人工智能导论

Chapter 1 - 6

人工智能的定义

像人一样思考	合理地思考
像人一样行动	合理地行动

任务环境及环境的种类

- 任务环境(PEAS)四要素：
  - 性能(Performance)
  - 环境(Environment)
  - 执行器(Actuators)
  - 传感器(Sensors)
- 种类：完全可观察的 vs.部分可观察的；单agent vs.多agent；确定的 vs.随机的；片段式的 vs. 延续式的；静态的 vs. 动态的；离散的 vs.连续的

搜索算法

- 宽度优先搜索
- 一致代价搜索
- 深度受限的搜索
- 迭代加深的深度优先搜索
- 贪婪最佳搜索
- A*树搜索

搜索算法

- 爬山法
- 模拟退火算法
- 极大极小值算法
- αβ剪枝算法

Chapter 7 & 8 & 9

大纲

- 命题逻辑
  - 推导证明
  - 模型检验
- 一阶逻辑
  - 语法语义
  - 合一
  - 前向，反向链接算法
  - 归结

命题逻辑：推理

- **模型：“可能世界”**
- 如果语句α在模型m中为真，称**m满足α**
- **逻辑蕴含：** $\alpha \models \beta$：如果在使α为真的每个模型中，β也为真
- **逻辑推理算法**
  - **可靠性**
  - **完备性**

命题逻辑：模型检验

- 语句的有效性，可满足性，不可满足性的定义
- 模型检验的方法
  - 真值表枚举算法
  - 推理规则
  - 归结算法
  - 前向链接算法
  - 反向链接算法

一阶逻辑：语义语法

- 全称量词：通常，$\Rightarrow$ 是 $\forall$ 的主要逻辑连接符
- 存在量词：通常，$\wedge$ 是 $\exists$ 的主要逻辑连接符

$$\forall x\, \forall y = \forall y\, \forall x$$

$$\exists x\, \exists y = \exists y\, \exists x$$

$$\exists x\, \forall y \neq \forall y\, \exists x$$

一阶逻辑：合一

- 合一Know(John, x)和know(y, z)
  - θ ={y/John, x/z} 或者
  - θ ={y/John, x/John, z/John}
- 第一个合一置换的结果更加一般化
- 存在唯一的一个最一般合一置换(MGU)
  - MGU= {y/John, x/z}

一阶逻辑：前向，反向链接算法

- 前向链接算法：
  - 从知识库的原子语句出发，在前向推理中应用假言推理规则，增加新的原子语句，直至不能进行任何推理。
- 反向链接算法：
  - 从要证明的结论出发，不断迭代地寻找子目标，直到要证明的子目标就是已知事实

一阶逻辑：归结

- 步骤：
  - 使用合一置换将一阶逻辑语句中的变量变成常量
  - 使用命题逻辑中的归结算法进行证明

Chapter 13 & 14 & 18 & 20

大纲

- 不确定性量化
  - 概率的基本公理
  - 先验概率，条件概率
  - 贝叶斯规则
- 贝叶斯网络
  - 条件概率表
  - 枚举法精确推理

大纲

- 机器学习
  - 朴素贝叶斯
  - 贝叶斯规则
- 人工神经网络
  - 感知器算法
- 线性模型与支持向量机
  - 线性回归与线性分类
  - 支持向量机

概率公理

- 对于任何命题A, B
  - $0 \leq P(A) \leq 1$
  - $P(true) = 1$ and $P(false) = 0$
  - $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

