

Intro to AI

1 人工智能概念

1.1 定义

人工智能 人工设计程序，让机器可以像人一样智慧地行动。也即合理地行动。

1.2 图灵测试

如果一位人类询问者在提出一些书面问题以后不能区分书面回答来自人还是来自计算机，那么这台计算机就通过测试。旨在设计测试以验证计算机是否真的具有智能。

1.3 Agent

Agent 通过传感器感知环境并通过执行器对所处环境产生影响。

理性 Agent 是做事正确的 Agent——对每一个可能的感知序列，根据已有的感知序列提供的证据和 Agent 具有的先验知识，理性 Agent 应该选择能使其性能度量最大化的行动。

任务环境的 **PEAS 描述**——
Performance, Environment, Actuators, Sensors.

1.4 环境

- 完全可观察的与部分可观察的
- 单 Agent 与多 Agent
- 确定的与随机的
- 片段式的与延续式的
- 静态的与动态的
- 离散的与连续的
- * 已知的与未知的

2 问题求解：搜索

2.1 搜索问题的建模方式

- 初始状态
- 行动
- 转移模型

- 目标测试
- 路径耗散

2.2 无信息搜索

广度优先搜索(BFS) 先扩展根节点，然后扩展该节点的所有后继，每次纵使扩展深度最浅的节点。完备性 b 有限时 时间复杂度 $O(b^d)$ 空间复杂度 $O(b^d)$ 最优性 单步代价相同时

一致代价搜索(UCS) 扩展路径消耗 $g(n)$ 最小的节点 n ，使用优先队列实现，贪心。完备性 b 有限且单步代价有正数下界 ϵ 时间复杂度 $O(b^{1+\lceil C^*/\epsilon \rceil}) \approx O(b^d)$ 空间复杂度 $O(b^{1+\lceil C^*/\epsilon \rceil}) \approx O(b^d)$ 最优性 是

深度优先搜索(DFS) 总是扩展搜索树的当前边缘节点集中最深的节点。完备性 在有限状态空间中避免重复状态与冗余路径的图搜索时 时间复杂度 $O(b^m)$ 空间复杂度 $O(bm)$ 最优性 否 (或许有浅层的最优解)

深度限制搜索 对深度优先搜索设置界限 l 。完备性 否 时间复杂度 $O(b^l)$ 空间复杂度 $O(bl)$ 最优性 否

迭代加深搜索(IDS) 不断地增大深度限制，直到找到目标。完备性 b 有限时 时间复杂度 $O(b^d)$ 空间复杂度 $O(bd)$ 最优性 单步代价相同时

2.3 有信息搜索

有信息搜索 (启发式搜索) 使用问题本身的定义之外的特定只是更有效地进行问题求解。使用启发函数 $h(n)$ 节点 n 到目标节点的最小代价路径的代价估计值。

贪婪最佳优先搜索 试图扩展离目标最近的节点。只用启发式信息， $f(n) = h(n)$ 。完备性 在有限状态空间中图搜索时 时间复杂度 $O(b^m)$ 空间复杂度 $O(b^m)$ 最优性 否

A*搜索 评价函数 $f(n) = g(n) + h(n)$ 为经过节点 n 的最小代价解的估计代价。完备性和最优性 需要保证可采纳性 ($f(n)$ 不会超过实际代价) 和一致性 $h(n) \leq c(n, a, n') + h(n')$ 。时间复杂度 $O(b^{ed})$ 空间复杂度 $O(b^{ed})$ 其中相对误差 $\epsilon \equiv \frac{h^* - h}{h^*}$

2.4 局部搜索

局部搜索 考虑对一个或多个状态进行评价和修改，而不是系统地探索从初

始状态开始的路径。这些算法使用于那些关注解状态而不是路径代价的问题。

爬山法 是一个简单的循环过程，不断向值增加的方向持续移动，在到达一个峰顶时终止 (邻接状态没有比它值更高的)。会在局部最大值、山脊、高原遇到问题。改进算法诸如随机爬山法、随机重启爬山法。

模拟退火 相对于爬山法没有使用最佳移动，选择随机移动，以适当弹出局部极小点。如果该移动使情况改善，则该移动被接受；否则，算法以某个小于1的概率 $e^{-\frac{\Delta E}{T}}$ 接受该移动。如果移动导致状态“变坏”，概率则成指数级下降。

开始可能允许“坏的”移动， T 的降低将会使其越不可能发生。如果调度让 T 下降的足够慢，算法找到全局最优解的概率逼近于1。

局部束搜索 记录 k 个状态而不是只记录一个。从 k 个随机生成的状态开始，每一步全部 k 个状态的所有后继状态全部被生成。如果其中有一个是目标状态，则算法停止。否则它从整个后继列表中选择 k 个最佳的后继，重复这个过程。

