人工智能原理与技术 (研) 课程笔记*

戴唯思

课程信息

课程重点介绍人工智能方法的一般性原理和基本思想, 包括问题求解的推理与搜索、计算智能、机器学习、Agent 等技术和方法.

1 绪论

1.1 人工智能的定义与发展

1.1.1 人工智能的定义

智能是一种认识客观事物和运用知识解决问题的综合能力, 包含感知能力, 记忆与思维能力, 学习与自适应能力, 行为能力. 智能来源于思维活动, 取决于可以运用的知识, 可以由逐步进化来实现. 人工智能目前尚无统一的定义. 智能机器: 能够在各类环境中自主地或交互地执行各种拟人任务的机器. 人工智能 (学科): 计算机科学中设计研究、设计和应用智能机器的一个分支. 近期目标: 研究用机器来模仿和执行人脑的某些智力功能, 开发相关理论和技术.

1.1.2 人工智能的起源与发展

主要理论来源: 数理逻辑, 关于计算的新思想. 标志: 1945 年夏天, 第一次人工智能研讨会在 Dartmouth 大学举行, 标志人工智能学科的诞生. 70–80 年代得到发展: 知识功能的提出与专家系统的成功应用确立了知识在人工智能中的地位. 如果机器拥有知识并且了解如何使用知识就有可能超越人脑.

1.2 人类智能与人工智能

1.2.1 智能处理信息系统的假设

信息处理系统又叫符号操作系统或物理符号系统, 符号就是模式. 一个完善的符号系统应具有下列六种基本功能: input 输入符号, output 输出符号, store 存储符号, copy 赋值符号, 建立符号结构 (找出各符号之间的关系, 在符号系统中形成符号结构), conditional transfer 条件性迁移: 根据已有符号, 继续完成活动过程

人可以看成一个智能信息处理系统. 现代计算机也具备物理符号系统的这 6 种功能.

物理符号系统的假设: 任何一个系统, 如果它能表现出智能, 那它必定支持上述 6 种功能. 任何系统如果具有这 6 种功能, 那它就能够表现出智能. 这种智能指的是人类所具有的那种智能.

*基于课堂笔记整理, 依据课件和更多资料修正和补充. 授课教师: 张选平. 编译日期: September 15, 2011

推论 1: 既然人具有智能, 那人一定是一个物理符号系统. 人表现出智能是基于他的信息处理过程.

推论 2: 计算机是一个物理符号系统, 所以一定能表现出智能. (人工智能的基本条件)

推论 3: 人和计算机都是物理符号系统, 所以可以用计算机模拟人的活动.

认知生理学研究认知行为的生理过程, 是认知科学研究的底层. 认知信息学研究认知行为的心理活动, 研究人的思维策略, 是认知科学研究的顶层. 认知信息学研究人的认知行为如何通过初级信息自然处理, 由生理活动变为心理活动及其逆过程, 是认知活动的中间层. 认知功能学研究认知行为的信息加工处理, 主要研究如何通过以计算机为中心的人工信息处理系统对人的各种认知行为进行信息处理.

1.2.2 人类智能的计算机模拟

推论 3 指出, 人和计算机所使用的物理符号是相同的, 所以计算机可以模拟人类的智能活动过程. 神经计算机可以用类似人类的方式进行思考. 一些国家希望通过对量子计算的研究生产量子计算机.

1.3 人工智能的学派

1.3.1 三大学派

- 符号主义 (Symbolicism), 又称为逻辑主义 (Logicism)、心理学派 (Psychlogism) 或计算机学派 (Computerism), 其原理主要为物理符号系统 (即符号操作系统) 假设和有限合理性原理.
- 联结主义 (Connectionism), 又称为仿生学派 (Bionicsism) 或生理学派 (Physiologism), 其原理主要为神经网络及神经网络间的连接机制与学习算法.
- 行为主义 (Actionism), 又称进化主义 (Evolutionism) 或控制论学派 (Cyberneticsism), 其原理为控制论及“感知 —动作”型控制系统的人工智能学派.

1.3.2 三大学派对人工智能发展历史的不同看法

- 符号主义认为人工智能源于数理逻辑, 主流.
- 联结主义认为源于仿生学, 特别是人脑模型的研究. 神经网络基于计算机得到实现.
- 行为主义认为源于控制论.

1.4 人工智能的研究与应用领域

- 问题求解
  国际象棋, . . .
  目前为止, 人工智能程序已知如何考虑要解决的问题, 即搜索解空间, 寻找较优的解答.
  典型技术: 搜索技术. 求解过程就是不断变换, 经过许多中间解. 将各种状态表示出来 (图), 则变成搜索问题.
- 逻辑推理与定理证明
  逻辑推理是研究最持久的子领域之一.
- 自然语言理解
  与“语音识别”等不同. 要让计算机明白一段文字表达的语义.

