
从算法复杂性到通用人工智能

李熙

北京大学哲学系

xieshenlixi@163.com

1956 年夏, McCarthy, Minsky, Simon, Newell, Shannon, Solomonoff 等十人召开了为期一月的达特茅斯会议, 这标志着人工智能 (Artificial Intelligence) 这门学科的正式诞生, 起初, 人工智能专家们野心勃勃, 目标就是创造出不逊于人类智力水平的智能机器, 但随后人工智能的每一个新浪潮, 都经历了一个从盲目乐观到彻底沮丧的轮回, 通用问题解决器、感知机技术、基于规则的专家系统、遗传算法、神经网络、概率图模型、支持向量机莫不如此. 自上世纪七十年代开始, 除了少数人在坚持传统外, 主流人工智能界已变得越来越谨慎, 目标开始转移, 主要致力于针对某种 (或某类) 特殊问题、特殊功能、特殊领域设计算法, 解决问题, 这完全脱离了人工智能先驱们如图灵、McCarthy, Minsky, Simon, Solomonoff 等人的预期规划. Minsky 甚至说, “人工智能的研究从七十年代开始已经 ‘脑死亡’ 了”. 现实环境是千差万别、纷繁复杂的, 对于如此多样的环境, 由人对每种环境设计具体算法, 这种工作什么时候是尽头? 针对特定领域设计的算法是否具有足够强的泛化能力、能够灵活地迁移到其它环境并迅速适应? 针对不同功能 (甚至是采用完全不同的技术) 设计的不同算法能够进行协调整合吗? 如何整合? 有没有可能设计一种能够对各种环境、各种目的迅速适应并作出反应、处理信息、达成目的的智能主体?

经过半个世纪的发展, 随着人工智能各个子领域的技术积累日渐成熟, 大约从 2004 年开始, 以 “通用性” 为目标的通用人工智能 (Artificial General Intelligence) 开始复兴.

Kurzweil 2005 年提出了他的奇点理论, 他相信技术进步的指数速度不会衰减, 并据此对人工智能的未来作出了极度乐观的预测, 这甚至引起了广大民众对通用人工智能的关注.

在学术领域, 无论是工程实践方面, 还是理论创新方面, 近年来, 新的进展不断出现. 尤其最近, DeepMind 公司把深度学习和加强学习结合起来, 用深度学习编码特征, 用加强学习寻找策略, 在 Stella 模拟机上让机器自己玩 Atari 2600 的游戏, 结果不仅在很多游戏上战胜了其它机器, 甚至在很多游戏上战胜了专业的游戏高手. 这显示了一定的通用性, 但它只针对 Atari 2600 上的简单游戏, 并对这一类游戏做了很多预处理, 所以离真正的通用智能仍然很遥远.

如此众多的方案各自为战, 虽然工程实践上互相有所借鉴, 但理论方面各有自己的体系, 归根到底是对 “智能” 本身的理解不同. 抽象的 “智能” 是个很难精确定义的复杂概念, 笼统地说, 人的智能涉及模式识别、分类、学习、记忆、归纳、类比、泛化、联想、规划、优化、创新、演绎推理、问题解决、语言处理、生存、繁衍等方面, 试图模拟人脑或模拟人类智能的各种功能模块而构建智能主体可以看作一种自下而上实现人工智能的方式, 有没有一种自上而下的从全局出发的对智能的研究而自动

包含各具体的功能模块或使得各种“智能表象”自动涌现出来的方式？抽象地看，智能主体是能够成功实现某种“目标”的主体，或说是能够在各种未知的环境中成功获取“最大效用”的主体。但“效用”是什么？如何最大化？未知的环境如何估测、适应？各种可能的环境有哪些？有没有一个能在各种可能的环境中表现最优的“最智能”的主体？

2005 年，通用人工智能领域的代表人物 Marcus Hutter[Hut05] 第一次给出了真正能适应各种不同环境的通用智能主体的自上而下的、严格形式化的、可靠的、通用的、无参数的数学模型，而且只用了下面一个公式：

$$a_k := \arg \max_{a_k} \sum_{o_k r_k} \dots \max_{a_m} \sum_{o_m r_m} [r_k + \dots + r_m] \sum_{q: U(q, a_1 \dots a_m) = o_1 r_1 \dots o_m r_m} 2^{-\ell(q)} \quad (AIXI)$$

通过 AIXI，上面提到的难题都可以得到某种解释。虽然 AIXI 本身是不可计算的，但是，在事先不告知具体游戏规则的情况下，即使对于 AIXI 的某种可计算的简单变种 MC-AIXI-CTW[Ven+11]，仍然可以完全通过试错法自己总结学习规则，玩好 Cheese Maze, TicTacToe, Pacman, Kuhn Poker 等各种小游戏。Hutter 本人将 AIXI 看作通用人工智能的“黄金标准”或指路明灯，经过这些年的发展，AIXI 甚至变成了 Hibbard, Yudkowsky 等人研究人工智能伦理的理论基础。如此强大的智能背后究竟隐藏着什么玄机？下面就让我们一步一步揭开通用智能主体 AIXI 的神秘面纱。

Hutter 的通用智能模型 AIXI 的核心是 Solomonoff 的通用归纳模型，事实上，将 Solomonoff 的通用归纳与序贯决策理论相结合就得到了通用智能模型 AIXI。序贯决策理论是一种研究在客观概率分布已知但具体状态不确定的动态环境中主体如何寻求最大化期望效用的决策理论，它从初始状态开始，每个时刻根据所观察到的状态和以前状态的记录，依照已知的概率分布，从一组可行方案中选用一个能够获得最大化期望效用的最优方案，接着观察下一步实际出现的状态，然后再作出新的最优决策，如此反复进行。但最关键的问题是，如果这种客观的概率分布未知怎么办？这时我们能否借助某种“主观”概率代替“客观”概率，然后在这种“主观”概率下寻求期望效用最大化？这恰是 Solomonoff 的“算法概率”大显身手的地方。那么，“算法概率”究竟是何方神圣呢？顾名思义，它是某种与“算法”相关的概率，可“概率”又是怎么与“算法”扯上关系的呢？我们知道，如果已知信源的概率分布，那么可以设计某种使得期望码长最短的最优码，比如霍夫曼码。笼统地说，这是通过对高概率事件赋予短的编码、低概率事件赋予长的编码实现的，而“算法”完全可以理解为其“输出”的“编码”，如果我们把借助已知概率设计最优码的过程反过来，设想首先知道的不是概率而是编码（算法），那么就可以通过“算法”反向诱导出某种“主观概率”，然后对所有可能的“算法”诱导出的所有可能的“主观概率”进行加权平均就得到了 Solomonoff 的“算法概率”。但为什么这种把霍夫曼编码思想反过来诱导出的“概率”会有用呢？因为科学是压缩的艺术，“简单性”是科学的基本假设，探索世界背后的运行“模式”就是在寻找简单的算法，这种把短的算法赋予高的概率的方式诱导出的是一种对各种可能世界/猜想的“先验信念”，这里体现的正是奥卡姆剃刀的“简单性”哲学，而“算法概率”正是综合权衡了各种可能的“算法”诱导出的各种可能的“主观概率”，归纳是一个不断“试错”的学习过程，“算法概率”使得我们可以根据经验不断“修正信念”、逼近“真理”。有了“算法概率”做武器，再借助序贯决策理论帮助我们追逐效用，能够自动适应各种可能环境的超级智能体 AIXI 就诞生了。

简言之, 通用智能的核心是通用归纳, 通用归纳将归纳转化为预测, 而预测的关键是压缩, 压缩可以理解为对数据的建模或编码表示, 它依赖于对模式的掌握, 模式可以用算法来衡量, 从数据到程序是编码, 从程序到数据则是解码. 编码越好 (即压缩越短) 则预测越准, 预测越准行为才越有效. 与智能相关的其它要素, 诸如分类、类比、联想、泛化等都可以理解为对模式的追求, 这些都可以在追求最大压缩的过程中涌现出来, 所以不是基本的. 但找寻最短编码的过程不是一个能行的过程, 所以我们只能通过试错不断逼近, 逼近的过程可以理解为一个信念修正的过程, 这可以通过贝叶斯更新来处理, 信念修正之前的“先验信念”的大小则取决于“模式”自身的“简单性”.

上面是对通用智能模型 AIXI 极其核心通用归纳的简单介绍, 下面详细展开.

Solomonoff 把归纳问题转化为序列预测问题而不是归纳出某个具体的模型. 序列预测问题是最常见的智商测试题型, 首先, 让我们从几道常见的智商测试题说起.

1 几道数字推理题

万物皆数.

— 毕达哥拉斯

序列预测问题是最常见的智商测试题型, 首先来看几道常见的智商测试题.

- (i). 1, 3, 5, 7, 9, 11, 13, 15, (?)
- (ii). 0, 1, 0, 1, 0, 1, 0, 1, 0, (?)
- (iii). 1, 1, 2, 3, 5, 8, 13, 21, (?)
- (iv). 1, 4, 1, 5, 9, 2, 6, 5, 3, (?)
- (v). 12, 23, 35, 47, 511, 613, (?)
- (vi). (7111, 0), (8809, 6), (2172, 0), (6666, 4), (1111, 0), (2222, 0), (7662, 2), (9313, 1), (0000, 4), (8193, 3), (8096, 5), (4398, 3), (9475, 1), (0938, 4), (3148, 2), (2889, ?)

答案及解析:

- (i). 17.
奇数列.
- (ii). 1.
0, 1 交替.

(iii). 34.

斐波那契数列.

(iv). 5.

圆周率小数点后的数位.

(v). 717.

数位拆分, 第一位自然数序列, 后面则为素数列.

(vi). 5.

数“圈圈”的个数.

前四个问题比较简单, 只要能够识别出给定数列背后的递推公式, 后面的数位就可以“能行”的计算出来. 这是一个先“归纳”再“预测”的过程, 虽然最后的目的是准确预测下一位数字, 但关键的步骤是归纳出预期的递推公式. 这里涉及两个核心要素: 预先给定的数列 (现象)、待估的递推公式 (模型). 归纳推理就是这种从现象到规律 (模型、假设), 从混沌到有序, 从结果到原因的过程.

数列 (现象) 是初始给定的, 所以真正需要解决的是 — 怎么找到那个递推公式 (“能行”的“模式”)?

在开始寻找递推公式 (“能行”的“模式”) 之前, 一个古老的哲学的问题 — 休谟问题需要解决, 那就是 — 归纳推理能保证确定性吗? 以问题 (i) 为例, 奇数列通项公式 $2n - 1$ 是我们想要的递推公式, 可是不难发现, 下面这个递推公式也满足给定的数串.

$$2n - 1 + \prod_{i=1}^8 (n - i)$$

多个“模式”符合同一系列“现象”, 这该如何取舍?

2 奥卡姆剃刀 — 哲学悖论? 还是科学方法论?

如无必要, 勿增实体.

— 奥卡姆

通过上节的几个智商测试题我们看出, 要解决序列预测问题, 这里涉及的不是一个问题而是两个问题:

问题一、如何寻找以递推公式为代表的“能行”的“模式”?

问题二、如果递推公式 (“能行”的“模式”) 不唯一该如何取舍?

