子植物
- 无茎叶 & 无根 -> 藻类植物
- 被子植物 & 有托叶 -> 蔷薇科
- 被子植物 & 吸引菜粉蝶 -> 十字花科
- 被子植物 & 十字形花冠 -> 十字花科
- 被子植物 & 缺水环境 -> 仙人掌科
- 被子植物 & 蔷薇科 & 有刺 -> 玫瑰
- 被子植物 & 水生 & 可食用 & 结果实 -> 荷花
- 被子植物 & 仙人掌科 & 喜阳 & 有刺 -> 仙人球
- 藻类植物 & 水生 & 药用 -> 水棉
- 被子植物 & 蔷薇科 & 木本 & 可食用 & 结果实 -> 苹果树
- 被子植物 & 十字花科 & 黄色花 & 可食用 & 结果实 -> 油菜
- 藻类植物 & 水生 & 可食用 & 有白色粉末 -> 海带
- 裸子植物 & 木本 & 叶片针状 & 结果实 -> 松树

**第三步：设计算法** 正向、反向推理函数所形成的双方向推理为整个代码的重难点，在该函数中涉及置信度的计算与重新计算等步骤，其中要使用特征变量和临时变量，特征变量控制循环的跳出，临时变量标记每次正向、逆向推理后所用到的特征值，方便遍历查找以及计算置信度。

**第四步：输出结果** 输入选择是否进入系统，否认则退出，确认则进入系统，打印显示用户可选特征，输出显示输入提示；用户输入特征，以-1 标志特征输入结束；系统做出判断后显示判断的植物名或征求更多特征的指令，用户确认新增特征是否存在后则返回最终结果：正确植物名、系统查找不到或按置信度依次输出可能植物；系统显示是否继续下一轮查询，否认则退出，确认则进入系统，再清屏，打印显示用户可选特征，输出显示输入提示。

**注：需要提交的内容要包含：完整的运行代码**

## 3. 实验提交要求

将实验 1-2 问题分析，实验运行情况按照实验报告样例编写并且保存到以“学号+姓名.doc/docx”形式的文件中，例如“202022408001+张三.doc/docx”；于 10 月 22 日之前交给袁雪婵同学即可。实验报告样例如下：



封面（提交时删除）

# 《人工智能》实验报告

实验名称 知识表征方法（每次实验自行更改）

学 号 \_\_\_\_\_

姓 名 \_\_\_\_\_

日期

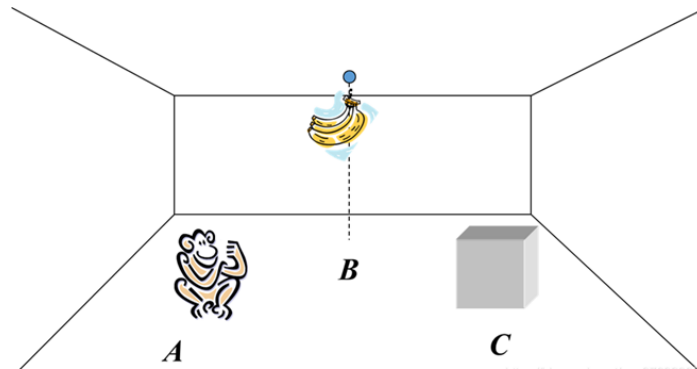
内页：（提交时删除）

## 实验一 知识表示方法

### 一、猴子摘香蕉问题

#### 1.实验内容：

房内有一个猴子，一个箱子，天花板上挂了一串香蕉，其位置如图 1 所示，猴子为了拿到香蕉，它必须把箱子搬到香蕉下面，然后再爬到箱子上。请定义必要的谓词，列出问题的初始化状态（即下图所示状态），目标状态（猴子拿到了香蕉，站在箱子上，箱子位于位置 b）。



#### 2.实验思路：

#### 3.程序清单：需加适当注释

#### 4.运行结果说明：

二、.....

...

文字用小 4 号或 4 号；程序和注释用 5 号，程序不能截图，需可再运行。