- 神经网络
  适合处理直觉和形象思维信息, 比传统处理方式好得多.
- 模式识别
  用计算机代替人类或帮助人类感知模式, 是对人类感知外界功能的模拟. 使一个计算机系统具有模拟人类通过感官接受外界信息、识别和理解周围环境的感知能力.
- 智能控制
  自动控制 → 智能控制, 无需 (或尽可能少) 人的干预就能独立驱动智能机器实现其目标的自动控制.
  具有复杂性, 不完全性, 模糊性 (不确定性), 包含不存在已知算法的数学过程.
- . . . . . .

2 问题求解与搜索技术

人工智能将问题的求解 (没有固定的算法) 看成搜索的过程. 问题: 目标和达到目标的方法. 问题求解需要考虑的两个基本问题: 如何用合适的状态空间表示问题? 该状态空间中, 目标状态是否出现?

2.1 问题表示

人工智能要解决的问题大多是**结构不良**或**非结构化**的问题, 对这样的问题, 智能系统的行为一般是**找到能够达到所希望目标的动作序列**, 并使其所付出的**代价最小**, **性能最好**. 问题表示描述问题的基本信息, 一般包括初始状态集合 (初始环境), 操作符集合 (从一个状态到另一个状态的动作), 目标检测函数 (确定一个状态是不是目标), 路径费用函数 (对每一条路径赋予一定费用).

2.1.1 状态空间表示法

状态: $Q = [q_0, q_1, \cdots, q_n]$, 其中的 q_i 称为一个状态变量. 给定每个分量的一组值就得到一个具体的状态. 算符: 使问题从一种状态变化为另一种状态的手段. 问题的状态空间: 表示问题全部可能的状态及其关系的图, 包含所有可能初态的集合 S, 操作符集合 F, 目标状态集合 G, 因此可将状态空间记为三元组 (S, F, G). 图的显式说明/图的隐式说明. 举例: 二阶 Hanoi 塔的状态空间表示. 问题求解就是对状态作用算符的过程, 解是由初始状态到目标状态所用算符的序列. 一个状态可能有多个后继状态 (搜索策略). 农夫过河的例子.

存在的问题: 搜索效率. 举例: n 阶 Hanoi 塔问题, n 元向量表示一个状态, 总状态有 3^n 个. 有的问题的状态空间是无限的 (连续取值的变量).

2.1.2 问题归约

已知问题的描述, 通过一系列变换把此问题最终变为一个子问题集合; 这些子问题的解可以直接得到, 从而解决了初始问题. 最后将初始问题归约为平凡的本原问题集合. 问题归约法的组成包括一个初始问题描述, 一套把问题变换为子问题的描述符, 一套本原问题描述. 举例: 三阶 Hanoi 塔问题. 表示方法: 与或图表示. 两种代表性的节点: 与节点, 或节点. 两类归约操作: 分解和等价变换. 前者形成 "与" 树, 后者形成 "或" 树. 有关术语: 父节点, 子节点, 或节点, 与节点, 终叶节点 (对应于原问题的本原结点, 可以直接求解 的叶子节点), 可解节点, 不可解节点.

2.2 图搜索策略

1. $G = \{S\}$, OPEN= $\{S\}$
2. CLOSED= {}
3. LOOP: if OPEN== {}, exit(error)
4. CLOSED.append(n=OPEN.pop())
5. if OBJ(n), exit(succ)
6. M=expand(n)/ancestor(n), G+=M
7. for each newToG(M.members): setPointerTon, addToOPEN;
   for each !newToG: evaluateIfShouldModifyPointer(by calculating prices)
   for each inCLOSE(M.members): evaluateIfShouldReversePointer
8. OPEN.sort
9. goto LOOP

搜索树 T 由第 7 步的指针来确定.

2.3 盲目搜索策略

又称无信息搜索, 没有启发信息, 一般只适用于求解简单的问题. 遵循一般搜索过程. 对同一问题的搜索过程是类似的.

2.3.1 宽度优先搜索 (BFS)

以接近起始结点的程度依次扩展 OPEN 表中的节点. "按层次搜索", OPEN 表表现为队列.

2.3.2 深度优先搜索

首先扩展最新生成的 (最深的) 节点. 不完备, 可能求不到存在的解.

2.3.3 有界深度优先搜索

规定深度优先搜索里搜索的最大深度. 根节点的深度为 0. 存在的问题: 界限难以确定 (预先确知解的最大距离; 从较小的界限尝试, 若搜不到则增大).