问题二是归纳推理能否保证确定性的哲学难题。其实在休谟之前，莱布尼茨就意识到了这个问题。他提出了如下的曲线拟合悖论：一张纸上的任何有限个点总是能找到无限条曲线把它们串起来。这意味着，给定任何有限的观测数据，总有无限的归纳推广方式，总可以有无限的“规律”符合有限的观测材料，“同果”未必“同因”。究竟哪条规律才是决定这些材料的真正原因呢？为什么大家倾向于接受 17 作为问题 (i) 的答案而不接受 $17 + 8!$ ？维特根斯坦“遵守规则”的悖论与此紧密相关，任何规则都无法唯一确定行为方式，有无穷多的行为方式可以和这条规则相符或者相违。表面上看，维特根斯坦“遵守规则”的悖论跟莱布尼茨“曲线拟合”的悖论说的是相反的两件事，维特根斯坦说有无限的行为方式可以符合或违反一条规则，莱布尼茨说可以有无限多的规则符合给定的有限多的行为。但“遵守规则”的悖论的根源事实上是，主体不能通过有限的行为方式习得唯一不变的规则概念，所以规则的语义概念不明确，规则本身只能通过主体在社会环境中各种“遵守规则的行为”的过程中获得“隐定义”。

哲学家古德曼的“绿蓝”悖论也可以看作类似的问题——到目前发现的所有翡翠都是绿的，但这个事实本身与以下两个假设吻合得同样好：

假设一、所有翡翠都是绿的；

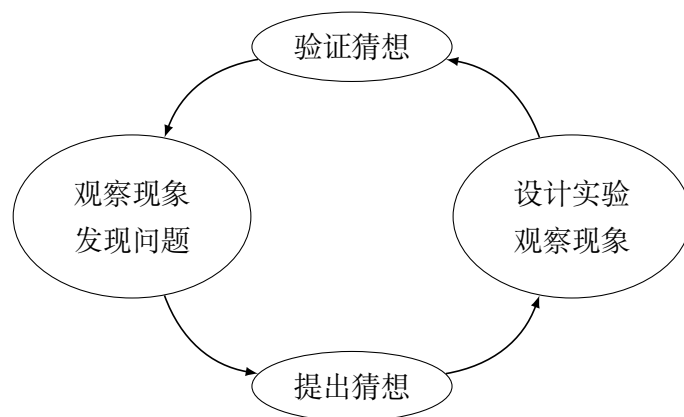
假设二、所有翡翠都是“绿蓝”的——即，在未来的某个时间点（比如 2050 年）前所有的翡翠都是绿的，其后都是蓝的。

因为诸如此类的悖论，归纳推理的有效性一直饱受质疑。莱布尼茨也认为有限的观测无法确保一般真理的普遍必然性，过去发生的未必将来同样发生。休谟认为，归纳仅仅是一种心理习惯，人不可能借助归纳推理确保结论的确定性，从个别到一般的推理不具有必然性。归纳推理需要借助于“未来与过去的相似性”，但这本身顶多是一个归纳结论，它的有效性如果再以归纳的方式辩护则陷入了逻辑循环。穆勒试图通过引进“自然的齐一性”的假设作为归纳推理的基础。但“齐一性”的精确含义究竟是什么？在莱布尼茨那里所有可能世界都必须符合“充足理由律”，都依照“数学规律”做“机械的”运转。在无穷多的“机械”的“数学规律”之间，莱布尼茨动用了奥卡姆剃刀——强调“简单性”的标准，认为，“规则”之所以为“规则”必须“简单”，如果允许任意高程度的“复杂性”的话，那么“规则”也就不能称其为“规则”而趋近“随机”了，规律性的缺乏或者说复杂性的过高将导致混沌甚至“随机”。在真正的原因不明朗的情况下，对于一个表述“简单”但解释力、预测力强的假设，如果所有已知的现象都跟它相符合而没有现象与它相违背，那么，在实践过程中，在与此矛盾的现象产生之前，这个假设就可以暂时拿过来当原因用。而且，莱布尼茨相信连续律，自然从不做跳跃，只要数据或材料（输入）充分接近，那么结论（输出）的误差也可以随之被降低到任意小，随着科学的发展，所选择的假设可以在极限处逼近真实的原因。¹

莱布尼茨的这种观点与当今科学方法论的主流观点契合。科学哲学家波普尔认为，科学的发展遵循如下规律：

$$P_1 \longrightarrow T_1 \longrightarrow E_1 \longrightarrow P_2 \longrightarrow \dots$$

¹ 参看 *Leibniz: Discourse on Metaphysics*.



针对问题 P_1 , 可以提出许多相互竞争的可错的猜想/假设或尝试性的理论 T , 然后逐一考察这些理论, 根据当前的观测消除错误、排除掉那些与现象不相容的猜想, 对于能够解决现存的问题的那些理论也可以排一个序, “可证伪性”越高的理论越值得重视, 然后可以再用他们进一步尝试解决产生的新问题 P_2 . 随着问题的不断深入, 越“可证伪”但能够经受得住严格的反复检验的理论越逼近“真理”, 科学就这样通过“试错法”而不断前行. 这种不断“提出猜想”反复“试错”的过程类似于生物的“基因变异”与“自然选择”的进化过程: 适合生物生存的变异基因得以保存, 不适合的被淘汰; 适合解决问题的理论得以留存, 不适合的被排除. 作为“全称”命题的真理不能被有限的事例证明, 但可以被“证伪”, 所以在此过程中重要的是对问题解决的“适应性”, 而不是寻求一劳永逸的证明. 但“可证伪性”是一个非常模糊的概念, 很难给出度量标准. 波普尔希望能用“简单性”代替“可证伪性”, 直观上越“简单”的理论越“可证伪”, 也就是说, 各种猜想应该按照“简单性”排序 $T_1 < T_2 < \dots$, 但“简单性”同样缺乏一个客观的衡量标准.

不可否认, 在现实生活中, 经验养成的心理习惯使人获益的时候多, 受损的时候少, 这是自然选择的结果. 虽然“简单性”缺乏一个客观的标准, 虽然“归纳仅仅是一个心理习惯”, 但对“简单性”的心理偏好得到了一些格式塔心理学实验的支持. 格式塔心理学家们通过一系列实验总结出, 人在知觉时倾向于按照一定的模式把感觉材料组织为一个有机的整体、而不是知觉为一堆个别的感觉材料的简单集合. 当人在将感觉材料组织为整体的过程中一般遵循某些经验法则, 如大小、形状、色彩的恒常性, 以及图形-背景、邻近、相似、连续、闭合、对称等等, 而这些都可以看作完形趋向法则的不同表现形式, 也就是人的认知有趋向于简单有序、闭合完整的倾向. 比如人在看到图1的第一张图片时倾向于自动脑补出第四张图片. 所以从认知上看, “简单性”体现在知觉空间的邻近性、连续性、完整性、事物规则的相似性、对称性 (比如镜像、平移、旋转、伸缩等变换下的不变性等等) 以及与以往经验的吻合性等特征上.

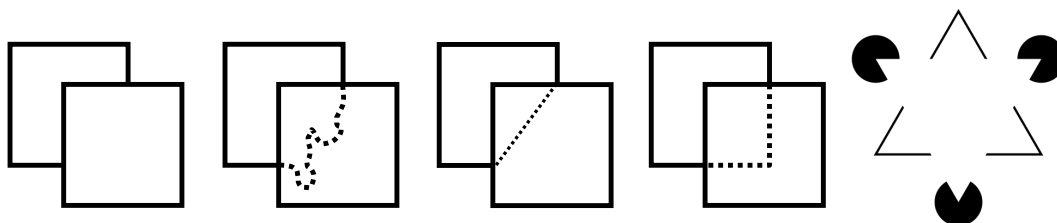


图 1: 完形

连续、闭合、相似、对称等仅仅是一些最基本的直观性质，具有这些属性的模式更易被大脑记忆存储，但这些远没有穷尽“简单性”的所有内涵。如何才算穷尽“简单性”的内涵？这就不能只考虑直观易见的基本模式，而要考虑所有可能的模式。

3 压缩 vs 预测、编码 vs 概率

手扶拐杖的外星绅士造访地球，想把地球文明传播到自己星球。临别时，地球人慷慨赠送百科全书：“全部人类文明尽在其中！”绅士谢绝：“不，谢谢。我只需在手杖上点上一点”。

在界定所有可能的“模式”之前，请先阅读上面的小故事，体会一下我们“主观”理解的“简单性”。历史悠久的地球文明直观上非常“不简单”，需要很多很多厚厚的百科全书才能记载，外星绅士手杖上的一点却让人觉得再“简单”不过，但二者却是等价的，仅仅通过“一点”外星绅士就能把整个地球文明无损的带走。这是怎么回事呢？答案是编码！通过编码进行数据压缩！

比如考虑一个数据压缩问题，如何对英文版的《战争与和平》用数字串 0,1 进行编码，要求翻译成数字串后还可以译回原来的英文，而且要让翻译后的数字串尽量短。

进行这种编码的最直观、最直接可用的信息可能是英文字母的出现的频率，有些字母像 *e, t, a* 出现的频率会远远高于某些字母如 *j, q, z*。试图让编码后的文本更短，就需要采用某种变长码，对频繁出现的字母分配较短一些的描述，而对不经常出现的字母分配较长一些的描述。但要使编码最短仅仅考虑字母的出现频率是不够的，语言不是掷色子产生的伯努利过程，考虑词的层面，英文中经常会有某些固定搭配。某些词的后面跟随另一些词的频率非常高。比如 *deal* 后面紧跟的词是 *with* 的可能性很大。上升到句子的层面也会有类似的规律，比如某些连词的固定搭配，看到 *if* 引导的从句后，下一个从句极有可能是由 *then* 引导的。任何类似的“规律”或“模式”，不管是字母层面、词语层面、短语层面还是句子、篇章层面，都可以帮助压缩编码后的文章长度。所以分析“简单性”必须考虑所有可能的模式，不管它们隐藏的有多深。

如果已知客观概率分布 P ，可以证明，编码的平均码长必大于等于某个下界，这个下界就是香农熵。采用某些好的编码方式可以渐近的接近甚至在某些理想情况下达到这个下界。如果忽略码长必须是某个整数的这个限制，那么，对于 x 采用长度为 $-\log P(x)$ 的方式编码，就可以达到这个下界。用这个长度或接近这个长度进行编码是可行的，比如霍夫曼码就可以以类似的码长实现平均码长最小的目标。它就是对高概率事件赋予短的编码，低概率事件赋予长的编码。

不管上帝以何种概率分布创世，假如他对可能的历史的编码方式是最优的，奥卡姆剃刀就有意义。假如客观的概率分布未知，而已知历史经验 x ，如果对 x 有一些编码方式 $\text{code}(x)$ ，然后把霍夫曼编码的思想反过来，对于短的编码应该赋予高的“概率”/“信念”，对于长的编码应该赋予低的“概率”/“信念”，这样就可以诱导出某种“概率”/“信念”

$$P(x) := 2^{-\ell(\text{code}(x))}$$

其中 $\ell(x)$ 表示 x 的长度。

根据奥卡姆剃刀,简单的“猜想” $\text{code}(x)$ 更“似真”. $\text{code}(x)$ 越短,它的真理性越高,那么,这种诱导出概率反映的是对产生 x 的真实分布的猜测. “理想的编码”可以诱导出最“似真”的“信念”,即更接近上帝创世采用的“客观概率”. 但什么是“理想的编码”方式呢?

4 界定“模式”——可计算的“路径”

上帝一计算,世界就创造出来了.

— 莱布尼茨

通过上节的讨论我们发现,复杂的事物要想“简单”的表达必须先将其“压缩”,“压缩”可以通过“编码”来实现,而“编码”与“概率”有着奇妙的联系,虽然我们现在还不知道这种“概率”究竟有多大用、能否帮助我们最终解决“序列预测”的问题,但我们猜测,通过“理想的编码”诱导出“概率”在某种意义上反映了序列(或现象)的真实分布,所以我们首先应该讨论清楚什么是“理想的编码”.

给定英文版的《战争与和平》,只针对这本小说来说,直观上“理想”的编码方式说的应该是,把它翻译成0,1序列后的书籍厚度最薄,但要做到这一点,是不是仅仅通过考虑前面提到的字母的出现频率进行编码就能实现呢? 答案是否定的. 因为,语言不是掷色子产生的,不是满足多项分布的伯努利试验,如果考虑词的层面,我们会发现,英文中经常会有某些固定搭配,某些词的后面跟随另一些词的频率非常高,比如deal后面紧跟的词是with的可能性很大. 上升到句子的层面也会有类似的规律,比如某些连词的固定搭配,看到if引导的从句后,下一个从句极有可能是由then引导的. 任何类似的“规律”或“模式”都可以为我所用,不管是字母层面、词语层面、短语层面还是句子、篇章层面,都可以帮助压缩编码后的书籍厚度. 所以分析“简单性”必须考虑所有可能的模式,不管它们隐藏的有多深.

从讨论序列预测涉及的第二个问题(归纳能否保证确定性)开始,我们一路追究到了“简单性”概念,追究到了“压缩”与“编码”的问题,而这竟把我们引到了序列预测涉及的第一个问题的核心概念上:以“递推公式”为代表的“规律”或“模式”究竟是什么? 这种所谓的“规律”或“模式”到底有多少?

在回答这个问题之前,不妨让我们先回顾一下前面给出的智商测试题. 前面几题很容易看出来它们的“规律”,无非是某种“递推公式”而已. 比如,下面的第一个数字串,我们都知道它是圆周率的小数位,可你能看出下面第二个数字串的规律吗? 估计很少有人能直接看出来,如果你看了半天仍然看不出来,那么估计你会丧气地感觉这个数字串跟随便抛硬币抛出来的数字串没什么区别. 但现在我告诉你,它其实也是圆周率的小数位,只是这次使用二进制表示而已. 现在你知道答案了,然后动手验证了一下,发现没错,然后你会同意,这这个数字串也是有“规律”的,只是它的“规律”隐藏的更深而已. 但为什么隐藏的这么深的“规律”仍然可以被称作“规律”? 我们会说,因为圆周率可以通过某种“能行”的“递推公式”运算得到,十进制数到二进制数的转换也是一种“能行”的运算,然后通过二者的复合运算即可得到我们想要的数字串.