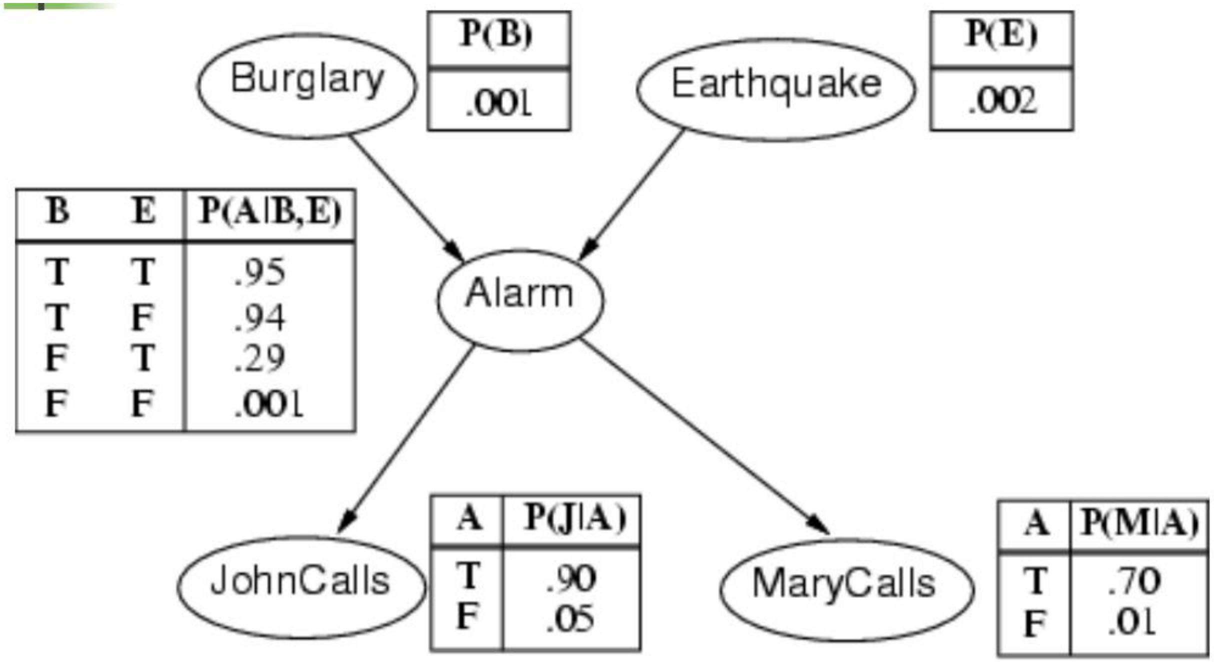
条件概率，先验概率，联合概率

- **条件概率：** $P(a \mid b) = P(a \wedge b) / P(b)$
  - e.g. P(cavity | toothache)
- **先验概率**
  - e.g. P(Cavity = true)
- **联合概率**
  - e.g. P(cavity ∧toothache)
- **独立性**
  - e. g. P(cavity | toothache,sunny) = P(cavity | toothache)

贝叶斯规则

- 贝叶斯规则
  - $P(a \mid b) = P(b \mid a)\, P(a) / P(b)$

条件概率表


Burglary
P(B)
.001
Earthquake
P(E)
.002
B E P(A|B,E)
T T .95
T F .94
F T .29
F F .001
Alarm
JohnCalls
A P(J|A)
T .90
F .05
MaryCalls
A P(M|A)
T .70
F .01

枚举法精确推理

- 枚举法：

$$P(X \mid e) = \alpha \sum_{y} P(X, e, y)$$

- e.g.　当MaryCalls=true,JohnCalls=true时，出现盗贼的概率是多少？
  - 查询变量Burglary
  - 证据变量JohnCalls,MaryCalls
  - 隐藏变量Earthquake,Alarm

近似推理

- 直接采样法
- 拒绝采样法
- 似然加权法

朴素贝叶斯

NaïveBayesLearn(*examples*)

For each target value v_j

$P'(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$P'(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

ClassfyingNewInstance(x)

$$v_{nb} = \underset{v_j \in V}{\operatorname{argmax}} P'(v_j) \prod_{a_j \in x} P'(a_i|v_j)$$

决策树

选择信息收益最大的属性进行分裂

- 属性A的信息收益为：原有熵值-属性A分裂后的熵值

$$IG(A)=H([p,n])-H(A)$$

- 一个训练数据集包含p个正样本，n个负样本

$$H([p,n])=-\frac{p}{p+n}\log_2 P(\frac{p}{p+n})-\frac{n}{p+n}\log_2 P(\frac{n}{p+n})$$

- 一个属性A的取值为A_1, A_2, …, A_v, 根据属性A的取值将训练数据集E分成E_1, E_2, …E_v, 则属性A的熵值为

$$H(A)=\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}H([\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}])$$

感知器算法

- **第一步：初始化**
  - 设定初始的权值$w_1, w_2, \cdots, w_n$和阈值θ为[-0.5,0.5]之间的随机数
- **第二步:计算激活函数值**
  - 根据输入$x_1(p), x_2(p), \ldots, x_n(p)$ 和权值$w_1, w_2, \cdots, w_n$计算输出$Y(p)$，其中p表示迭代的轮数