2.3.4 代价树搜索

对每次状态转换定义了代价 $g(x_2) = g(x_1) + c(x_1 + x_2)$, 要求解的代价最小. 策略: 从 OPEN 表中每次选择代价最小的节点. 维护 OPEN 表中的节点的代价序. 所有节点代价相同时退化为 BFS. 代价树也可以用于深度优先搜索: 从刚扩展出的子节点中扩展代价最小的子节点. 因为是深度优先搜索, 因此是不完备的.

2.4 启发式搜索

给定初始状态, 算符, 目标状态的定义和搜索空间, 如何有效地搜索这个给定空间? 启发信息的作用: 决定要扩展的下一个节点, 扩展节点时决定生成哪些后继节点, 决定修剪或抛弃的节点.

估价函数

用于评价节点重要性, 一般形式为 $f(x) = g(x) + h(x)$. $g(x)$ 是初始节点 S_0 到节点 x 已经付出的代价, $h(x)$ 是从节点 x 到目标节点 S_g 的最优路径的估计代价, 它体现了问题的启发性信息, 形式根据问题的特性确定. $h(x)$ 称为**启发函数**. $g(x)$ 指出了搜索的横向趋势, 有利于搜索的完备性. 示例: 移动牌游戏中 $h(x) = 3\sum$(W 左边 B 的个数)

2.4.1 局部择优搜索

扩展一个节点之后扩展估价最小的子节点. 实质上只需要计算 $h(x)$. 是 DFS 的推广.

2.4.2 全局择优搜索

扩展一个节点之后扩展 OPEN 表的全体节点中估价最小的节点. 是 BFS 的推广. 示例: 8 数码问题, 定义 $g(x) = d(x)$, 其中 $g(x)$ 代表 x 节点的深度, $h(x)$ 表示 x 节点格局与目标格局不同的牌数

2.4.3 A* 搜索算法

OPEN 表中的节点按估价函数 $f(x)$ 升序有序; $g(x)$ 是 $g^*(x)$ 的估计, $g(x) > 0$; $h(x)$ 是 $h^*(x)$ 的下界, 即 $h(x) \leqslant h^*(x)$. 其中, $g^*(x)$ 是从初始节点 S_0 到节点 x 的最小代价, $h^*(x)$ 是从节点 x 到目标节点的最小代价. 如果有多个目标节点则取最小的 $h^*(x)$. 在搜索过程中, 恒有 $g(x) \geqslant g^*(x)$, 随着更多信息的获得, $g(x)$ 呈下降趋势. $h(x)$ 的确定依赖问题领域的启发信息. $h(x) \leqslant h^*(x)$ 保证了 A* 算法能找到最优解.

搜索算法的可纳性: 对于可解状态空间图来说 (初始节点到目标节点有路径存在), 如果一个搜索算法可以在有限步内中止并能找到最优解, 则称为**可纳的**. A* 算法是可纳的.

在满足 $h(x) \leqslant h^*(x)$ 时, $h(x)$ 越大, 携带的启发性信息越多, 扩展的节点数越少, 搜索的效率越高.

可以证明, 若 $h(x)$ 满足单调性: $h(S_g) = 0$; 设 x_j 是 x_i 的任意子节点, 则有 $h(x_i) - h(x_j) < c(x_i, x_j)$ 即 $h(x_i) \leqslant h(x_j) + c(x_i, x_j)$, 则可以得到两个结论: 若 A* 算法选择扩展 x_n 节点, 则 $g(x_n) = g^*(x_n)$; 由 A* 算法所扩展的节点序列的 f 值是非递减的.

2.5 与/或树的搜索策略

叶子节点对应本原问题时可以直接求解. 搜索目标是一棵 "可解/不可解树". 1. 把原始问题作为初始节点 S_0, 并把它作为当前节点 2. 应用分解或等价变化算符对当前节点进行扩展 3. 为每个字节点设置指向父节点的指针 4. 选择一个合适的子节点作为当前节点, 反复执行第 2 步和第 3 步, 在此期间要多次调用可解和不可解标识过程, 直到初始节点被标识为可解或不可解节点为止. 当一个节点已经标识为可解节点, 则其不可解的后裔节点就不再有用. 同样, 当一个节点已经标识为不可解节点, 则其全部后裔节点就不再有用.

盲目搜索: 广度优先搜索, 深度优先搜索; 启发式搜索: 有序搜索.