141592653

00100100001111110110101010001000

因此, 这里我们一直把“规律”或“模式”看作某种“能行”的计算, 正是为了刻画这种“能行”的计算模式, 图灵发明了“图灵机”.

图灵机是个超级简单的计算装置, 但它的计算能力却异常强大, 可以证明, 图灵机可计算的函数类对应了部分递归函数类, 而当今最前沿最高深的理论物理学所用到的函数都不会超出这个类.

论题 1 (图灵论题及各种等价的计算模型).

能行可计算
 \parallel
 递归 = 图灵可计算
 \parallel
 有穷可定义 = *Herbrand-Gödel* 可计算
 \parallel
 任何一个协调的包含 \mathcal{R} 的形式系统可表示
 \parallel
 λ -可定义 = 流程图 (或 ‘while’ 程序) 可计算
 \parallel
 附加一条无穷带的神经网络可计算 = “生命游戏”
 \parallel
Post/Markov/McCarthy/Kolmogorov & Uspensky 可计算 ...

鉴于图灵机强大的威力, 图灵提出了“图灵论题”: 任何“能行可计算”的函数都是“图灵机可计算”的. 这一论题至今没有被推翻. 人们从各种角度做出了各种刻画“能行”可计算的尝试, 试图超越图灵机可计算的概念, 但最后发现这些模型都是彼此等价的. 如果人的“意识”是可计算的话, 这种简单的图灵机也将可以涌现出“意识”现象; 如果现实世界本身都是可计算的话, 那么现实世界的“终极真理”将不过是某个写在这种图灵机上的“程序”.

公理	$\xrightarrow{\text{形式系统}}$	定理
程序	$\xrightarrow{\text{通用图灵机}}$	输出
编码	$\xrightarrow{\text{解码}}$	原始数据
科学理论	$\xrightarrow{\text{推演}}$	经验现象
DNA	$\xrightarrow{\text{演化}}$	有机体
终极理念	$\xrightarrow{\text{上帝}}$	宇宙

表 1: 科学是对经验的理解, 理解就是压缩, 预测可以看作某种解压缩

“可计算”的概念相当稳定, 上面提到的理论 \mathcal{R} 是一个非常弱的形式系统, 比鲁滨逊算术还弱, 而

鲁宾逊算术又远弱于我们常见的皮亚诺算术, 但只要任何比它强的系统, 不管有多强, 最后“可表示”的函数都是一样的, 都是递归函数, 也只有递归函数形式系统“可表示”. 从上面列出的各种等价的定义可以看出, 到目前为止, 从各种不同角度对“能行可计算”概念的把握都聚焦了同一个东西. 这些都极大地强化了我们对于图灵论题的信念, 甚至远远大于对任何主流的物理学理论的信任程度. “人脑”、“世界”可计算的猜想越来越受重视的原因, 很大程度上正是源于图灵论题牢不可破的信念.

如果不限制计算资源的话, 人脑完全可以支持通用计算, 也就是说, 可以模拟任何可能的计算, 如果现实世界确实是可计算的, 那么, 在忽略计算资源限制的情况下, 人脑原则上可以模拟现实世界 (包含人脑自身) 的运行. 人脑甚至可以枚举所有可能的计算 (所有可能世界). 所有“可能”尽在掌握, 是我们尽情删减、挑选的时候了.

5 量化“简单性”——算法复杂性

上帝走捷径.

让一切尽量简单, 但不更简单.

— 爱因斯坦

我们把“规律”或“模式”看作可计算的函数, 贯穿有穷个点的“路径”有 (不可数) 无穷多, 但我们只关注那些可计算的“路径”, 但这种可计算的函数仍有可数无穷多, 这么多的“路径”哪条才是我们想要的呢? 如果我们不知道哪条是我们想要的, 又怎么估计下一个点会落在哪里呢? 虽然我们把穿过有限个点的“路径”从不可数无穷多条减少到了可数无穷多条, 但莱布尼茨的曲线拟合悖论依然困扰着我们.

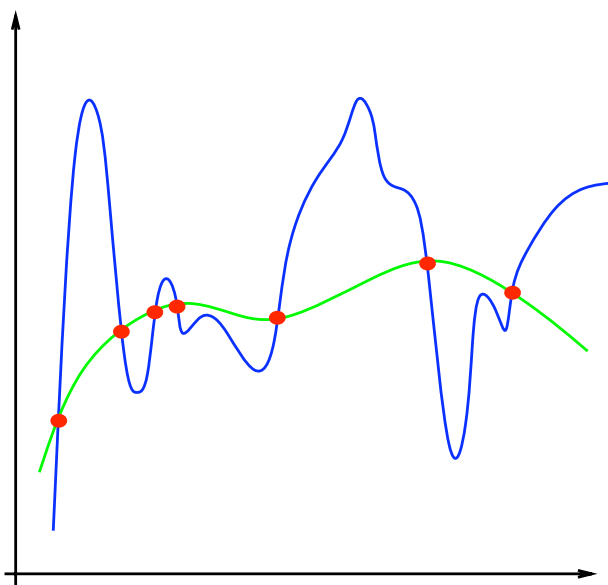


图 2: 莱布尼茨曲线拟合悖论

我们无法选出唯一的一条一往无前地沿着走下去,这就是一直挑战着哲学家们的“同果未必同因”的归纳问题. 莱布尼茨给出的解决方案是借助于“简单性”,虽然我们无法选出唯一的一条,但我们可以有所偏好,越“简单”的路径我们可以赋予越多的“心理预期”(心理习惯),然后根据观测到的新“点”可以不断调整这种“心理预期”.至于如何调整“心理预期”后面再讨论,首先面对的问题是,“简单性”有客观的衡量标准吗? Kolmogorov 通过定义算法复杂性正面回答了这个问题.

前面我们谈到了把“规律”或“模式”等同于图灵可计算的函数,在图灵机上真正计算这个函数的程序就有一个长度,用这个程序的“长度”衡量“简单性”就是一个很直观的想法,但我们不禁要问,即使是通用图灵机也有无穷多种,虽然计算的是同一个函数,但会不会在一个通用图灵机上的程序很短,但换另一个通用图灵机上的程序可能需要很长?不必担心,我们可以证明某种“不变性”! 定义序列 x 的算法复杂性为某通用图灵机 U 上输出它的最短程序的长度.

相对于任何其它通用图灵机 U' ,在某个(依赖于 U' 但不依赖于 x 的)常数界内,通用图灵机 U' 上计算 x 的最短程序的长度近似等于 x 的算法复杂性. 因为,通用图灵机可以模拟任何图灵机(包括各种通用图灵机),比如常见的程序语言 C 或 Java 等都可以看作通用图灵机,你可以在 Java 语言中写一个“翻译程序”,将任何 C 语言的程序自动翻译成 Java 语言的程序,假如 p 是 C 语言中计算 x 的最短程序, p' 是借助“翻译程序”把 p 从 C 语言翻译到 Java 语言的对应程序,所以 p' 在 Java 中计算 x ,而且 p' 不会比 p 长多少,最多相差一个“翻译程序”的长度而已. 反之,从 Java 到 C 也可以有类似的翻译.

因此,在与输入无关的常数界内不依赖于具体哪个通用图灵机的意义上,算法复杂性概念“客观”的刻画了“简单性”概念.

现在有了“简单性”概念,我们忍不住要定义“心理预期”进而解决归纳问题了. 如何定义“心理预期”呢? 分配“心理预期”无非就是分配某种“权重”,而这种“权重”需要与“简单性”或这里的“算法复杂性”成负相关的关系,越简单越偏好. 最自然的想法就是,定义穿过序列 x 的一条“路径”(可计算的函数 f) 的“权重”,可以借助通用图灵机上计算函数 f 的程序 p ,回顾前面我们用“编码”诱导“概率”的方式,我们把 p 看作 x 的“编码”,对于短的“编码”赋予高的“概率”,对于长的“编码”赋予小的“概率”,“编码长度”与“心理预期”负相关. 这样,对于任何对 x 的编码方式 p ,即 p 与 x 一致,或说 p 输出 x ,那么我们对 p 赋予的“偏好”/“信念”大小就是 $2^{-\ell(p)}$. 在没有任何经验的情况下,我们的总“信念”就是所有停机程序的“信念”之和,但很不幸,它是发散的! 这意味着我们赋予信念的方式无法归一化为合适的概率测度,我们从“算法”诱导“概率”/“先验信念”的方式是有问题的,所以不能考虑所有可能的停机程序!

为了解决这个问题,我们不得不放弃一般的图灵机,而必须考虑某种特殊类型的图灵机——“前缀图灵机”,这种图灵机有三条带子,一条单向的输入带,一条单向的输出带,一条双向的工作带,输入带只读,输出带只写,单向带上的读头、写头都只能从左往右移动. 这种前缀图灵机上所有停机的程序构成前缀码: 没有一个程序是另一个程序的前缀. 这是一种“自定界”的程序——只读头读完输入带上的程序就知道这个程序结束了、而不必担心有另一个更长的程序是它的延伸. 事实上,在计算同样的函数的意义上,任何普通的图灵机程序都可以等价地改写为这种“自定界”的程序,因此可以说,虽然我们限制了图灵机的类型,但事实上我们并没有真的丢失任何“假设”.

什么是“单调图灵机”呢？“单调图灵机”与“前缀图灵机”的“硬件”完全一样，唯一的区别是，对于“前缀图灵机”我们只考虑那些停机的程序，所有停机的程序构成“前缀码”，而“单调图灵机”不必停机，可以无限运行下去。但对于任何给定的输出 x ，输出 x 的所有单调图灵机程序构成“前缀码”。

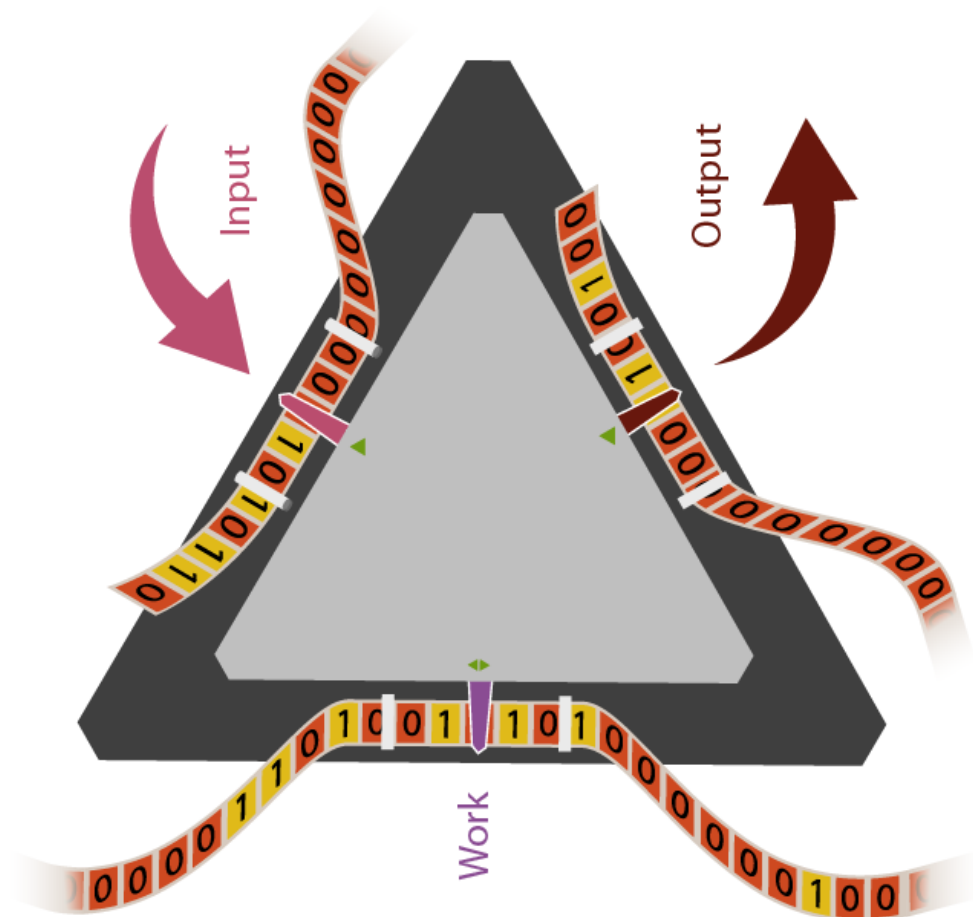


图 3: 前缀/单调图灵机

针对这类前缀/单调图灵机，可以构建通用前缀/单调图灵机，然后可以定义字符串 x 的算法复杂性 $K(x)$ 为通用单调图灵机输出 x 的最短程序的长度²。

²有意思的是，可以证明，对于任何可计算的概率分布，算法复杂性的期望近似等于 Shannon 熵。

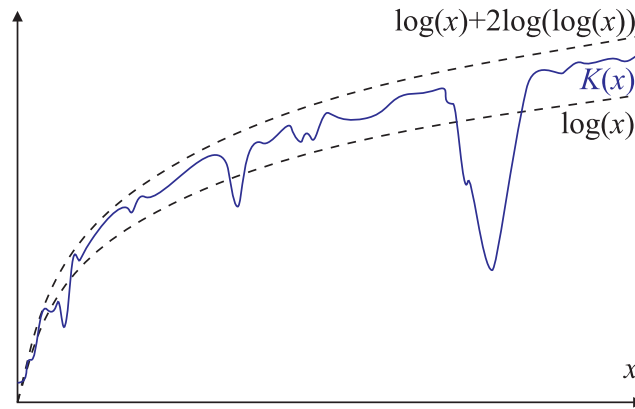


图 4: 算法复杂性

依照前面的方式, 现在可以定义对前缀图灵机程序 p 的“权重”(信念/偏好) 为 $2^{-\ell(p)}$, 而且可以证明, 所有停机的前缀图灵机程序的信念之和是收敛的³, 所以可以归一化为合适的概率测度.

