

Toggle navigation

Greenwicher's Blog

- [首页](#)
- [归档](#)
- [分类](#)
- [标签](#)
- [订阅](#)
- [读书](#)
- [CV](#)
- [Resume](#)

请输入您的邮箱

一键订阅

深度增强学习【2】从多臂赌博机问题到蒙特卡洛树搜索

一键订阅 | Greenwich's Newsletter

输入邮箱，为您科普 人工智能|量化交易|算法之美

订阅

两个重要的增强学习（强化学习）问题：多臂赌博机问题和蒙特卡洛树搜索

有的人苦于没有选择可选，只能一条路走到黑；而有的人选择过多、权衡过多，反而无从下手，相当于没做选择。约束太多有约束太多的烦恼，太过自由有太过自由的烦恼，这也算是一种 [资源诅咒](#) 吧。然而面对选择困难症，到底有没有灵丹妙药来解决它。科学家说，必须有！先来考虑一个问题，某天你走了狗屎运天上掉下了1000块钱，你觉得自己的运气正旺，而且反正是~~不义之财~~，不如去赌场来以小博大。赚了算自己的，赔了就当没捡过这钱。你兴冲冲的跑去了赌场去玩老虎机，然而到了赌场却傻了眼，竟然有200台老虎机！随便选一个老虎机可不行，因为你听人说过有的老虎机赢率比较高，有的比较低。那么问题来了，给定这1000块钱，假定玩一次老虎机要支付1块钱，那么应该玩哪些老虎机、以怎样的顺序去玩才能使得自己的累积收益最大？在 [上篇深度增强学习系列文章](#)，我们讲到了Alpha Go中的两个关键技术：深度学习和增强学习。本文将首先介绍 多臂赌博机问题（[Multi-armed Bandit Problem](#)），然后基于此，介绍Alpha Go的另一项核心技术，即 蒙特卡洛树搜索（[Monte Carlo Tree Search](#)），最后会简要介绍我的研究方向 仿真优化（[Simulation Optimization](#)）同这些领域的关系。

多臂赌博机问题

问题描述

In probability theory, the multi-armed bandit problem (sometimes called the K- or N-armed bandit problem) is a problem in which a gambler at a row of slot machines (sometimes known as "one-armed bandits") has to decide which machines to play, how many times to play each machine and in which order to play them. When played, each machine provides a random reward from a probability distribution specific to that machine. The objective of the gambler is to maximize the sum of rewards earned through a sequence of lever pulls. –Wiki

给定 K 个老虎机，每玩一台老虎机，在损失一次机会的同时我们以一定的概率收到报酬。假定第 $i = 1, 2, 3, \dots, K$ 个老虎机给我们报酬 $r \in \mathbb{R}$ 的概率是 $P_i(r)$ ，且报酬的均值为 h_i 。那么决策便是，给定有限的机会次数 T ，如何玩这些老虎机才能使得期望累积收益最大化。记第 t 轮，老虎机给我们的报酬为 r_t ，那么至少有三种方式来刻画这个期望累积收益函数

- T 步累积奖赏： $E[\frac{1}{T} \sum_{t=1}^T r_t]$
- γ 折现累积奖赏： $E[\sum_{t=0}^{T-1} \gamma^t r_{t+1}]$
- 累积遗憾： $Th^* - E[\sum_{t=1}^T r_t]$ ，其中 $h^* = \max_{i=1,2,3,\dots,K} h_i$



Las_Vegas_slot_machines.jpg

可以看到多臂赌博机问题其实是一个很宽泛的框架，能够应用到各种 排序选择问题（[Ranking & Selection](#)）上。比如说新药的研发需要临床检验，那么给定多款新药以及一批受试者，如何利用有限的机会选择出效果最好的新药并减轻对受试者不必要的副作用；再比如说商家在互联网上投放广告，那么给定几款广告设计方案，如何展示这些不同的广告给不同的网民，以致能够尽快的最大化利润并且不流失顾客。后一个例子最简单的情况就是 A/B 测试，[这篇文章](#) 详细描述了Google Analytics内容实验背后的统计引擎，即多臂赌博机实验。国外也有一些初创公司在试图利用多臂赌博机问题的思想来优化企业的收益管理问题，比如说[Optimizely](#)，以及国内的[吆喝科技](#)。

