

算法之美

- 最优停止理论
 - 可能被拒
 - 掌握更多信息
 - 当等待有成本（卖房子）
- 探索与利用
 - 基廷斯指数
 - 上限置信区间
- 排序
 - 排序和搜索
 - 噪音
 - 自下而上的排序
 - 竞争和争斗
- 缓存
 - 整理归纳
 - 大脑记忆
- 时间调度理论
 - 应对时限
 - 密度
 - 切换任务
 - 颠簸
 - 中断合并
- 贝叶斯
- 过度拟合
 - 无处不在
 - 交叉验证
 - 惩罚复杂性
- 松弛
 - 约束松弛
 - 离散松弛
 - 拉格朗日松弛
- 网络
 - 指数退避
 - 流量控制

- 缓存膨胀

最优停止理论

最优停止理论的代表是秘书问题：面对依次出现的可选项你可以选择接受或者拒绝，你可以判断出哪个选项相对更加优秀，一旦你选择接受，游戏就会结束。

这个游戏的难点在于：过早接受会使得优秀的申请人可能还没有亮相就被踢出局，而太晚接受则可能错过最优秀的选项，最后不得不仓惶地做出选择。我们需要找到二者的平衡点。

要建立观察期，观察期之后发现更优人选果断选择，分界线一般在百分之37左右。即使样本持续扩大也对挑选到最好候选人的几率（约37）影响不大

可能被拒

邀约可能被拒：如果被拒可能性是一半，要在四分之一时就开始准备接受

延迟邀约可能被拒：如果延迟有一半可能被拒，可能把观察期扩大到60

掌握更多信息

我们以上之所以需要观察期是因为对于“优秀的分布”缺乏了解，但是如果从一开始掌握了更多信息（“优秀线”），我们就可以跳过观望直接选择一个阈值，超过阈值立刻录用。但是要注意，我们知道的只是一般“信息池”优秀的标准，对于每一个具体的信息池，仍然有可能存在遇不到超过阈值的情况，因此要随着进度的发展逐渐降低阈值，就像年纪越大越需要降低相亲要求一样。

当等待有成本（卖房子）

一般来说等待都有成本，这里的成本更显著一些，比如延迟买房子要交税和维护成本。因此买房子的目标不是得到最高的标价，而是最大化（标价-等待成本）。这里要考虑到报价平均差值，新报价出现频率，等候成本，这就成了一个比较期望值的问题。

探索与利用

是探索新的，寄希望于发现更好的呢，还是利用已知的好的，不去冒险，充分发挥其价值呢？

探索的价值随着时间的推移越来越低，利用的价值则越来越高，就像我们不会在要从这个村子搬出去时才考虑哪里适合建造新的水井。

那么如何找到二者的平衡呢，有人提出了赢留输变的策略，但这个策略可能筛掉那些偶尔失误但总体较好的选项，对所有选项的发挥优异概率进行确定虽然结果不错但是信息量工作量太大。

基廷斯指数

基廷斯指数可以动态的反应探索利用的潜在收益。

基廷斯指数的计算很复杂，但是如果我们已经得到了选项的基廷斯指数，我们可以发现一些规

律：

1. 赢就留下：同样的失败数，赢得次数越多指数越高。而每次失败则会相对降低指数。
2. 亲睐未知：没有玩过的选项的基廷斯指数要高于赢六输三，这说明基廷斯指数鼓励探索未知。亲睐未知的另一个体现是：赢一输一要比赢10输10更高，因为后者的确定性更高，而前者虽然期望值也是0，但是样本量小，偶然性大。

不过，不同选项之间的切换有时需要成本，这种时候基廷斯指数就不再是最有效的辅助工具。比如在选择如何分配精力的时候就要考虑到转移注意力会造成效率的降低，但这种降低并没有反映在指数里。同样，复利和收益递减也会扭曲基廷斯指数，二者都是持续投入某种能量，物质，信息随投入量或者时间的变化规律。这些因素使得我们在衡量探索利用的平衡时要具体情况具体分析。