哲学家伊壁鸠鲁认为, 不仅要考虑与经验一致的最简单的假设, 所有不违背经验的假设都要保留.

那么, 下面考虑如下一个问题: 如果在单调通用图灵机的输入带上随机的抛掷一枚质地均匀的硬币, 抛出正面写 1, 抛出反面写 0, 那么输出带上会输出序列 x 的概率为多大?

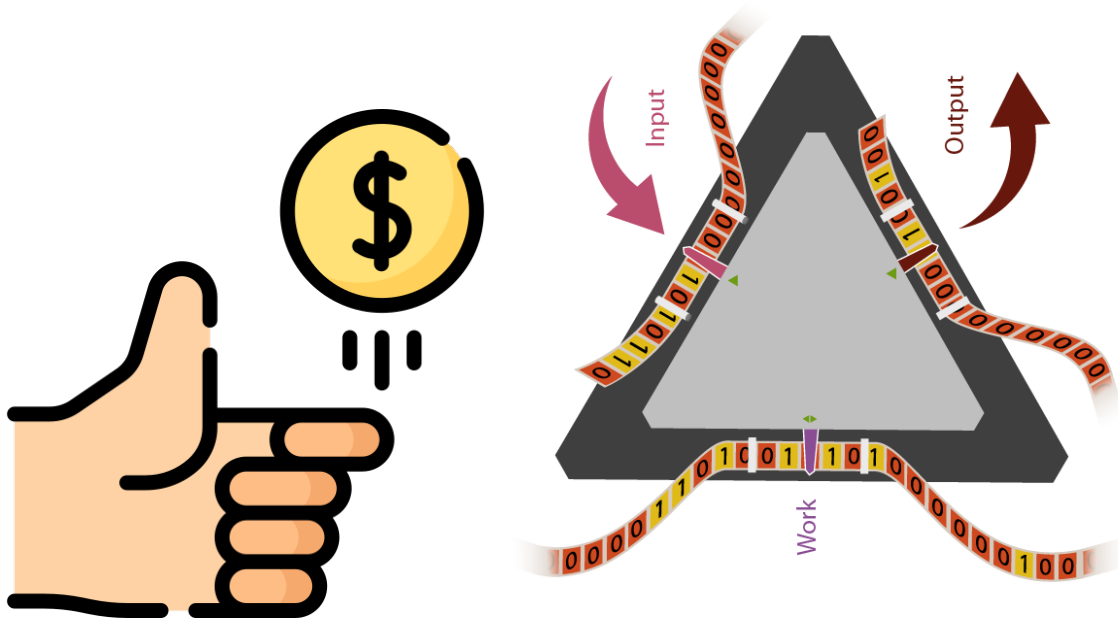


图 5: 上帝抛硬币, 抛在图灵机上.

³而且, 这就是著名的 Chaitin 常数, 它是算法随机的, 借助它可以证明某种版本的不完全性定理: 任何包含初等算数的可递归公理化的协调的形式系统都只能判定 Ω 的有限位数字!

不难发现, 答案为

$$M(x) := \sum_{p:U(p)=x^*} 2^{-\ell(p)} \quad (5.1)$$

这就是 Solomonoff 定义的“算法概率”. 其中 $U(p) = x^*$ 表示 p 输出 x 后未必停机, 可以继续输出下去. 这也是奥卡姆与伊壁鸠鲁的折中, 伊壁鸠鲁要求保留所有与经验一致的假设, 奥卡姆独钟最简单的假设, 而算法概率既保留了所有与经验一致的假设, 又对简单的假设赋予了更高的偏好, 同时兼顾了二者.

设想一下, 如果这枚质地均匀的硬币不是抛在图灵机的输入带上, 而是直接抛出什么就作为结果输出什么, 那么历史 x 的概率将是 $2^{-\ell(x)}$, 下一刻历史的延伸是 0 还是 1 的条件概率将是 $\frac{1}{2}$, 也就是说, 如果我们的世界是以一种完全随机的方式创世的, 那么我们还有任何办法进行某种可靠的预测吗? 那该是多么令人沮丧的世界啊!

我们允许上帝掷色子, 但必须掷在图灵机上! 上帝如此至真至善, 表面上道貌岸然看似无私 (“随机”), 事实上以更高的可能性输送更简单的模式供我们归纳学习. 换句话说, 奥卡姆剃刀是种信念, 是对“简单性”的信念, 是对上帝编码的最优性的信念.

6 如何“学习”? — “信念修正”与“通用先验”

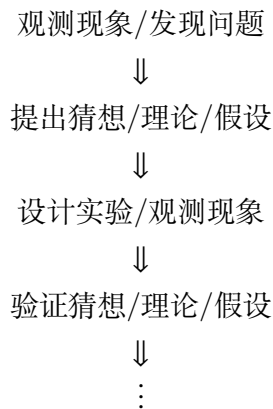
追求简单性, 但要质疑它.

— 怀特海

“任何‘可能’都要求‘存在’. 任何‘可能’都将‘存在’ — 除非它被其它同样要求‘存在’但与之不相容的‘可能’所阻碍.”

— 莱布尼茨

有了算法概率, 如何进行归纳学习呢? 回顾前面讨论的科学发现的过程,



这个过程可以如下解读: 虽然每次只有一个假设被提出、被检验, 但仍可认为所有可能的假设构成的假设空间是固定的, 记之为 \mathcal{M}_T , 然后按照简单性从小到大的原则为所有可能的假设 \mathcal{M}_T 排一个序,

其中每个假设都可以看作现实世界物理状态演化过程的可能的解释 — 只要物理世界的演化过程是可计算的, 它就可以被某个程序模拟或解释 — 每次只有排在最前面的假设可以被看到, 然后依据观测到的物理状态挨个验证这些假设, 到目前为止与经验相协调未被证伪的所有假设 \mathcal{M}_T 中最简单的那个就暂时当做临时真理, 直到它被以后的经验证伪为止, 一旦证伪就迅速选择下一个最简单的与观察经验一致的假设当临时真理. 我们期望某一天某个假设会稳定下来永远不会被证伪 — 也就是说, \mathcal{M}_T^* 永远不会再变小, \mathcal{M}_T^* 中排在第一位的也就是最简单那个假设所代表的计算过程恰好对应了现实世界物理状态的演化过程. 换句话说, 这个假设是最终的真理.

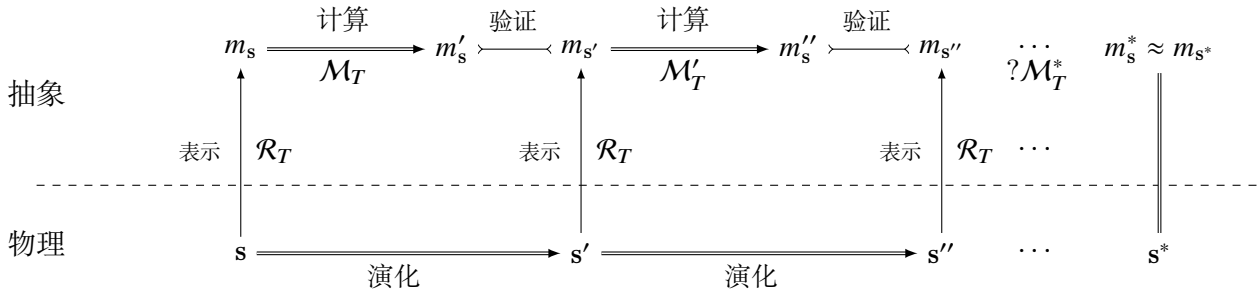
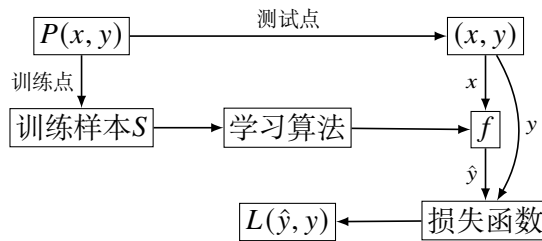
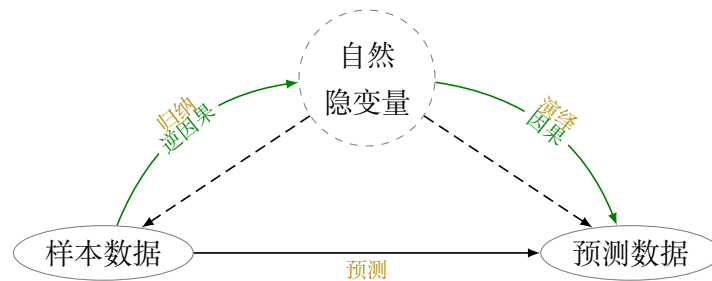


图 6: 抽象世界与物理世界

这种提出猜想比对经验的过程类似一个根据样本归纳模型的监督学习过程. 开始时会根据问题和观测 (训练样本) 首先提出最直观、最简单的猜想 (学习算法) 尝试解释现象, 如果这种猜想被新的观测否决的话 (损失过大), 再提出新的猜想, 然后依次循环往复、不断地进行下去.



真实的自然作为一个隐变量, 我们无法直接观察, 我们只能通过输入和输出去猜测逼近真实的自然.



设想一下, 这些猜想不是一个一个排着队逐一起来等待检验, 不仅简单的猜想, 而是一开始所有可能的猜想都以某种方式一下子产生了出来一同被摆在前台, 这时, 就需要认为的给它们排队, 对他们有所偏好. 我们所有的“偏好”或“信念”都是以前经验的体现. 要想协调的更新“信念”、合理的调整“心理预期”, 首先要考虑各种可能性, 预想与目前的经验相一致的各种可能的未来历史延伸. 也可以

说是与历史经验一致的各种可能世界 (可能环境/猜想), 未来的历史将是现在对我们敞开的所有可能世界之一. 如果把每个可能世界看作一系列状态的时间序列, 可以为一系列状态持续到某个时间点的未来历史赋予概率 (信念), 它可以是与那段历史相一致的可能世界占所有可能世界的比重. 随着观测的不断深入, 每观测到一新的状态意味着一些与它不一致的可能世界被排除, 存活下来的可能世界则依照贝叶斯公式重新分配“后验信念”. 但在没有任何经验时、在没有进行任何贝叶斯更新之前 — 还必须有一个“先验信念”. 根据奥卡姆剃刀, 对现象的更短压缩理应享有更高的“心理偏好”. 所以, 如果把单调图灵机上的一个程序 p 跟由它确定的可能历史 $U(p)$ 等同看待, 直接把程序当作一个可能世界的话, 按照“简单性”对它们排队, 也就是按照“简单性”对它们进行“偏好”, 开始时用 $w_p := 2^{-\ell(p)}$ 作为对可能世界 p 的“先验信念”/“偏好”, 也就是按照 $2^{-\ell(p)}$ 从大到小的顺序为所有的程序 \mathcal{M}_T 排队. 这时, 算法概率 $M(x)$ 可以看作对未来历史 x 可能被产生的“主观信念”.

前面一直在讨论的都是确定性的“世界”或“环境”, 而如果“环境”不确定怎么办?