探索（Exploration）vs 利用（Exploitation）

如果我们有先知的本领，那么为了最大化期望累积收益，每一次选择报酬均值 h_i 最大的那个老虎机就好了。然而，正因为信息的缺乏，我们需要通过手上的资源来逐步获取信息。

先考虑一种极端的方式，那就是均匀的玩每台老虎机，这样可以保证对每台老虎机的收益情况我们都能足够了解。然而，这往往会浪费不必要的资源在太差的老虎机上，就像无头苍蝇一样。这种策略被称之为 探索（exploration），很显然如果信息太过于匮乏的话，这种策略不失为一种好方法。

再考虑另一个极端的方式，那就是只玩当前给我们收益报酬最高的那台老虎机，显然这一策略可以在初期较快的获得更高的回报，然而却因为过度贪婪捡了芝麻丢了西瓜，以至于长远错过真正好的老虎机，就像辛勤的蜜蜂一样。这种策略被称之为 利用（exploitation），很显然如果信息足够充分的话，这种策略不失为一种好方法。

最好的策略显然不是这二者之一，中和这两种截然矛盾的资源分配策略可以给我们更好的思路。比如说，前期信息匮乏，我们采用更多的探索；而后期，信息了解差不多后，我们转向利用，诸如高中生以及博士生的关系。再比如说，再边利用的同时也进行探索，诸如Google的传统部门以及Google X之间的关系。但这里值得一提的是，如果问题规模过大，比如说资源的数量不足以支撑探索尽量多的信息，那么反而利用是一种更『现实』的策略。金融危机之后，经常有人说

大而不能倒。Too big to fail.

虽然这句话的本意是为避免大公司的倒闭而引发系统性风险，政府不能坐视不管。但让我们从创新的角度来重新看这个问题，很多知名的初创公司就像一只独角兽，因为它代表一种[相对]罕见的新事物；而大公司呢，估计就是骆驼了吧，虽说瘦死的骆驼比马大，并且骆驼凭借自己积累的驼峰便足以让他在沙漠当中生存，但受限于自己的优势反而不能开拓新的疆界。正如万维钢老师说的，[『创新是落后者的特权』](#)。

基本算法

具体的算法实施请见[我的Github](#) 或者 [Bandit Algorithms for Website Optimization](#) 这本书，下文只是简要叙述每种算法的思路。如日后有时间，再将各自的代码补上。值得一提的是，该领域主要的理论在于求解累积遗憾的上下界。

ϵ -greedy算法

顾名思义，就是每次选择老虎机的时候，以 ϵ 的概率进行exploration，然后以 $1 - \epsilon$ 的概率进行exploitation。值得注意的是，exploitation和exploration的比例可以通过动态的调整 ϵ 来实现，比如说让 $\epsilon = 1/\sqrt{t}$ ，这样在算法初期将主要进行exploration，而

在算法后期进行exploitation，这样的处理思路也比较符合直觉。

Softmax算法

这一算法是基于Softmax distribution¹来选择老虎机的，值得一提的是Softmax distribution可谓在机器学习里随处可见，比如说神经网络的输出层，Logistic Regression以及模拟退火算法，个人感觉是因为softmax distribution实际上和矩母函数的关系非常紧密，另外也算是对max函数的光滑处理吧²，最后最直观是这是一个multi-choice model（因而在经济管理领域也有应用）。总而言之，我们选择老虎机 i 的概率服从Softmax分布，

$$P(\text{select arm } i) = \frac{\exp(\bar{h}_i/\tau)}{\sum_{j=1}^K \exp(\bar{h}_j/\tau)}$$

这里的 $\tau > 0$ 类似于模拟退火中的温度参数，若 $\tau \rightarrow 0$ ，那么将只选择最好的那个，也就是exploitation；反之，若 $\tau \rightarrow +\infty$ ，那么将均匀的选取老虎机，也就是exploration。类似的，我们对参数 τ 也可以进行退火处理（温度逐渐的降低），即 $\tau = 1/\sqrt{t}$ 。另外，关于均值的计算，可以采用当前观测值以及上一期的均值来进行增量计算，这样可以减少重复计算。