感知器算法

➢ 第三步：更新权值

在轮数p的错误大小

$$e(p) = Y_d(p) - Y(p)$$

理论输出

实际输出

$$\Delta w_i(p) = \alpha \cdot x_i(p) \cdot e(p)$$

学习率

$$w_i(p+1) = w_i(p) + \Delta w_i(p)$$

错误大小大于零，则要增大实际输出

线性模型与支持向量机

- 线性模型
  - 带硬阈值的线性分类器
  - 带logistic回归的线性分类器
- 支持向量机

机器学习

- 强化学习
- 深度学习

试题讲解

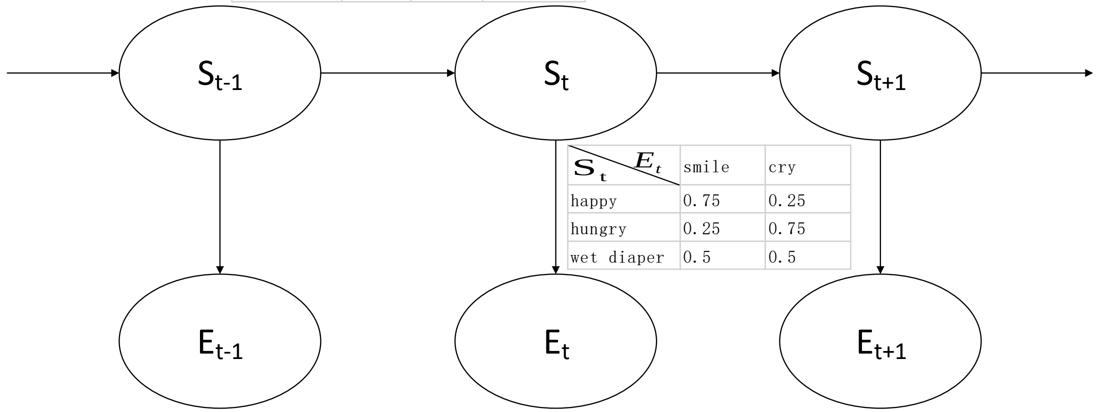
- 安迪是个三个月大的婴儿。他有时快乐(happy)，有时饥饿(hundry)，有时湿尿布(having a wet diaper)。起初，他在睡完午睡1点醒来，他很快乐。如果他快乐，那么在1个小时后他仍有50%的机率保持快乐，有25%的机会饿着肚子，另外有25%湿尿布的机会。同样，如果他饿了，一小时后他有25%的机会会快乐，25%的机会仍然饥饿，另外有50%的机会湿尿布。如果他尿布湿了，一小时后他会有50%的机会高兴，25%的机会饿着，25%的机会湿尿布。当他快乐时，他有75%的时间微笑(smile)，25%的时间哭(cry)。当他饿了，他有25%的时间微笑(smile)，75%的时间哭(cry)。当他湿尿布时，他有50%的时间微笑(smile)，50%的时间哭(cry)。
- a) 根据以上的故事画出HMM模型(画出3个时间节点即可)，并标出相应的传感器概率和转移概率；(8分)
- b) 假定“1pm: smile. 2pm: cry. 3pm: smile”.
- 这个观察序列发生的概率是多少？(5分)
- c) 对于b中对应的最有可能的隐藏状态是什么？(7分)

试题讲解

答：

a): 我们用S表示安迪的状态(happy, hundry, wet diaper)，用E来表示安迪的表情(smile,cry)，其HMM模型如下图所示：

S_{t-1} \ S_t	happy	hungry	wet diaper
happy	0.5	0.25	0.25
hungry	0.25	0.25	0.5
wet diaper	0.5	0.25	0.25



S_t \ E_t	smile	cry
happy	0.75	0.25
hungry	0.25	0.75
wet diaper	0.5	0.5

b):

t=1:smile

P(1-happy)=1*0.75=0.75

t=2:cry

P(2-happy)= (P(1-happy)*0.5)*0.25=0.09375

P(2-hungry)= (P(1-happy)*0.25)*0.75=0.140625

P(2-wet)= (P(1-happy)*0.25)*0.5=0.09375

t=3:smile

P(3-happy)= (P(2-happy)*0.5+P(2-hungry)*0.25+P(2-wet)*0.5)*0.75=0.09667969

P(3-hungry)= (P(2-happy)*0.25+P(2-hungry)*0.25+P(2-wet)*0.25)*0.25=0.02050781

P(3-wet)= (P(2-happy)*0.25+P(2-hungry)*0.5+P(2-wet)*0.25)*0.5=0.05859375

所以，该序列发生的概率为：P(1-smile,2-cry,3-smile)= P(3-happy)+ P(3-hungry)+ P(3-wet)=0.1758

c):

t=2时，MAX(P(2-cry))= P(2-hungry)

t=3时，MAX(P(3-smile))= P(3-happy)

所以，最有可能的隐藏状态为{1pm:happy,2pm:hungry,3pm:happy}

Qa？