我们将“环境”、“世界”、“模型”、“猜想”、“假设”等概念等同视之, 当它们是确定性的时把它看作某种程序或程序决定的输出, 当讨论不确定性的“环境”时把它看作某种概率分布. 确定的可能世界 p 对应唯一的可能历史 $U(p)$, 不确定的可能世界 (概率测度) ν 对应很多可能历史 $\{x : \nu(x) > 0\}$. 确定的可能世界 $U(p) = x$ 可以看作不确定的可能世界的特例 $\nu_p(x) = 1$.

即使对于不确定的世界, 它也必须具有某种“规律”或“模式”. 虽然它的下一个状态不能完全由其先前历史完全唯一的决定, 它也必须具备某种“可计算”的性质. 准确地说, 是“下半可计算的半测度”. 记这种不确定的世界类为 \mathcal{M}_U . 它是一个递归可枚举集, 除了下一刻的状态不由其历史唯一决定外, \mathcal{M}_U 与前面确定性的世界类 \mathcal{M}_T 很相似.

对于任何序列 x , 对于任何不确定的可能“世界”(不确定的“路径”或“环境”) $\nu \in \mathcal{M}_U$, 只要 $\nu(x) > 0$ 就认为这种可能世界与历史 x 一致. 计算 ν 的最短程序可以看作 ν 的理想中的“最优编码”. 这种“最优编码”的长度即是它的 Kolmogorov 复杂性 $K(\nu)$. 然后, 依照前面借助“理想编码方式”诱导“概率”的方法, 可以定义对于不确定性环境的“先验信念” — Solomonoff 先验

$$w_\nu := 2^{-K(\nu)}$$

也就是说, 类似假设空间 \mathcal{M}_T 的情形, \mathcal{M}_T 开始时是以简单性 $2^{-\ell(p)}$ 从大到小的顺序排队的, 而 \mathcal{M}_U 里所有的假设开始时是按照简单性以 $2^{-K(\nu)}$ 从大到小的方式排序的.

然后, 类似算法概率 $M(x)$ 的定义, 综合奥卡姆和伊壁鸠鲁的哲学, 对于任何可能的未来历史 x , 它可能被产生出来的主观信念就是 Solomonoff 通用贝叶斯混合:

$$\xi_U(x) := \sum_{\nu \in \mathcal{M}_U} w_\nu \nu(x)$$

Solomonoff 算法概率 $M(x)$ 可以看作以 $2^{-\ell(p)}$ 为通用先验的通用贝叶斯混合.

$$M(x) = \sum_p 2^{-\ell(p)} \mathbb{I}[U(p) = x^*]$$

已经经历历史 x 后对于“环境”或“世界” v 的后验信念可以由先验信念用贝叶斯更新,

$$w_x^v := \frac{w_v v(x)}{\sum_{v \in \mathcal{M}_U} w_v v(x)} = \frac{w_v v(x)}{\xi_U(x)}$$

科学方法过程好比假设空间 \mathcal{M}_T 排好队后一个一个被检验, $\mathcal{M}_T \xrightarrow{s} \mathcal{M}_T'$ 是一个逐步变小的过程, 与经验 s 不一致的假设全部被剔除出去, 但变小后的假设空间里各假设的排队次序不变, 仍按照最初规定的简单性标准排列. 而通用归纳的过程好比整个假设空间同时接受检验, 这意味着, 已经经历状态 x 后, 假设空间 \mathcal{M}_U' 跟以前 \mathcal{M}_U 比不仅可能缩小了, $\mathcal{M}_U \xrightarrow{x} \mathcal{M}_U'$ 不仅某些假设被经验证伪了, 而且, \mathcal{M}_U' 里的所有假设的排序方式也可能变了, 不是按最初的简单性标准 w_v 排序了, 而变为以 w_x^v 的方式重新排序.

已经经历历史 x 后对于任何可能的未来历史 x' 的主观信念就是

$$\sum_{v \in \mathcal{M}_U} w_x^v v(x' | x) = \frac{\xi_U(xx')}{\xi_U(x)} =: \xi_U(x' | x)$$

这也就是“主观信念”的条件概率. 也就是说, 仅仅通过一个简单的 ξ_U , 前面几段所说的 — 对假设空间开始时按照“简单性”排队、有了经验后对其进行删减、然后更新信念、根据更新后的信念对假设空间重新排队的过程都可以体现.

有意思的是, 可以证明, 对于确定性的“可能世界”类 (单调图灵机上所有可能的程序 \mathcal{M}_T) 诱导出的“主观信念” M , 与对于不确定性的“可能世界”类 (下半可计算的半测度 \mathcal{M}_U) 诱导出的“主观信念” ξ_U 近似相等 (Hutter[Hut05]).

$$M(x) \cong \xi_U(x) \quad (6.1)$$

这意味着, 不管生活其中的现实世界是确定性的可能世界类中的一员还是不确定性可能世界类中的一员 (前一类是后一类的特例), 都假装生活在确定性的可能世界里、对于可能的未来历史 x 抱守相同的主观信念 $M(x)$ 是同样合理的. 但以此进行归纳预测的效果究竟如何呢?

7 如何预测? — 通用归纳

虽然不可计算, 但“算法概率”可以作为归纳系统的“黄金标准”.

— Ray Solomonoff

Solomonoff[Sol78] 证明了算法概率的通用性.⁴

定理 7.1 (Solomonoff 完全性定理 [Sol78]).

$$\sum_{t=1}^n \sum_{x_{1:t} \in \mathcal{X}^t} \mu(x_{<t}) (M(x_t | x_{<t}) - \mu(x_t | x_{<t}))^2 \stackrel{+}{\leq} K(\mu) \ln 2$$

⁴这里的“完全性”是指算法概率可以收敛到任意可计算的概率测度, 有时也称作“收敛性”, 它与逻辑学中的“完全性”无关.

对于确定性的环境, 即, 世界的后续状态由其先前历史完全唯一地决定, 通过有限的错误, 算法概率能够准确地预测未来. 即使对于不确定性的环境, 只要它仍然是“有规律的”, 或可以看作某种可计算的测度, 那么, 仍然仅需有限几个错误, 算法概率就能帮我们准确的逼近真实的概率分布. 而且, 这种逼近的快慢取决于真实环境的复杂性, 越简单的环境越容易估计, 越复杂的环境则需要更多的错误才能逼近, 但不管怎么样, 只要环境是“有规律的”, 总可以用有限个错误去逼近. 只要现实世界是“有规律”的, 虽然不知道它的复杂性有多高, 也不知道什么时候才能真正精确地进行预测, 但只要要坚持用算法概率估测未来, 肯定在某个时候, 在经历了足够多的试错之后, 就可以逼近现实世界背后真实的运行机制.

遗憾的是, 算法概率是不可计算的. 如果试图保证对任何“有规律”的环境都可以逼近, 也就是说, 如果试图保证“通用性”的话, 这种不可计算的特性是不可避免的. 因为, 如果用某种可计算的测度去估计真实分布的话, 总可以针对这种可计算的测度构造一个确定的序列, 使得用这个可计算的测度预测这个序列的时候, 其表现无比糟糕. 比如, 如果预测用的是一个可计算的测度, 它预测下一位最不可能发生的是什么呢就直接构造这个序列的下一位是什么.

回到前面提到的几个数字推理题, 这些都可看作某种序列预测题, 所以都可以用通用归纳的方式来求解, 因为, 在理论上, 算法概率给出了这些问题的统一解. 其中前四个问题比较简单, 后两个看上去有些奇怪, 第 (v) 题中的一个数字并不真的是“一个数字”, 而需要拆解成两个数字来找寻模式, 最后一题根本就不能当作数字看, 而是要利用它们的图形信息, 这样还能看作数字序列的预测问题吗? 虽然利用的不是数字本身的数值大小, 而是其高层的图形信息, 但直观上, 仍然必须承认, 整个求解过程是一个“能行可计算”的过程. 这意味着, 可以借助图灵论题, 对它们进行重新编码, 编码为另一种形式的数字序列, 编码后的序列是数值层面可计算的. 也就是说, 如果真的要利用算法概率 M 解决问题, 那么所有问题都要事先编码为图灵机 U 的语言才行. 只要现实世界是可计算的, 现实世界中的问题都可以翻译为 U 的语言.

但仍有一个问题, 因为图灵停机问题不可解, 这里的算法概率是不可计算的, 不仅如此, 如果试图保证对任何“有规律”的环境都可以逼近, 也就是说, 如果试图保证“通用性”的话, 这种不可计算的特性是不可避免的. 因为, 如果我们用某种可计算的测度去估计真实分布的话, 总可以针对这种可计算的测度构造一个确定的序列, 使得用这个可计算的测度预测这个序列的时候, 其表现无比糟糕. 比如说, 如果我们预测用的是一个可计算的测度, 它预测下一位最不可能发生的是什么呢我们就构造这个序列的下一位是什么.

总之, Solomonoff 把归纳问题转化为序列预测问题来解决, 为了解决序列预测问题 Solomonoff 把“简单性”与“概率”联系起来, 通过概率可以进行贝叶斯学习, 而要量化“简单性”首先要界定什么是“模式”, 图灵通过图灵机概念清楚地界定了所有可能的“模式”, Kolmogorov 在此基础上通过 Kolmogorov 复杂性给出了“简单性”的客观定义, 通过整合奥卡姆、伊壁鸠鲁、贝叶斯、图灵、Kolmogorov 等人的成果, Solomonoff 成功定义了算法概率并用它解决了序列预测问题.

通用智能的核心 — 通用归纳的讲解到此结束, 接下来, 我们只需要在此基础上添加一个序贯决策过程即可创造出通用智能主体 AIXI 了.

8 主体与环境的交互 — 通用智能

智能是主体在各种各样的纷繁复杂的环境中实现目标的能力。

— Shane Legg and Marcus Hutter

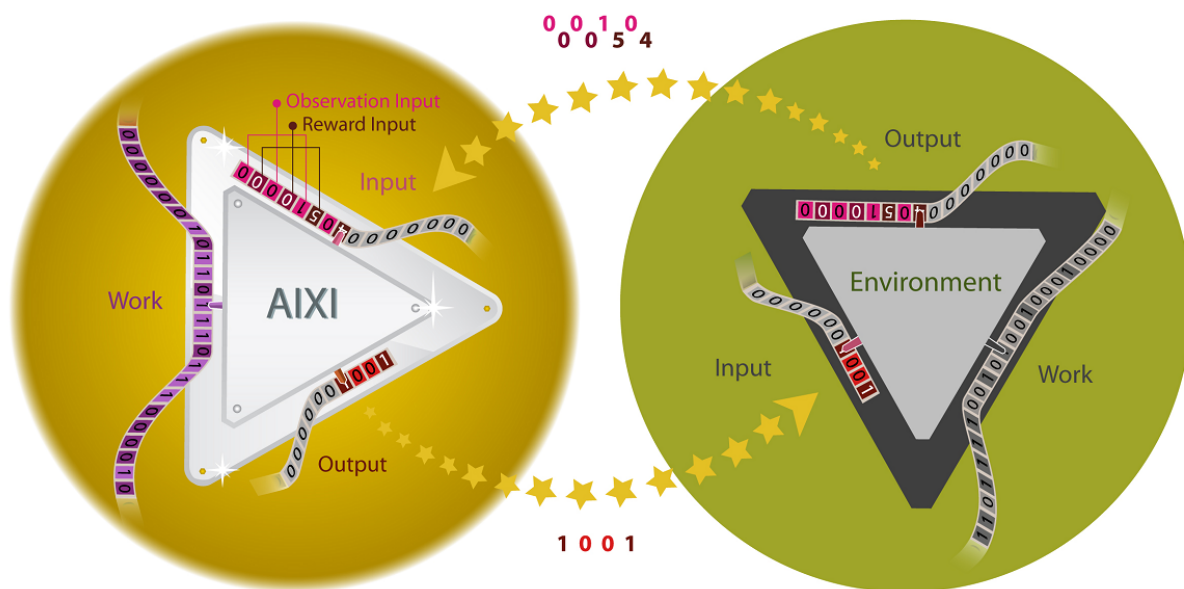


图 7: AIXI 与“环境”的较量

Solomonoff 通用归纳模型事实上是一个预测模型, 它不考虑主体对环境的改变以及环境对主体的反馈. 而在现实环境中, 主体需要与环境不断交互. 如果“主体”和“环境”都是确定性的, 那么二者的交互可以看作两个程序 (p 和 q) 的交互, 其中一个的输出是另一个的输入, 一个的输入是另一个的输出.