有序搜索: 求取代价最小的解树. 定义节点的代价, 终止节点 $h(x) = 0$, 或节点 $h(x) = \min\{c(x, y_i) + h(y_i)\}$, 与节点 $h(x) = \Sigma(c(x, y_i) + h(y_i))$ 或 $h(x) = \max\{c(x, y_i) + h(y_i)\}$, 不可扩展的终止节点 $h(x) = \infty$. **希望树**: 有希望成为最优解树的节点构成的子树, $S_0 \in T$, 若

$x \in T$, 则若 x 为或节点, 则 $\min\{c(x, y_i) + h(y_i)\}$ 的 $y_i \in T$; 若 x 为与节点, 则其子节点全在 T 中. 与/或树的有序搜索就是不断选择和修改希望树的过程.

2.5.1 博弈树的启发式搜索

博弈树: 二人零和, 全信息, 非偶然博弈: A,B 双方轮流采取行动, 结果有三种: A 胜, A 败, 和; 任何一方都了解当前的格局及历史格局; 任何一方都根据当前格局采取最有利于自己的行动. 博弈的初始格局是初始节点; 在博弈树中, 或节点和与节点是逐层交替出现; 能使自己获胜的终局为本原问题, 相应节点为可解节点, 使对方获胜的终局对应的节点为不可解节点.

极大极小分析法: 为某一方寻找一个最优的行动方案, 需要考虑每一方案实施后对方可能采取的行动并计算可能的得分; 需要根据问题的特性定义一个估价函数用来估计每种格局的得分; 当端节点的得分计算出来后, 根据与/或节点推算父节点及其祖先节点的得分; 如果一个方案能获得较大的倒推值, 则它就是当前最好的行动方案.

α-β 剪枝技术: 对于与节点, 取子节点中的最小倒推值作为倒推值的上界, 称为 β 值; 对于或节点, 取子节点中的最大倒推值作为倒推值的下界, 称为 α 值. 一般规律: 任何或节点 x 的 α 值如果不能使其父节点的 β 值降低, 则对节点 x 以下的分支停止搜索, 这种剪枝称为 β 剪枝; 任何与节点 x 的 β 值如果不能使其父节点的 α 值升高, 则对节点 x 以下的分支停止搜索. 这种剪枝称为 α 剪枝. https://secure.wikimedia.org/wikipedia/en/wiki/Alpha-beta_pruning

```
function alphabeta(node, depth, alpha, beta, Player)
    if  depth = 0 or node is a terminal node
        return the heuristic value of node
    if  Player = MaxPlayer
        for each child of node
        alpha := max(alpha, alphabeta(child, depth-1, alpha, beta, not(Player) ))
            if beta <= alpha
                break                             (* Beta cut-off *)
        return alpha
    else
        for each child of node
        beta := min(beta, alphabeta(child, depth-1, alpha, beta, not(Player) ))
            if beta <= alpha
                break                             (* Alpha cut-off *)
        return beta
(* Initial call *)
alphabeta(origin, depth, -infinity, +infinity, MaxPlayer)
```

2.6 搜索的完备性与效率

2.6.1 完备性

对于一类可解的问题和一个搜索过程, 如果运用该搜索过程一定能求得该类问题的解, 则称该搜索过程为**完备的**, 否则为不完备的. 完备的搜索过程称为**搜索算法**, 不完备的搜索过程不是算法, 称为**过程**. 广度优先搜索、代价树的广度优先搜索、改进后的有界深度优先搜索以及 A* 算法都是完备的搜索过程, 其它搜索过程都是不完备的.

2.6.2 搜索效率

一个搜索过程的搜索效率不仅取决于过程自身的启发能力, 而且还与被解问题的有关属性等多种因素有关.

外显率: $P = L/T$, L 为从初始节点到目标节点的路径长度, T 为整个搜索过程中所生成的节点总数. 外显率反映了搜索过程中从初始节点向目标节点前进时搜索区域的宽度. 当 $L = T$ 时, $P = 1$, 表示搜索过程中每次只生成一个节点, 它恰好是解路径上的节点, 搜索效率最高. P 越小表示搜索时产生的无用节点愈多, 搜索效率愈低.

有效分枝因数: $B + B^2 + B^3 + \cdots + B^L = T$, 表示在整个搜索过程中每个有效节点平均生成的子节点数目. $B = 1$ 时, $L = T$, 此时所生成的节点数最少, 搜索效率最高. 当 B 一定时, L 愈大则 P 愈小; 当 L 一定时, B 愈大则 P 愈小; 对同一个 L 而言, B 愈大则 T 愈大.

2.7 衡量搜索策略性能的准则

1. 完备性
2. 尽量避免无用搜索. 增强搜索的目的性, 尽量避免产生及考察那些无用的节点。
3. 控制开销小. 要求搜索策略实现简单, 选择及调度可用知识的开销尽可能小