Bayes Bandit算法

这一算法的基本思路是给定老虎机收益的先验分布，然后通过该分布来决定玩哪个老虎机，收集到信息之后再更新后验分布。为了保持先验分布和后验分布的形式一致，往往需要 共轭分布（[Conjugate Distribution](#)）的帮助。比如说各个老虎机的回报服从伯努利分布，为了估计伯努利分布的参数，起初我们假设该参数均匀分布在[0,1]之间，然后其共轭分布便是beta分布，并且在更新的过程中，总是保持beta分布的形式，只是相应的参数有所变化。

Upper Confidence Bound算法

这一算法不同于上面的三个算法，每次选择老虎机依据的标准是确定的，即上置信值（Upper Confidence Bound Value），即老虎机给我们回报的置信区间上界值。具体而言，老虎机 i 在第 t 轮的这一值通过如下公式进行计算，

$$\text{ucb}(i) = \bar{h}_i(t) + c \sqrt{\frac{\log t}{n_i(t)}}$$

其中 $\bar{h}_i(t)$ 和 $n_i(t)$ 是老虎机 i 在第 t 轮的样本均值以及观测次数， c 决定了置信概率。然后在每一轮，选择上置信值最大的老虎机去玩就可以了。

其他算法

- Exp3 / Exp4: [The Nonstochastic Multiarmed Bandit Problem](#)
- Knowledge Gradient: [A knowledge-gradient policy for sequential information collection](#)
- Randomized Probability Matching: [A modern Bayesian look at the multiarmed bandit](#)
- Thompson Sampling: [An Empirical Evaluation of Thompson Sampling](#)
- Gittins Index: [Bandit Processes and Dynamic Allocation Indices](#)

Demo展示

下面的幻灯片是我博一的时候，在IE5504 Systems Modeling and Advanced Simulation上做的展示，分析比较了UCB算法、Bayes Bandit算法以及OCBA算法在解决Bernoulli Bandit问题上的差异。具体的程序代码以及文档，请见 [我的Github](#)。

蒙特卡洛树搜索

问题描述

Monte Carlo Tree Search (MCTS) is a method for making optimal decisions in artificial intelligence (AI) problems, typically move planning in combinatorial games. MCTS combines the generality of random simulation with the precision of tree search.

John von Neumann's 1928 minimax theorem paved the way for adversarial tree search methods that have formed the basis of decision making in computer science and AI almost since their inception. Monte Carlo methods were later formalised in the 1940s as a way to approach less well-defined problems unsuitable for tree search through the use of random sampling. Rémi Coulomb combined these two ideas in 2006 to provide a new approach for move planning in Go now known as MCTS.

Research interest in MCTS has risen sharply due to its spectacular success with computer Go and potential application to a number of other difficult problems. Its application extends beyond games, and MCTS can theoretically be applied to any domain that can be described in terms of {state, action} pairs and simulation used to forecast outcomes.