上限置信区间

上限置信区间和基廷斯指数有相似之处，前者也对未知保持着乐观的态度：被检测次数较少的选项我们对其上限和下限的估计的范围更大，二者的平均值也会高于期望值。随着检测次数越来越多，置信区间会逐渐缩小，并越来越接近期望值。

排序

排序的规模越大，难度就以更大的幅度增大。

排序和搜索

尽管很多人把排序本身当成了目的，但排序本身是为搜索服务的，如果一定程度的混乱反而有利于搜索，就没有必要进一步排序。因此，我们要在排序和搜索之间找到一个平衡

噪音

真实世界的排序要比计算机中的排序存在更多噪音和误差。在各类排序算法中，比较计数排序抗噪能力最好，在这个算法里，每个被排序对象都会与其他所有对象比较大小从而确定位次。由于总共 n 个个体每个都要被比较 n 次，因此其运行速度只是 n^2 ，相比于合并算法来说很慢（ $n \log n$ ），但其抗噪性要远远好于合并算法。

我们可以把合并算法看作体育中的淘汰赛，而比较计数排序看作循环赛。如果说在一场对垒之中，更好的球队战胜对手的概率是0.7，那么四场淘汰赛过后，最好球队夺冠的概率是0.24，这样的排序误差就很大了。相比之下比较计数排序可以得到更符合真实实力的排名。

自下而上的排序

刚才讲的都是自上而下的排序，在自然界自下而上的排序更普遍一些，因为很少会存在一个“上帝”为他们排序。

动物之间的互动会让动物产生等级意识，即那部分动物有更高的资源获取的权限的意识。这样的意识可以避免很多动物之间一对一的争斗。这种情况与单挑扑克游戏类似，玩家在头脑中有关于自己大概所处的级别的认知，并可以选择与自己等级相近的对手单挑从而提高位次，这避免了很多跨级的无效争斗。

竞争和争斗

如果我们为需要排序的对象设立一共共同的基准，对他们的排序就可以简单很多。比如拳击比赛需要不断地一对一确认位次，而马拉松可以一次性排序几万人，因为完成马拉松的时间成为了这里的基准点。又或者把国家依据gdp排位。找到统一的量化基准就可以把争斗变成竞争

缓存

在存储和提取使用之间存在一个叫做缓存的过渡区域，用于保留一些在某段时间内可能被频繁提取的存储，进而极大的提高运行效率。就像你写论文的时候从图书馆借书回家放在办公桌上，而不是每次引用完一个段落就回图书馆还书一样。

因为容量较小，缓存面对的重要问题是，选择那些内容存于缓存，如何更新缓存里的内容。在这方面，有先进先出，也有最近最少使用。最近最少使用会清楚最近这段时间最少被使用的缓存内容，也表现出一些未卜先知的能力，因为我们需要的下一个内容可能就是之前需要的最后一个内容。

整理归纳

一般的整理归纳系统采用的是“物以类聚”和“金字塔”组合的方式。但使用缓存的最近最少使用原则可以有效的减少搜寻整个归纳系统的时间。和计算机一样，缓存要放置在距离使用地点较近的地方，具有比归纳系统小很多的内存。比如，在衣柜旁放置一个衣服桶，每次穿过的衣服放回桶内，并选择一个最近较少穿的衣服从桶内移出，每次选择衣服的时候则优先浏览桶内，没有满意的再去衣柜检索，久而久之常用的和不常用的衣服就被区分开来，省去了大量检索时间，也辅助我们进行断舍离。