这个过程中有用的信息在并行的搜索线程之间传递。改进有随机束搜索。

遗传算法 是随机束搜索的变形，通过把两个父状态结合来从生成后继，而不是通过修改单一状态进行。

从 k 个随机生成的状态开始 (种群)，每一个个体用一个有限长度的字符串表示。每个状态都由他的适应度函数给出评估值，对于好的状态应当返回较高的值。根据评估值所得到的概率随机地选择父本进行繁殖 (将独立发展出来的能执行有用功能的字符区域结合起来，提高搜索粒度)，对于要配对的没对个体，在字符串中随机选择一个位置作为杂交点。最后，每个位置都会按照某个小的独立概率随机变异 (提高随机性)。

2.5 对抗搜索

MINIMAX算法 假设两个游戏者始终按照最优策略行棋，那么节点的极小极大值就是对应状态的效用值 (对MAX而言)。

MINIMAX(s) =

$$\begin{cases} \text{UTILITY}(s), & \text{终止} \\ \max_a \text{MINIMAX}(\text{RESULT}(s, a)) & \text{MAX} \\ \min_a \text{MINIMAX}(\text{RESULT}(s, a)) & \text{MIN} \end{cases}$$

$\alpha - \beta$ 剪枝 α =到目前为止路径上发现的MAX的极大值选择, β =到目前为止路径上发现的MIN的极小值选择。这样可以将复杂度从 $O(b^m)$ 降低到 $O(b^{m/2})$ 。metareasoning 的一种形式。

2.6 CSP 问题

约束满足问题 当每个变量都有自己的赋值同时满足所有关于变量的约束时, 问题得到解决。

回溯算法 用于深度优先搜索, 每次为一变量选择一个赋值, 当没有合法的值可以赋给某变量时就回溯。把叶子节点个数从 $n!d^n$ 减少到 d^n 个。

SELECT-UNASSIGNED-VARIABLE 最受约束变量或最少剩余值 (MRV) 启发式。

INFERENCE 前向检验: 只要变量 X 被赋值了, 前向检验过程对它进行弧相容检查: 对每个通过约束与 X 相关的未赋值变量 Y , 从 Y 的值域中删除与 X 不相容的那些值。

AC-3 算法 维护一个弧相容队列。首先队列中包含CSP中所有弧, AC-3从队列中弹出弧 (X_i, X_j) , 首先使 X_i 相对 X_j 弧相容。如果 D_i 没有变化, 算法则处理下一条弧。但如果 D_i 发生变化 (变小), 那么每个指向 X_i (尾) 的弧 (X_k, X_i) 都必须重新插入队列中准备检验。之所以这么做是因为 D_i 的改变可能引起 D_k 的缩小, 如果 D_i 变成了空集, 我们就知道整个CSP没有相容解, AC-3直接返回失败。否则, 我们继续检查, 试图缩小变量值域知道队列中没有弧。

弧相容性 $X_i \rightarrow X_j$, 若对 D_i 中的每个数值在 D_j 中都存在一些数值满足弧 (X_i, X_j) 的二元约束, 则称 X_i 相对 X_j 是弧相容的。

3 知识表示、自动推理

3.1 一阶逻辑

一阶逻辑 (一阶逻辑推算, FOL) 是围绕对象、关系、函数建立起来的。

符号 常量符号 (表示对象)、谓词符号 (表示关系, Brother)、函词 (表示函数); 变量、连接词、等价词、量词 (\forall, \exists)。

\Rightarrow 蕴含, $\alpha \Rightarrow \beta$ iff $\alpha \wedge \neg \beta$ 是不可满足。

3.2 知识表示

知识表示 把知识转化为计算机可接受的符号, 并以某种形式描述出来。知识的符号化过程。

语义网络 使用三种图形符号: 框 (节点)、带箭头及文字标识的线条 (弧) 和文字识别线 (指针)。

专家系统 由知识库与推理引擎组成。根据一个或者多个专家提供的知识和经验, 通过模拟专家的思维过程, 进行主动推理和判断, 解决问题。

知识图谱 以结构化的形式描述客观世界中概念、实体及其之间的关系, 将互联网的信息表达成更接近人类认知世界的形式, 提供了一种更好的组织、管理和理解互联网海量信息的能力。

3.3 贝叶斯网络

概率

$$\begin{cases} 0 \leq P(\omega) \leq 1, & \forall \omega \in \Omega \\ \sum_{\omega \in \Omega} P(\omega) = 1 \\ P(\phi) = \sum_{\omega \in \phi} P(\omega) \end{cases}$$

条件概率 (后验概率)

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \quad (1)$$

联合概率

$$\begin{aligned} P(X, Y) &= P(X|Y)P(Y) \\ P(x_1, \dots, x_n) &= \prod_{i=1}^n P(x_i|x_{i-1}, \dots, x_1) \end{aligned}$$