- <http://www.cameronius.com/research/mcts/about/index.html>

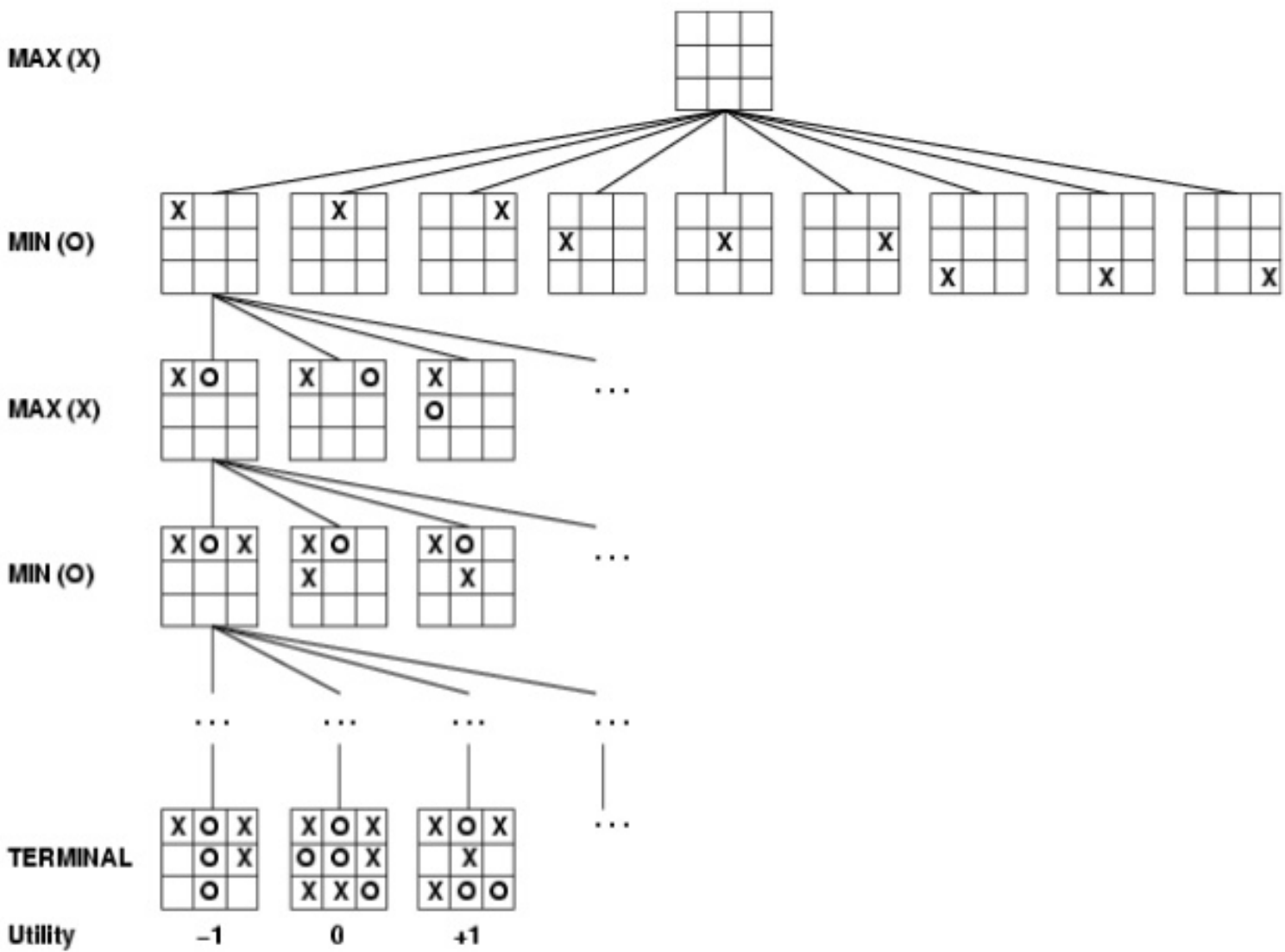
如果说多臂赌博机问题被看做 单步强化学习任务（只用一步决策玩哪个老虎机，然后就收到回报），那么蒙特卡洛树搜索可以看做是解决 多步强化学习任务 的工具。树 是一种天然的用来刻画或者存储多步决策的数据结构。正如所有的动态规划问题可以被转化为图搜索³，而所有的线性规划问题可以被转化为二分图⁴一样。在树上进行搜索再常见不过了，利用树进行仿真其实也没那么不常见，学金融的同学势必都曾利用过二叉树来为各类奇异期权进行定价。至于蒙特卡洛树搜索，实际上可以分为两步