大脑记忆

大脑记忆符合艾宾斯号曲线，而这个记忆规律也体现了缓存的原理：大脑记忆就像缓存，为最需要的事物留出空间，而判断什么是最需要的事物很大程度上使用了最近最少原则。

时间调度理论

我们应该如何分派时间，以什么样的顺序来做事，安排我们的生活

应对时限

现在我们先讨论单机调度，也就是只考虑一条时间线的问题。

如果我们把减少总共的延迟时间为目的，最佳的策略就是从截止日期最近的任务开始完成。

密度

切换到另外一个标准：我们不再有时间限制，不过希望让我们所做的事产生最大的价值。我们可以给每项任务赋予一个重要度，然后再估算他耗费的时间，就可以得到重要性密度，优先去做那些重要性密度较高的任务。

切换任务

如果我们可以随意暂停任务而没有成本，使用密度原则是很好的办法：在有新的任务出现时，评估其重要性，如果重要性高就切换到新任务，否则继续当前任务。

但现实中任务切换往往会付出代价。对人类来说，一些任务往往需要对于全局的关注，分割注意力可能会极大的影响效率。

颠簸

电脑运转速度的提升很大一部分要归功于缓存的使用，一个项目运行达到一定的时间，所需要频繁提取的部分就转变为内存。这时，切换任务就需要另一段较长的时间来覆盖之前的信息，因而导致了电脑全速运行但却一无所成的情况。

解决颠簸的方法可以是增加内存，也可以是选择性的接收新任务。就像我们可以通过提高工作记忆和拒绝干扰分心来提高生产力一样。

中断合并

反应速度和吞吐量在很大程度上是负相关的概念。如果我们的反应速度很快，也就是时间切片越小，我们就会不断地进行上下文切换。如果我们减慢以下反应速度，延长最小切片时间，就可以提高吞吐量，这也是番茄时钟的原理。对于需要进行很多上下文切换的工作，我们可以进行中断联合：把那些细碎的工作合并到一起处理

贝叶斯

根据事物的分布状态，按不同的原则进行预测

幂律分布：一个事物已经存在越长，可能就还会存在更长。 n^2

正态分布：使用平均法则，事物会趋向于平均值，当前落后平均值越少，越应该被赋予一个更大的“预期加成”

厄兰分布：事件在任何时期发生的可能性均等，比如购买同一个彩票多次。预测时使用加数法则，预计事物的状态还会再持续一个常量。

过度拟合

我们可能认为同一个问题考虑的因素，利弊越多越好。机器学习告诉我们，刻意的减少关注点，有时可以更好的做出预测。

一个包含了更多因素的模型，要比少因素的模型收集更多的用以预测的数据，但这并不意味着更好的结果。多因素模型相较于少因素模型更加敏感，更容易受到噪声的干扰，不容易保持稳定的状态。

如果我们的数据样本是那种具有完全代表性，没有差错，完全代表我们试图评估的事物时，使用更复杂的模型就可以更准确的预测未来。但是一旦数据不符合这些特征，就可能发生过度拟合。过度拟合是一种数据崇拜，他把人们的注意力放在了繁多的数据而非真正的问题上。

无处不在

味觉也是一种过度拟合，因为我们的味觉系统不断地收集那些有利于我们生存的食物和对应的味道。在古代，这些食物包括糖，盐，脂肪等等，但这些营养素在现代社会过于多见，反而成为了不利于健康的食物。这体现了我们的味觉系统在收集数据时出现的过度拟合，把不能完全代表我们希望评估的事物（健康度）的数据做成了预测模型。

或者例如一些军官在作战任务中会不自觉地收集空弹壳，因为在训练这些装备需要回收，这样的条件反射，体现了过度拟合。

交叉验证

检测过度拟合要求被检测者不仅可以出色的概括已有的数据，还要灵活应对未知的数据，这需要我们使用交叉验证的方法。

在人工智能里，可能需要我们将数据分成更多的子集，用部分子集进行训练，其余子集用于测试，若模型可能很好的概括训练的子集而在测试时表现不佳，则说明出现了过度拟合。

在其他场景中，也需要测试过度拟合。比如如果一个教育机制下学生对于常见模式的题目掌握度较高，但对于“非标准化”测试却表现较差，这说明存在着应试教育（过度拟合）。这类非标准测试可以包括口头面试，写作等等模式化较低的方法。