边缘化 (求和消元)

$$P(Y) = \sum_{z \in Z} P(Y, z) \quad (2)$$

条件化

$$P(Y) = \sum_z P(Y|z)P(z)$$

独立性

$$\begin{aligned} P(a|b) &= P(a) \\ P(b|a) &= P(b) \\ P(a \wedge b) &= P(a)P(b) \end{aligned}$$

贝叶斯规则

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

$$P(Y|X, e) = \frac{P(X|Y, e)P(Y|e)}{P(X|e)}$$

条件独立性

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$

朴素贝叶斯

$$\begin{aligned} P(Cause, Effect_1, \dots, Effect_n) \\ = P(Cause) \prod_i P(Effect_i|Cause) \end{aligned}$$

贝叶斯网络联合概率因子分解

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i|parents(X_i)) \quad (3)$$

马尔科夫毯 给定一个节点的父节点、子节点以及子节点的父节点, 这个节点条件独立于网络中所有其他节点。条件独立性判断需要考虑 d-separation。

消元法 保存中间结果。

贝叶斯采样

- 先验采样
- 拒绝采样, 拒绝所有与证据不匹配的样本
- 似然采样, 只生成证据一致事件, $w \leftarrow w \times P(X_i = x_i|parents(X_i))$
- Gibbs采样, 对 X_i 的采样条件依赖于 X_i 的马尔科夫毯中的变量的当前值, 每次修改一个变量的值, 但保持证据变量的值固定不变。条件概率计算: (1), (2), (3)。

3.4 HMM

隐马尔可夫模型是用单个离散随机变量描述过程状态的时序概率模型。

$$\begin{array}{ccccccc} y_1 & \rightarrow & y_2 & \rightarrow & \dots & \rightarrow & y_T \\ \downarrow & & \downarrow & & & & \downarrow \\ x_1 & & x_2 & & & & x_T \end{array}$$

转移概率

$$P(y_t^j = 1|y_{t-1}^i = 1) = a_{i,j}$$

发射概率

$$P(x_t|y_t^i = 1) = b_{i,1}$$

前向算法（滤波与预测）

$$P(y_{t+1}^k | x_{1:t+1})$$

$= \alpha' P(x_{t+1} | y_{t+1}^k) \sum_i P(y_{t+1}^k | y_t^i) P(y_t^i | x_{1:t})$ 开始，算出右边方程的值，再把它代入

$$\alpha_{t+1}^k = b \sum_i a^i \alpha_t^i$$

$$f_{1:t+1} = \alpha \text{FORWARD}(f_{1:t}, x_{t+1})$$

后向算法（平滑）

$$P(y_t^k | x_{1:t}) = \alpha' \alpha_t^k \beta_t^k$$

$$P(x_{t+1:T} | y_t^k)$$

$$= \sum_i P(x_{t+1} | y_{t+1}^i) P(x_{t+2:T} | y_{t+1}^i) P(y_{t+1}^i | y_t^k)$$

$$\beta_t^k = \sum_i b^i \beta_{t+1}^i a^i$$

$$b_{t+1:T} = \text{BACKWARD}(b_{t+2:T}, x_{t+1})$$

```
fv=new int[t+1]; // foward messages
b=1; // backward
    messages
sv=new vector[t+1]; // smoothed
    estimates

fv[0]=prior;
for(int i=1; i<=t; ++i)
    fv[i]=forward(fv[i-1],x[i]);
for(int i=t; i>=1; --i){
    sv[i]=normalize(fv[i]*b);
    b=backward(b,x[i]);
}
return sv;
```