- 利用树结构来重新表达决策问题
- 利用蒙特卡洛方法来进行搜索

下面将简要的介绍这两个方面。

决策树结构

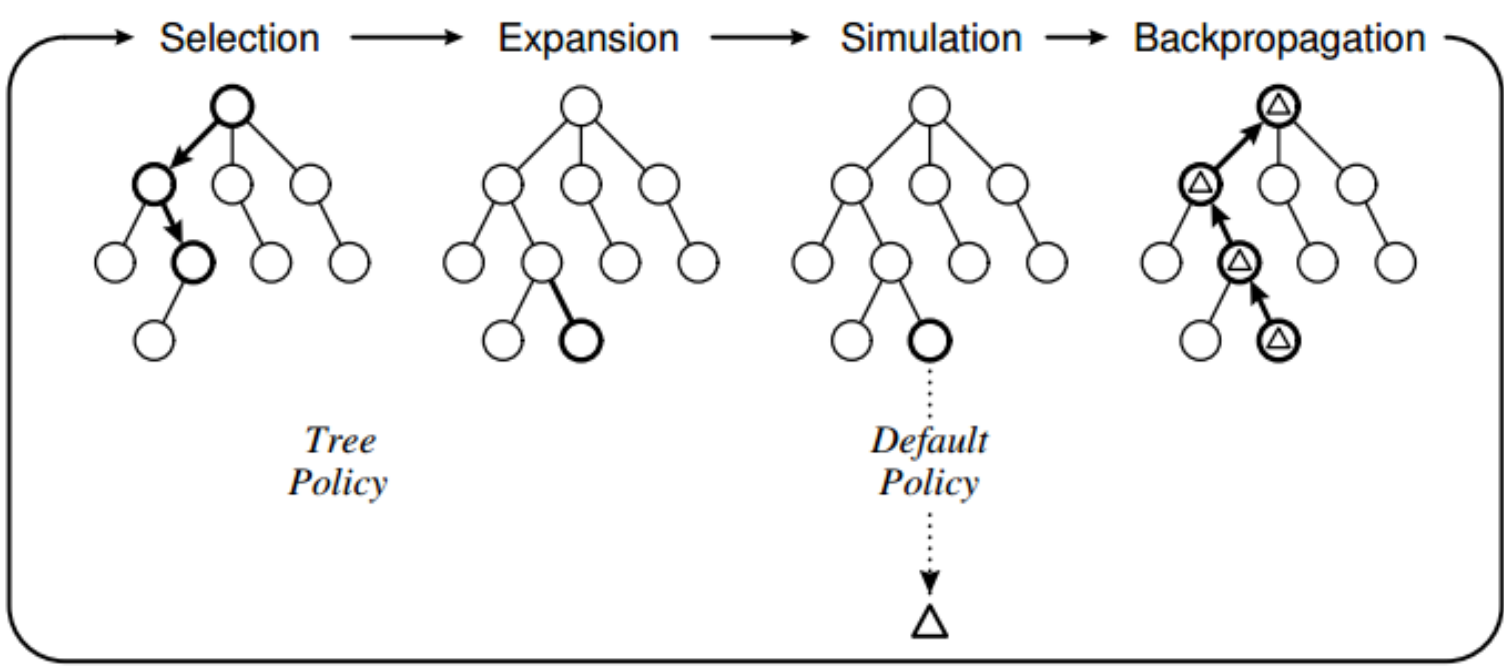
对于多步强化学习任务而言，我们做完一个决策之后，将依据该决策和导致的新状态来决定我们下一步的决策集合。因此给定一条决策路径，我们站在当前的决策节点上，之后的所有决策可以形成一棵子树。任意一条从根节点到叶结点的路径变形成了一个完整的决策，当我们做完最终的决策，将得到一定的回报，正如下图所示。



Decision_Tree.jpg

蒙特卡洛方法

蒙特卡洛方法是一种基于仿真或者说采样来获取信息的方法。给定决策树结构，假想我们站在某个节点上，接下来有 K 个选择，那么我们应该选择哪一个？实际上，从这 K 个选择中给定任意一个，其之后的决策构成了一棵子树，我们都可以利用蒙特卡洛方法来对这棵树的所有路径进行随机抽样（Simulation），根据得到的样本均值来评估这个子树的根节点选择，并且将得到的信息传递给所有的父节点（Backpropagation）。然后确定好下一步的选择之后（Selection），我们将继续基于该选择来展开未来的选择集合（Expansion）。下图是经典的蒙特卡洛树搜索概念图，



Monte_Carlo_Tree_Search_Idea.png

简而言之，从这 K 个选择中进行抉择实际上可以转化成一个多臂赌博机问题。比如说，最著名的Upper Confidence Bound 1 applied to trees（UCT）算法。好玩的是，我们甚至可以利用蒙特卡洛树搜索来搜索更优秀的蒙特卡洛树搜索⁵，这浓浓的自指味道实在让人惊叹！

Demo展示

说起来也蛮幸运的，感觉冥冥之中自有力量。博一上学期在IVLE里不断搜自己想要上什么课。后来发现计算机系开了一门CS5330 Randomized Algorithms，但苦于找不到具体的授课大纲，于是给授课老师发了邮件，大致了解了这门课的内容。后来发现这门课虽然叫 Randomized Algorithms，但其核心内容还是多臂赌博机问题以及延伸过去的蒙特卡洛树搜索，关于各种概率不等式以及上下界的介绍相对较少。了解了相应内容后，发现和自己的研究方向还蛮相关的（当然了，后来发现其授课内容其实算是online learning），于是就选了这门课。