惩罚复杂性

施加一种下行压力使得各因素权重总体降低，只有对结果有很大影响的因素才得以被保留，这提高了模型的泛化能力。

松弛

约束松弛

面对一些即使计算机处理起来也很困难的问题，我们可以先尝试去除掉问题中的一些限制条件，取得进展之后再把约束条件添加进去。这个算法可以求一些无解问题的近似最优解。

也可以应用到生活中问题的思考上，因为生活问题总是充满了各种显性隐性的限制，比如：如果你获得足够经济独立的钱，会做什么？

离散松弛

很多问题的解决方案不是“连续的统一体”，而是断断续续的。比如我们邀请1, 2, 3等数量的客人，但是不能邀请1.2, 3.14个客人。但其实可以把离散问题改为连续问题，这并不意味着我们要向3.14个人发邀请，而是可以把这些数字当作概率或者设立一个采取行动的阈值等等。

拉格朗日松弛

另一种对问题进行松弛的方法要求我们把规则约束不再视为不可打破的铁律，而是某种一旦打破就会到来的惩罚。再通过把这个惩罚加入到问题中，我们便可以把问题当作数值最大化的问题。比如再安排体育赛事时，要考虑到转播，气候，主客队关系，国际赛程等等因素，其中某些约束被打破要比其它约束付出更大的代价，但是又很难找到完全符合所有约束的排列方式。这种情况下，把约束转化成惩罚分数就可以松弛问题。

网络

网络中两个端点的沟通总是存在信息丢失或者扭曲的可能性，因此双方交流中确认对方已经发送或接收到信息是很关键的。这就需要确认包和确认字符。确认字符保证每次发送的数据包都增加一个序列号，更具序列号的连续性就可以确认传输中有无丢失数据包。

指数退避

每一次信息传递都可能是双方在近期最后一次传递，因为对方随时可能掉线或者终止对话。我们需要一个机制来让双方确定多次时间的不响应应该被视为会话结束。

指数退避是为了避免双方同时发送信息进而产生干扰设置的机制，他是“打破对称”（比如两个相迎的人同时换道但是却继续重道）的好办法。在初次传输失败后，会在1-2回合内重新尝试，再失败则在1-4回合重新尝试，下次是1-8回合，依此等待的回合数区间不断指数扩大。指数退避原则可以很好的适应线路的繁忙情况，因为连续传输失败可能意味着线路非常繁忙，越繁忙退避的级数也就越大。

在生活中，指数退避可以被应用在需要多次接触时“接触可能失败”的情况。比如如果服务器繁忙，可以不断尝试访问，但是访问的频次指数级下降直到成功为止。甚至于在和一个人接触时，如果遇到“下头”的情况，可以采用指数退避的接触原则，减少对其接触但同时对接触的可能性予以保留。

流量控制

如何避免网络过载？

计算机采用了一种动态调节的方法：和式增加积式减少。因为线路可能出现拥堵，这种方法让沟

通中的双方，每次成功传输一个数据包，下次就会多传输一个。但一旦出现传接失效，则下一次传输的数据包数量减半。这样的设置可以防止过载同时让传输量尽量达到当前允许的最大。这种动态调节在动物界缺少中央决策者，比如蚂蚁中，一样存在。

和式增加积式减少并不是一个严格的逻辑教条，而是代表着一种无计划下动态调节的理念，对于其他有限资源的最大分配问题同样适用。有的公司也采用了类似最低职位聘用毕业生考察之后升职or辞退的策略。

缓存膨胀

为了使网络线路尽可能多的承载信息，网络队列存在一个缓冲区，就像餐厅排队的人群一样。缓冲区造成了处理的延迟，但这可以保证网络始终在最大限度的运行。当网络的缓冲区也已经塞满时，将不得不使用尾部丢弃的方法把最后达到或最老的数据包丢弃