如图8所示, p 的输入带是 q 的输出带, p 的输出带是 q 的输入带. 在第 k 个回合, 主体 p 输出 (做动作) a_k , 环境 q 读取 a_k , 然后输出 o_k , 伴随着 o_k 还反馈给主体 p 某种“奖励” r_k , 主体 p 读取 (感知到) o_k 和 r_k , 然后进行下一个 (第 $k+1$ 个) 回合 ...

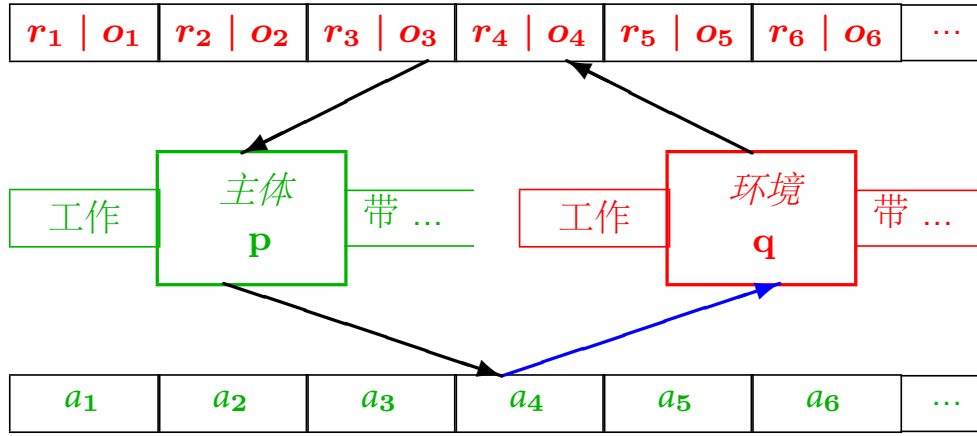


图 8: “主体”与“环境”的交互

前面处理归纳问题时, 我们用算法概率来估测可能的未来历史, 这里也一样, 需要评估主体与所有可能的环境 q 交互出的所有可能的历史, 对于简单的环境赋予高的“偏好”, 对于复杂的环境赋予低的“偏好”, 然后用

$$\sum_{q: U(q, a_1 \dots a_m) = o_1 r_1 \dots o_m r_m} 2^{-\ell(q)}$$

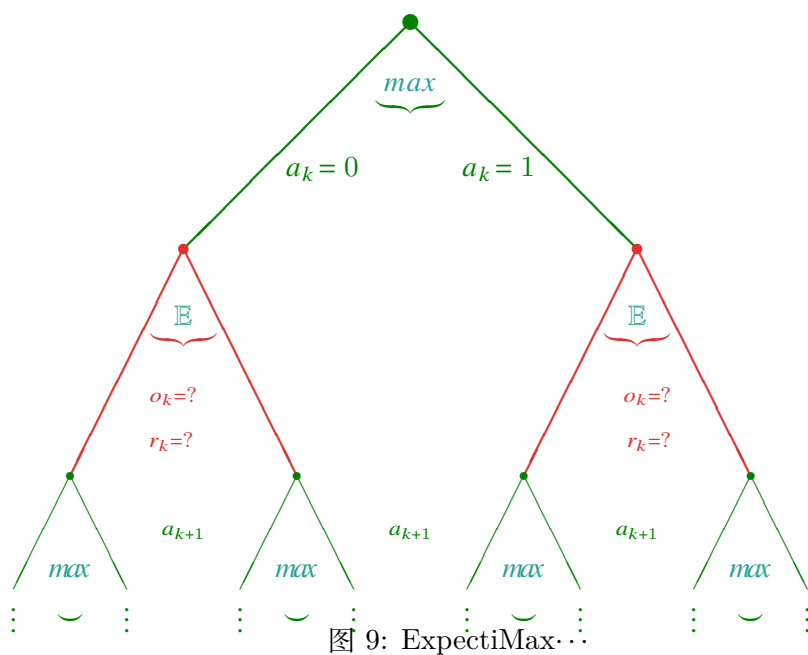
来评估可能的历史 $a_1 o_1 r_1 \dots a_m o_m r_m$ (假设主体的寿命长度为 m).

AIXI 就是在这种不确定的环境中用上面的概率最大化未来的期望效用的策略. 也就是说, 在主体与环境交互的第 k 个回合, AIXI 的行为方式是:

$$a_k := \operatorname{argmax}_{a_k} \sum_{o_k r_k} \dots \max_{a_m} \sum_{o_m r_m} [r_k + \dots + r_m] \sum_{q: U(q, a_1 \dots a_m) = o_1 r_1 \dots o_m r_m} 2^{-\ell(q)} \quad (\text{AIXI})$$

Hutter 只用上面一个公式就定义了能在不同环境中实现目标的“超级智能体”.

也就是说, 如果环境是确定性的 q , 那么未来就是可计算的 $\dots o_k r_k \dots o_m r_m = U(q, a_1 \dots a_m)$, 未来的累积奖励就是 $r_k + \dots + r_m$. 但因为真实的环境是未知的, 所以 AIXI 要最大化期望累积奖励, 这就要预先考虑自己所有可能的行为方式与所有可能的环境交互生成的所有可能的未来历史, 对简单 (复杂) 的环境 q 赋予高 (低) 的权重 $2^{-\ell(q)}$. 因为主体与环境的交互是时序交错进行的, 所以期望 (Σ) 与最大化 (\max) 也要交错进行.



决策论	=	概率	+	效用
+			+	
通用归纳	=	奥卡姆	+	贝叶斯
				图灵
通用强化学习模型	=	AIXI		

图 10: Hutter 的通用强化学习模型 AIXI 可以看作最大化期望效用的序贯决策过程与 Solomonoff 序列预测的通用归纳模型的结合, 是一种以“算法概率”寻求期望效用最大化的决策.

AIXI 能处理什么问题? AIXI 究竟能处理什么具体问题呢? 事实上, 几乎所有的 AI 问题都可以被处理. 下面给出几个例子.

一、序列预测. 显然, Solomonoff 的序列预测可以看作 AIXI 的特例, 所以 AIXI 拥有序列预测的功能, 像股票走势、天气预报、彩票投注之类的问题都可以转化为类似的序列预测问题, 只要现实世界中这些问题真的是可计算的, 那么 AIXI 都可以成功预测.

二、最优化. 比如寻找某个函数的最小值问题, 这时 AIXI 可以权衡计算该函数的所有程序, 然后只要把 AIXI 的效用函数设为跟自变量的函数值相关的某个函数就好了, 函数值越小则效用越高, 为了寻求最大效用, AIXI 会自动寻找函数的最小值. 计算经过所有城市路途最短的旅行商问题、求解生产某产品的最小成本问题等都属于此类问题.

三、策略博弈. AIXI 还可以进行各种策略博弈, 只需要把博弈的另一方看作“环境”就好了, 比如

象棋之类的两人零和游戏, 而且, 如果对手是“理性人”的话, 这时 AIXI 的“期望最大化”策略会收敛到通常的“极小极大化”策略. 传统的博弈理论只能处理理性主体间的博弈, 如果博弈的另一方有一些非理性的行为, 只要这些行为仍然具备某种“模式”, 那么 AIXI 都可以探索出来, 然后加以开发, 从而谋利. 也就是说, AIXI 可以对抗有限理性或非理性玩家.

四、监督学习. 给定一系列 $(z, f(z))$, AIXI 可以轻易地预测 $(z', ?)$, 所以, 监督学习也很容易处理. 比如识别物体的属性、根据属性分类的问题都可以划归为这种样式, 比较复杂一点儿的, 给定一些状态, 然后教它在合适的状态做合适的动作, 这意味着给定一个 $(state, action)$ 序列, 然后 AIXI 就可以学会以后遇到什么状态该采取什么行动了.

诸如此类的几乎所有问题都可以划归为 AIXI 能处理的问题, 与智能相关的各种要素也都应该可以从其中涌现出来. 这里最关键的还是对环境的压缩问题, 而压缩包含着对任何模式的探索, 而一般来说具体的问题往往只是针对某些或某类具体的模式.

智能的定义与测量 从图灵提出图灵测试开始, 人工智能专家们提出了各种对智能的定义或测试方式, 莱格 (Legg)[Leg08] 对此进行了细致的调查分析, 然后与 Hutter 一起提出了他们对智能的理解——智能就是在各种各样纷繁复杂的环境中实现目标的能力.

一个主体 π 在与环境 μ 的交互过程中可以获得的期望效用记为 V_μ^π , 依照对各种环境的 Solomonoff 通用先验, 可以定义主体 π 适应各种环境能力的“智能”为

$$\Upsilon(\pi) := \sum_{\mu \in \mathcal{M}_U} 2^{-K(\mu)} V_\mu^\pi$$

以前对“智能”的定义和测试都是非形式化的, Υ 是第一个对“智能”形式化的数学定义. 根据 AIXI 的定义和智能 Υ 的定义, AIXI 可以获得最高的智能 $AIXI \in \arg\max_\pi \Upsilon(\pi)$, AIXI 是一个超级智能体.

Hutter 认为, 正如“算法概率”可以看作各种归纳系统的“黄金标准”, AIXI 可以看作各种智能系统的“黄金标准”. 其它所有智能体的智能都应该是对最高智能 $\Upsilon(AIXI)$ 的逼近.

★ = yes, · = no, ● = debatable, ? = unknown. Intelligence Test	Valid	Informative	Wide Range	General	Dynamic	Unbiased	Fundamental	Formal	Objective	Fully Defined	Universal	Practical	Test vs. Def.
Turing Test	●	·	·	·	●	·	·	·	·	●	·	●	T
Total Turing Test	●	·	·	·	●	·	·	·	·	●	·	·	T
Inverted Turing Test	●	●	·	·	●	·	·	·	·	●	·	●	T
Toddler Turing Test	●	·	·	·	●	·	·	·	·	·	·	●	T
Linguistic Complexity	●	★	●	·	·	·	·	●	●	·	●	●	T
Text Compression Test	●	★	★	●	·	●	●	★	★	★	●	★	T
Turing Ratio	●	★	★	★	?	?	?	?	?	·	?	?	T/D
Psychometric AI	★	★	●	★	?	●	·	●	●	●	·	●	T/D
Smith's Test	●	★	★	●	·	?	★	★	★	·	?	●	T/D
C-Test	●	★	★	●	·	★	★	★	★	★	★	★	T/D
Universal $\Upsilon(\pi)$, AIXI	★	★	★	★	★	★	★	★	★	★	★	·	D

图 11: 智能的定义与测量

AIXI 作为一个定义是清晰的, 但它不具有实用性, 并不能真的运用它测量智能. 而且, 在依赖通用图灵机的意义上, 它也不是客观有效的. 事实上, 用来预测的通用贝叶斯混合和衡量智能的 Ex-interim 期望效用都依赖于通用先验, 二者如果不是基于同样的先验就会面临“价廉未必物美”的问题. 即使采用 Solomonoff 先验也一样. 如果估测环境的“主观信念”所依赖的通用图灵机与衡量智能所依赖的通用图灵机不是一个, 也就是说, 如果定义“算法概率”时的“通用先验”所依赖的通用图灵机与量化“智能”衡量标准时的“通用先验”所依赖的通用图灵机不是同一个, 那么 AIXI 对智能的追求可能一开始就在一个错误的方向上, “简单性”未必带来“有用性”.

9 通用智能主体 AIXI 的逼近与变种

所有模型都是错的, 但有些是有用的.

— George E. P. Box

AIXI 的逼近 Hutter 的通用强化学习模型 AIXI 不可计算, 但这并不意味着它毫无用处. 有一种限制可能的环境类通过蒙特卡洛方法和上下文树加权方法而对 AIXI 作出的逼近 MC-AIXI-CTW[Ven+11] 具有较好的实际效果, 它可以在事先不知道游戏规则的情况下, 通过试错法玩好 Cheese Maze, Tic-Tac-Toe, Pacman, Kuhn Poker 等各种稍微复杂的游戏.

究竟对不可计算的 AIXI 的逼近有没有实用价值呢? 首先来看一种对 AIXI 的粗暴的逼近 — 对图9中的期望最大化树进行暴力的截枝, 只往前看固定的几步, 然后仅考虑只依赖于过去几步记忆的马

尔科夫环境, 然后在不告知游戏规则的情况下让这样处理后的 AIXI 重复玩囚徒困境、猎鹿博弈、猜硬币博弈、性别大战博弈、胆小鬼博弈等简单的游戏. 发现它仍能较好的预测对手的策略, 获得令人满意的效果.