维特比算法（最可能解释）

$$\max_{y_1, \dots, y_{t-1}} P(y_1, \dots, y_{t-1}, y_t^k | x_{1:t})$$

$$= \alpha P(x_t | y_t^k) \max_i \left[P(y_t^k | y_{t-1}^i) \right]$$

$$\max_{y_1, \dots, y_{t-2}} P(y_1, \dots, y_{t-2}, y_{t-1}^i | x_{1:t-1})$$

$$V_t^k = b \max_i a^i V_{t-1}^i$$

4 机器学习

4.1 强化学习

强化学习 在强化序列（奖赏和惩罚组合的序列）中学习。任务是利用观察到的回报来学习针对某个环境的最优（或接近最优的）策略。

马尔科夫过程(MDP) 对于完全可观察的环境，使用马尔科夫链转移模型和累加回报这种序列式决策问题。

效用（贝尔曼方程）在该状态得到的立即回报加上下一个状态的期望折扣效用值。

$$U(s) = R(s) + \gamma \max_a \sum_s P(s'|s, a) U(s')$$

策略 指定在Agent可能到达的任何状态下，Agent应当采取什么行动。

价值迭代算法 从任意的初始效用值开始，算出右边方程的值，再把它代入左边——从而根据他们的邻接状态的效用值来更新每个状态的效用值，直到达到一种均衡。

策略迭代算法

策略评估 给定策略 π_i ，计算 $U_i = U^{\pi_i}$

策略改进 通过基于 U_i 的向前看一步方法，计算新的MEU（最大期望效用 Max Expectation Utility）策略。

$$\pi^*(s) = \arg \max_a \sum_{s'} P(s'|s, a) U(s')$$

Q-Learning

$$U(s) = \max_a Q(s, a)$$

4.2 机器学习

机器学习 Machine Learning is a scientific discipline that explores the construction and study of algorithms that can learn from data.

分类

- 无监督学习
- 强化学习
- 监督学习

预留法随机将可用数据分为**训练集**和**测试集**。训练集用于学习算法产生 h ，测试集用于评估 h 的精度。

k 折交叉验证让每个样例都负担双重责任——既作为训练数据又作为测试数据，执行 k 轮次学习。极限情况下 $k = n$ ，称为**留一交叉验证**。

为了避免偷窥，可以将真正不会用到的数据放在测试集，剩余的数据分为训练集和验证集。

深度学习 相比于浅层学习，学习多层表示和抽象，来学习更加复杂的场景。

Bias-Variance 平衡

$$EL_2 = \text{Bias} + \text{Variance} + \text{Noise}$$

- 高Bias会导致欠拟合
- 高Variance会导致过拟合
- Noise不可消除误差

结构风险最小化（SRM）

$$\text{Bias}^2 + \text{Variance}$$

正则化 将损耗和复杂性组合成一种量纲，从而允许我们一次性发现最佳假

说，寻求更正规或更简单的函数。

$$\text{Cost}(h) = \text{EmpLoss}(h) + \lambda \text{Complexity}(h)$$

$$\hat{h}^* = \arg \min_h \text{Cost}(h)$$

L_1 正则化 极小化绝对值之和。

$\text{Complexity} = \sum_i |w_i|$ 经常将很多权重置0，从而有效地说明对应属性是不合适的。

L_2 正则化 极小化平方和。

$\text{Complexity} = \sum_i w_i^2$ 是球面的，具有旋转不变性。

特征选择 用来丢弃看起来不适宜的属性，降低模型的维度。

过拟合 我们通过机器学习建立的模型对于训练数据拟合过当，导致其在训练集上的表现很好、但是在测试集与新数据上的表现较差。 **策略** 获得更多的训练数据，从而减小噪声的影响；提供数据的质量；对原始数据的特征进行特征选择；可以尝试限制模型的复杂度；对模型加入加入正则化方法；在训练过程中进行交叉验证，在训练过程中进行交叉验证。

欠拟合 我们通过机器学习建立的模型在训练集上未学习到足够的特征与规律，从而使得模型在训练集与测试集上表现都很差。 **策略** 通过特征组合等方式，形成新的特征；过增加模型的复杂度来使得模型有更强的拟合能力；正则化是用来防止过拟合的，但当模型出现欠拟合现象时，则需要有针对性地减小正则化系数。

4.3 神经网络

更新规则

$$\begin{aligned} m' &= m - \eta \frac{\partial E}{\partial m} \\ &= m - \eta \frac{\partial E}{\partial \text{net}} \frac{\partial \text{net}}{\partial m} \\ &= m - \eta \frac{\partial E}{\partial x} \frac{\partial x}{\partial \text{net}} \frac{\partial \text{net}}{\partial m} \end{aligned}$$

4.4 无监督学习

比如：聚类、降维、自监督学习。

层次化聚类 $O(n^3)$ 优点 简单直观，可解释性好，不需要提前假设任何超参数 缺点 时间复杂度高；对于噪声非常敏感

K-Means $O(ikn)$ 优点 算法整体非常简单，时间复杂度小于层次化聚类；并且对于凸优化效果好 缺点 算法中一些超参数难以决定（cluster 数目 k ）；

cluster 的初始中心点位置的选取对于算法结果很重要；噪声对算法结果影响大；不能处理离散太大的分类；不能处理非凸数据集

DBSCAN $O(n^2)$ 优点 对于噪声的

容忍强，可以处理非凸集合，不用提前定义 cluster 的数目 k 缺点 要提前定义半径 ϵ 与点数目 minpts DBSCAN 用固定参数识别聚类，但当聚类的稀疏程度不同时，相同的判定标准可能会破坏聚类的自然结构，即较稀的聚类会被划分

为多个类或密度较大且离得较近的分类会被合并成一个聚类

PCA 无监督特征提取方法（降维），计算协方差。 $(XX^T)v = \lambda v$ ，主元由特征值大的选取。