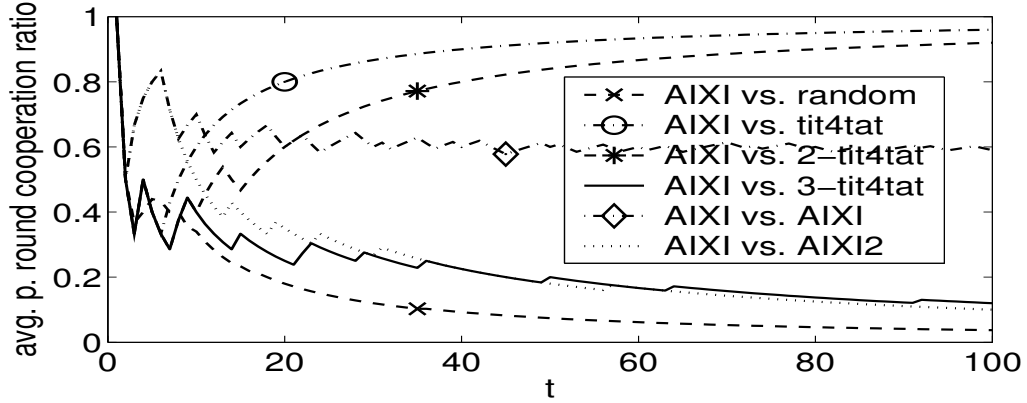


图 12: AIXI 重复囚徒博弈: 如果对手随机选择背叛或合作, AIXI 可以很快发现对手的策略从而选择持续背叛, 如果对手采用以牙还牙策略, 则 AIXI 很快会倾向于合作.

有一种限制可能的环境类通过蒙特卡洛方法和上下文树加权方法而作出的逼近 MC-AIXI-CTW[Ven+11] 具有更好的效果, 它可以在事先不知道游戏规则的情况下, 通过试错法玩好 Cheese Maze, TicTacToe, Pacman, Kuhn Poker 等各种稍微复杂的游戏.

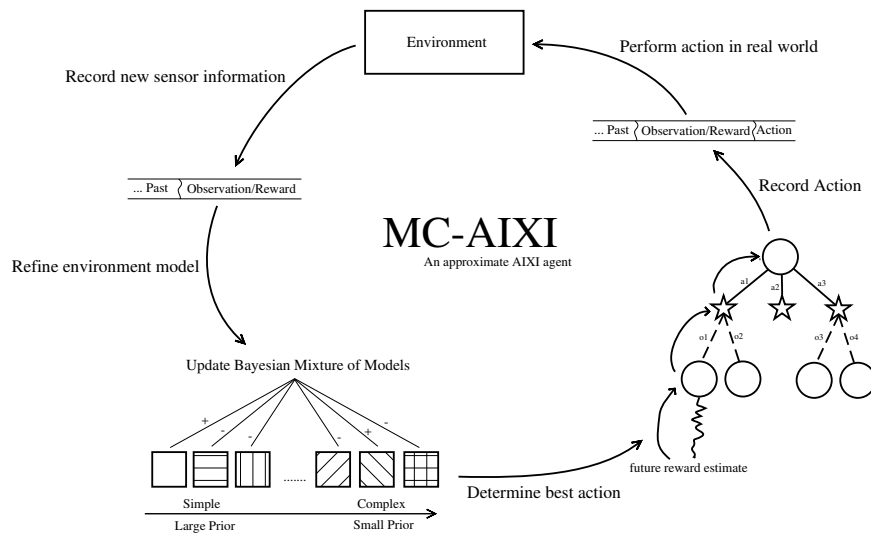


图 13: MC-AIXI-CTW

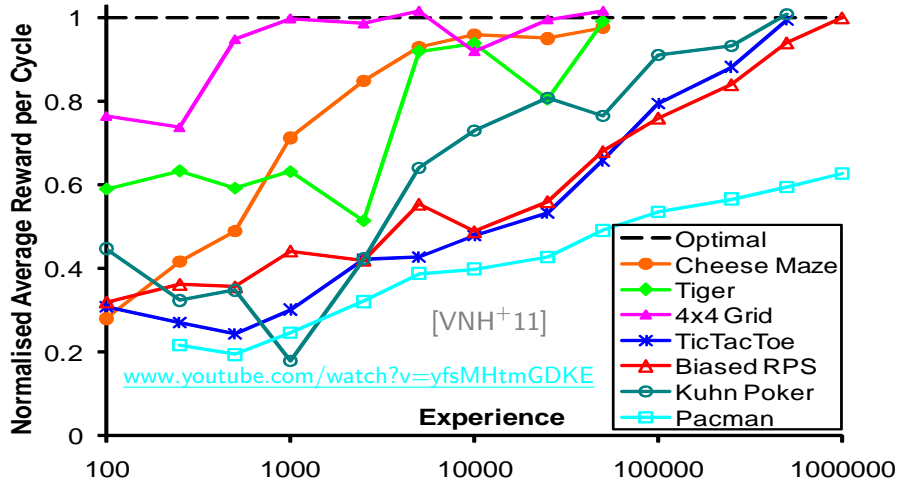
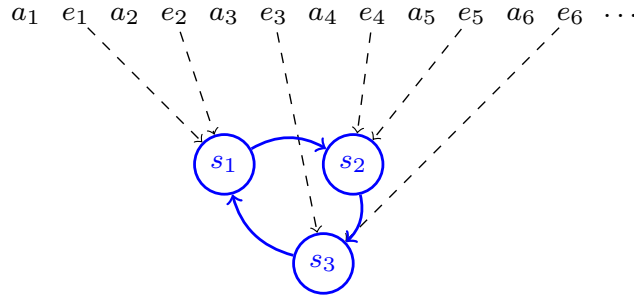


图 14: 在事先不知道具体领域知识的情况下, 同一个主体可以自动适应各种环境

AIXI 的每一步决策都依赖于其全部过去历史, 现实世界虽然复杂但也没有那么复杂. 其中存在很多相对独立的模式, 只依赖它们就可以进行很好的预测. 因此, Hutter[Hut09] 提出了“特征强化学习”的逼近方法, 通过一种类似“极小描述长度原则”的思想, 可以将主体的“历史” h 自动映射到合适的“状态” s 上. 类似传统物理学把物体轨迹的预测归约到位置和速度两个变量, 而速度又可以由相间的两个位置给出, 所以, 物体的轨迹预测是二阶马尔科夫过程.



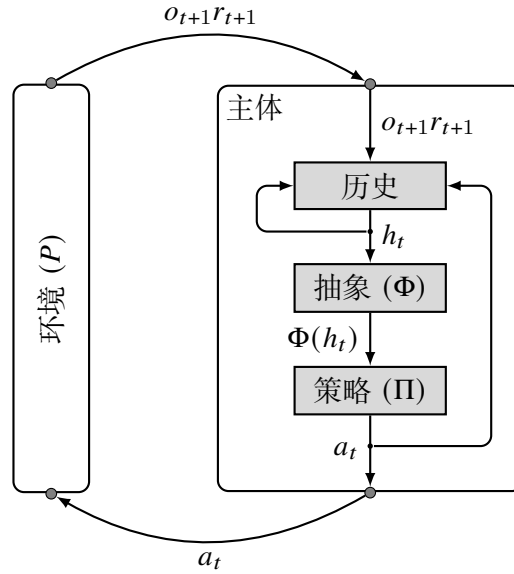
$$\Phi : h \mapsto s$$

$$\Phi^{\text{best}} := \underset{\Phi}{\operatorname{argmin}} \operatorname{Cost}(\Phi | h)$$

$$\operatorname{Cost}(\Phi | h) := \operatorname{cl}(s_{1:n}^{\Phi} | a_{1:n}) + \operatorname{cl}(r_{1:n} | s_{1:n}^{\Phi}, a_{1:n}) + \operatorname{cl}(\Phi)$$

其中 cl 是编码长度函数.

关键问题是, 对任意给定的历史, 是否能够高效地归约到合适的状态? 只要这一步解决了, 就可以将难处理的逼近问题划归到相对简单的马尔科夫决策过程上. 如果要处理的现实问题是简单的, 那么通过这种方法就能自动找到一个简单的马尔科夫决策过程去刻画它. Hutter 等人曾在一些简单游戏上通过特征强化学习取得了不逊于 MC-AIXI-CTW 的实验效果.



虽然 AIXI 也可以有一些实用是逼近, 但它的效果远不及各种深度强化学习模型 [Mni+15], 所以, AIXI 的作用并不在实用方面, 而在于它提供了一个通用强化学习乃至通用人工智能的描述模型. 通过它, 我们可以形式化地研究通用强化学习的原理和机制, 也可以帮助我们更精确的理解“智能”本身.

AIXI 的变种 在 AIXI 的框架中, 主体和环境可以看作两个完全独立的“主体”在交互, 这是一种为了易于处理问题进行的简化, 现实世界往往更加复杂, 主体并不具有游离环境之外的超越地位, 主体也是环境的一部分, 主体就在环境中, 主体的计算资源受到环境的时空限制, 为了刻画这些复杂的情形, AIXI 的各种变种应运而生.

首先, 根据效用函数和贴现函数的不同, AIXI 可以有几种不同类型的变种. 然后根据能否读取和修改自身源代码、内存、以及环境能否读取和修改主体的源代码等又可以定义几种不同的变种, 这主要是 Orseau[RO11; OLH13; OR12] 等人发展的.

比如, 加强学习的主体, 对于加强学习的主体来说, 效用是它外部感知的一部分, Hutter[Hut05] 最开始提出的 AIXI 就是一种加强学习的主体; 追逐目标的主体, 对于追逐目标的主体来说, 效用很单纯, 只要在规定的时刻完成目标效用就是 1, 否则就是 0; 专职预测的主体, 对于专职预测的主体来说, 效用函数也很单纯, 如果成功环境下一步的反馈, 效用就是 1, 否则为 0; 寻求知识的主体, 寻求知识的主体是最有意思的一类变种, 它的效用不是外部环境赋予的, 而是自发驱动的, 纯粹为了追求“好奇”, 这种主体纯为探索“模式”而生, 所以对它来说, 往往不存在传统的探索/开发 (exploration/exploitation) 两难, 探索就是开发, 这使得它是“弱渐进最优的”; 自修改源代码的主体, 自修改源代码的主体有两部分组成, 它自身的源代码以及源代码的执行器, 代码执行器将源代码作用于当前历史并产生一个输出, 这个输出由一个动作和下一版自己将要变身的源代码构成; 可修改和被修改源代码和内存的主体, 一个能够进行自我欺骗的自修改源代码的主体有一个欺骗盒子可以把环境反馈回来的输入进行修改, 而且环境也有办法把主体想升级的源代码进行修改, 有意思的是, 对于“寻求知识”的 AIXI 变种, 即使

环境可以修改它的代码、即使允许它可以自己修改“观测数据”，它也不会进行“自我欺骗”；只为求存的主体，只为求存的主体的效用函数很简单，如果能保持初始的源代码不变其效用就为 1，否则为 0；内嵌时空的主体，对于内嵌时空的主体来说，它完全是环境的一部分，环境可以修改主体的任何部分，环境执行主体的代码，它的第一步可以是人为限定的某个不超过固定长度的程序，后面主体怎么改变就完全由环境控制了，下面的哥德尔机 Schmidhuber[Sch09] 可以看作这种“内嵌时空”的可修改源代码的主体的一例。

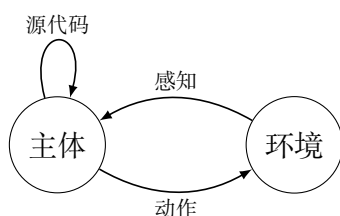


图 15: 自修改源代码的主体

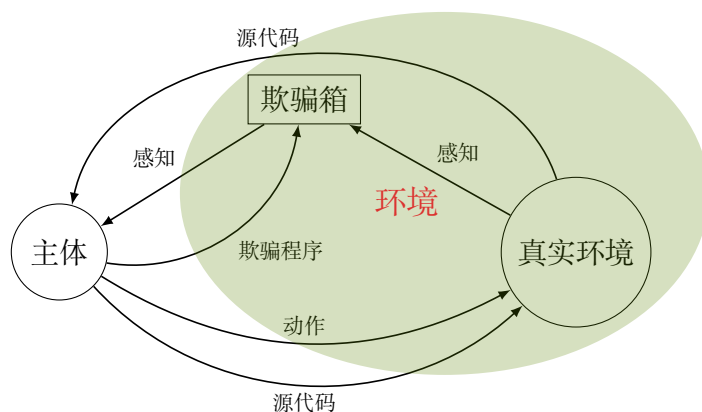


图 16: 可修改和被修改源代码和内存的主体

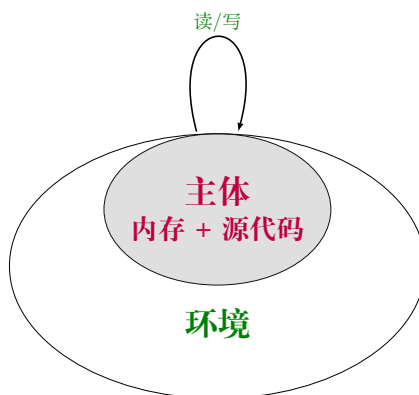


图 17: “内嵌于时空”的主体

哥德尔机 抽象地看, 一个智能体也无非是一段程序, 所以不妨设计某种“元程序”负责搜索整个“程序空间”、自动寻找“聪明”的程序, 然后通过经验学习寻找更“聪明”的程序. Hutter[Hut05] 在定义这种“元程序”时借鉴了 Levin[Lev73] 的通用搜索思想, 给出了 AIXI 的变种 $AIXI^l$, 限定在在时间 t 、空间 l 上, 它 (在乘上一个很大的常数界内) 理论上优于任何其它限定在时间 t 、空间 l 上的智能主体. Schmidhuber[Sch09] 把 $AIXI^l$ 进一步改进, 定义了“哥德尔机”, 它可以进一步缩小 $AIXI^l$ 的常数界. “哥德尔机”包含两个并行运行的部分 — “Solver”和“Searcher”: Solver 负责与环境交互, 尽可能最大化期望累积效用; Searcher 内嵌了一个形式系统, 形式系统里有对“Solver”、“Searcher”、“效用函数”的完全描述以及对环境的部分描述, “Searcher”可以对“哥德尔机”各部分 (包括 Solver 和 Searcher 自身) 的源代码进行彻底的修改 — 条件是 — 它内嵌的形式系统的定理证明器能证明“修改后的主体在未来的时间里将获得比现在更大的期望累积效用”. 这在一定程度上保证了对源代码的修改是相对安全的. 这样 Solver 和 Searcher 就可以比较安全的不断自我进化升级、趋向最优. 但是, 既然内嵌了形式系统, 那么它就面临哥德尔不完全性定理的障碍, 有一些重要且必要的“变身”可能无法被形式系统找到.

当前的主体技穷的时候, 人们总寄希望于进化的力量, 希望演化后的主体能更强大. 对于哥德尔机来说, 只要每一代给下一代装配更强的形式系统, 那么不完全性定理的障碍就可以突破, 但问题是 — 哥德尔第二不完全性定理, 主体 1 在构造主体 2 时如果不能在自己的形式系统内证明主体 2 的形式系统的一致性, 那么它根本无法保证主体 2 的可靠性. 无法保障可靠性, 就无法回避完全坍塌的风险; 对于人来说, 可靠性得不到保障也意味着机器智能的伦理问题得不到保障. 但是, 如果要求每一代主体必须严格证明下一代主体的形式系统的一致性的话, 那么, 这种进化在某种意义上是一种退化. 而生物的进化则不需要一致性的保证, 好的变异、不好的变异都可能产生, 自然选择的结果常常是、但不必然是优胜劣汰, 变异和自然选择不能保证可靠性, 哥德尔机面临的也是同样的问题. 退一步讲, 即使不谈演化, 如果把外部世界看做一个大的形式系统 T' , 那么, 哥德尔机内嵌的形式系统 T 的证明能力需要严格强于 T' , 因为 T 试图模拟 T' , 就必须比 T' 演化得快才有意义, 只有在一个严格强于 T' 的系统里, 对于同样的命题的最短证明长度才会短于 T 里的证明长度, 也就是说, 一方面, 从表达力和证明强度上说, 现实世界 T' 可以看做“模拟世界” T 的子系统, 另一方面, “模拟世界” T 又是现实世界 T' 的一部分, T 必须可以通过编码方式嵌入到现实世界 T' 中, 但是, 只要强系统 T 可以编码到弱系统 T' 中, 只要这种编码嵌入可行, 那么, 第二不完全性定理就无法绕过, T 自身的可靠性也得不到保障.

抛开“智能”的可靠性不谈, 退一步看, 哥德尔不完全性定理的幽灵是否仍对通用智能的发展设置了障碍?

10 通用智能与不完全性定理

关于这个世界, 最不可理解的是 — 它竟是可以理解的.

— 爱因斯坦

有些人倾向于相信, 现实世界不是可计算的, 它不比算术模型简单, 总有些真理我们不能以完全形式化的方式把握, 哥德尔不完全性定理为人工智能设置了障碍. 但如果现实世界是可计算的呢? 如果

“智能”不需要不可计算性呢？如果对不可计算的 AIXI 的某种可计算的逼近也可以涌现出“智能”现象呢？不完全性定理是否仍然对人工智能的实现构成威胁？

我们可以轻易枚举并模拟任何可能的计算模式，如果世界本身是可计算的，那么没什么是我们原则上不可理解的，而且，在混沌和分形中我们也曾多次见识到了简单程序生成“表观复杂”现象的神奇，这使得我们乐观的猜测，在这个五彩缤纷的世界背后起决定作用的程序是不是也异常简单呢？这里“表观复杂”的现象只要有简单的生成机制，按照算法复杂性的定义，它仍然是简单的。但如果它稍微复杂一些，依照 Chaitin 版本的不完全性定理⁵，形式系统就可能无法帮助我们区分它究竟是复杂还是简单，是“真随机”还是“伪随机”。世界是可计算的是种假设，对于任何可计算的世界，算法概率都可以很好的逼近它，为了保证可以逼近任何可计算的世界这种“通用性”，算法概率本身不是可计算的。如果我们限制要逼近的环境的类，有没有某种可计算的“通用模型”可以逼近所有算法复杂性不超过 n 的环境呢？对于任何复杂性水平 n ，这种可计算的“通用模型”都是存在的，但 Marcus Hutter 的学生 Shane Legg[Leg08] 证明，这种模型本身的算法复杂性也不会小于 n 。也就是说，只有自身也足够复杂的模型才能具有某种水平的“通用性”。要想具备强大的智能，自身必须达到一定的复杂性。奢求通过极度简单的程序应付任何复杂环境的希望注定落空。Shane Legg 还证明了一个类似 Chaitin 版本的不完全性定理：对于任何包含初等数论的形式系统 T ，存在某个复杂性水平 c ，对于任何高于 c 的复杂性水平 n ，形式系统 T 都无法帮助我们找到——可以逼近任何复杂性不超过 n 的环境的——“通用模型”，尽管这种模型是确实存在的。不严格的说，强大的智能体必然复杂，复杂且强大的智能体是存在的，但只要它足够复杂，形式系统将无法帮助我们找到它。

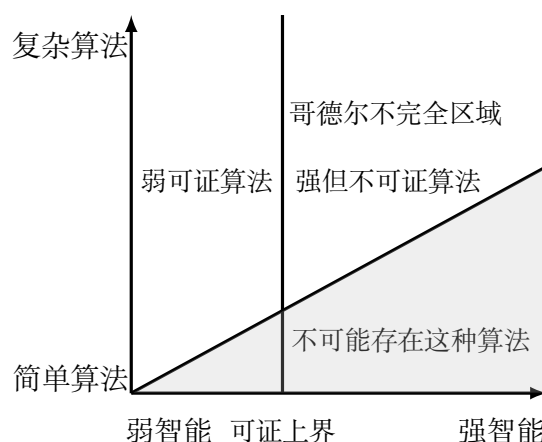


图 18: 不完全性定理的制约：强大的智能体本身必然复杂；对于任何形式系统，都存在一个界限，足够复杂且强大的智能体是存在的，但此形式系统无法帮助我们找到。

通用归纳、通用智能为人工智能的发展指明了方向、也设置了上界。如果现实世界没有那么纷繁复杂、可能的复杂模式都出现——或许现实世界真的没有想象的那么纷繁复杂，比如，一个简单的幂律分布就能在各种尺度上各种环境中支配着各种看上去不相干的现象，某些复杂的模式可能是存在的，

⁵Chaitin 版本的不完全性定理：对于任何包含初等数论的形式系统，都存在某个常数界，对于复杂性水平高于这个常数界的任何现象，此形式系统都不能告诉我们这现象的复杂性水平是否真的高于这个常数界。

但某些复杂 (虽然是可计算) 的模式可能不过是我们数学上的抽象构造, 未必真的都会被物理例示 — 那么, 算法概率就可以迅速的收敛到真实的现实世界, 某个可计算的且足够强大的能适应足够复杂的环境的智能体也可能不难被找到; 但如果现实世界确实包含很高复杂度的各种可能的模式, 那么, 简单的数学理论将帮不上忙, 寻找 AIXI 的可计算的且高效的逼近将是一件艰巨的任务. — 而且, 对于生活其中的人来说, 由于 Chaitin 定理, 如果复杂性高于某个常数, 我们可能无法明确区分二者.

除此之外, AIXI 还有一个缺点, 虽然通用归纳模型不依赖于通用图灵机的选取, AIXI 还是依赖的, 如果估测环境的“主观信念”所依赖的通用图灵机与衡量智能所依赖的通用图灵机不是一个, 也就是说, 如果定义“算法概率”时的“通用先验”所依赖的通用图灵机与量化“智能”衡量标准时的“通用先验”所依赖的通用图灵机不是同一个, 那么 AIXI 对智能的追求可能一开始就在一个错误的方向上.

总之, 一方面, 虽然在满足形式系统的限制、在忽略通用图灵机的影响、在不计算资源的情况下, 理论上存在最优的智能主体, 但革命实践尚未成功, 长路漫漫, 还需且行且珍惜! . 另一方面, 尽管 AIXI 摆脱不了哥德尔的幽灵、摆脱不了通用图灵机的依赖、在资源受限下对其逼近也非易事, 但 AIXI 的整体框架还是比较合理的 — 图灵可计算性概念抓住了物理世界有序性的本质、可能世界/假设就是可能的图灵机程序或可计算的测度 (这是数字/计算主义哲学乐于接受的), 枚举所有可能的假设 (伊壁鸠鲁), 根据“简单性”原则分配对各假设的先验信念 (奥卡姆剃刀), 用贝叶斯方法更新信念 (科学发现过程的体现), 用最大化期望累积效用的方法规划行为策略 (“理性人”的选择) — AIXI 的理论自身已足够为我们提供诸多指引, 虽然 AIXI (及其变种) 看上去是一个简单的模型, 但它如此优雅的以显式或隐式的方式整合了目前人工智能领域的方方面面, 如主体、效用、概率、假设、不确定性、归纳、压缩、预测、规划与决策、简单性与复杂性、泛化与过拟合、知识表示与存储、环境建模、逻辑定理证明、搜索与优化、内在驱动、增量学习、探索与开发、自我升级等等, 对 AIXI 的深入研究必将推动通用人工智能的发展.

限于篇幅, 本文主要介绍了 AIXI 背后的哲学思想, 想要详细了解 AIXI 及其变种的各种“最优”性质及详细证明的读者, 请参考 Hutter[Hut05] 以及本文第9节提到的参考文献.

References

- [Hut05] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Berlin, Heidelberg: Springer, 2005.
- [Hut09] Marcus Hutter. “Feature reinforcement learning: Part I. unstructured MDPs”. In: *Journal of Artificial General Intelligence* 1.1 (2009), pp. 3–24.
- [Leg08] Shane Legg. “Machine Super Intelligence”. PhD dissertation. University of Lugano, 2008.
- [Lev73] Leonid A Levin. “Universal sequential search problems”. In: *Problemy Peredachi Informatsii* 9.3 (1973), pp. 115–116.
- [Mni+15] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), pp. 529–533.

-
- [OLH13] Laurent Orseau, Tor Lattimore, and Marcus Hutter. “Universal knowledge-seeking agents for stochastic environments”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2013, pp. 158–172.
- [OR12] Laurent Orseau and Mark Ring. “Space-time embedded intelligence”. In: *International Conference on Artificial General Intelligence*. Springer. 2012, pp. 209–218.
- [RO11] Mark Ring and Laurent Orseau. “Delusion, survival, and intelligent agents”. In: *International Conference on Artificial General Intelligence*. Springer. 2011, pp. 11–20.
- [Sch09] Jürgen Schmidhuber. “Ultimate cognition à la Gödel”. In: *Cognitive Computation* 1.2 (2009), pp. 177–193.
- [Sol78] Ray Solomonoff. “Complexity-based induction systems: Comparisons and convergence theorems”. In: *IEEE Transactions on Information Theory* 24.4 (1978), pp. 422–432.
- [Ven+11] Joel Veness et al. “A monte-carlo aixi approximation”. In: *Journal of Artificial Intelligence Research* 40.1 (2011), pp. 95–142.