那以下的幻灯片就是我在这门课上的大作业展示，描述了如何利用蒙特卡洛树搜索来解决一个图染色的组合优化问题。具体而言，给定一个无向图和一串红蓝相间的颜色序列，初始给一个节点按照颜色序列的顺序给其染色。接下来，我们可以给未染色并且是已染色点的相连点进行染色，并且必须按照颜色序列的顺序染色。如果颜色序列用尽或者所有的点都被染上色，那么任务结束。最终的收益为连接两个不同颜色点（已染色）的边的数量，并且越大越好。

值得一提的是，我的代码基于[Monte Carlo Search Algorithm Discovery for One Player Games](#)这篇文章，给出了实现蒙特卡洛树搜索的通用框架。也就是基于这个框架，可以很简单的构造出任意类型的蒙特卡洛树搜索算法，也就是从某种程度上说，可以用算法来生成算法（将代码作为数据，或者说一切都是对象，函数式编程的思想）。具体而言，我对比了Reflexive Monte Carlo搜索算法、Nested Monte Carlo搜索算法以及著名的Upper Confidence Bound 1 applied to trees搜索算法。

关于这个组合优化问题，当时课上同学有两个思路让我记忆犹深：

- 有人用Wolfram Mathematica可视化了不同数据集的无向图，基于此来找规律，有点类似于领域知识的探索
- 有人没用蒙特卡洛树搜索，而是用了贪心算法，考虑到有的无向图规模太大，反而取得了不错的成绩

具体的程序代码以及文档，请见[我的Github](#)。

作为同构的仿真优化

类似的问题在运筹学/管理科学领域被称之为 [仿真优化](#)。简而言之，就是如何利用高效的仿真来解决随机优化问题。常见的分类包括

- Ranking & Selection
- Discrete Optimization via Simulation (Heuristic methods / Random Search)
- Response Surface Methodology
- Stochastic Approximation
- Sample Average Approximation
- Stochastic Gradient Estimation
- Variance Reduction Techniques

更多详细的介绍请见[Handbook of Simulation Optimization](#)。下面将大概介绍Ranking & Selection以及Discrete Optimization via Simulation，同多臂赌博机问题以及蒙特卡洛树搜索之间的同构关系以及差异。

Ranking & Selection

类似于多臂赌博机问题，都是给定有限个选择，一般数量不多，但每个选择的仿真时间相当长，因而需要合理的分配仿真资源；不同的是多臂赌博机问题更在乎累积收益，而Ranking & Selection主要是要尽量以高概率找到最好的选择出来，并不在乎仿真过程中的收益或损失。因此，在Ranking & Selection这个领域，主要的目标是如何通过分配仿真资源（即怎么玩老虎机），以使得最终选出最好老虎机的概率最大。这个概率被称之为 Probability of Correct Selection。下图是一个简要的文献文类，我目前正在做的主要是OCBA这一块。

CATEGORIZATION OF LITERATURE FOR R&S PROBLEMS				
Formulation	Single Objective	Multi-objective	Feasibility	Constrained Optimization
IZ	Frazier [19]	Teng et al. [20]	-	Andradóttir and Kim [21], Healey et al. [22]
OCBA	Chen et al. [8]	Lee et al. [23]	Gao and Chen [24]	Lee et al. [9]
EVI	Chick and Inoue [16]	Frazier and Kazachkov [25]	-	-
LD	Glynn and Juneja [11]	Feldman et al. [26], Hunter and Feldman [27] & This work	Szechtman and Yücesan [28]	Hunter and Pasupathy [29]

Ranking_Selection_Literature.png

OCBA的核心思路是利用信噪比来分配仿真资源，即仿真资源应该尽可能的分配给噪声大（比如样本方差）的选择以及信号比较强烈（同当前最优选择比较接近）的选择。可以看到从多臂赌博机问题的角度出发，OCBA相对更倾向于exploration，因为其目的只是关心『最终』能选择出最好的选择，而不在乎每次仿真得到的收益。

Discrete Optimization via Simulation

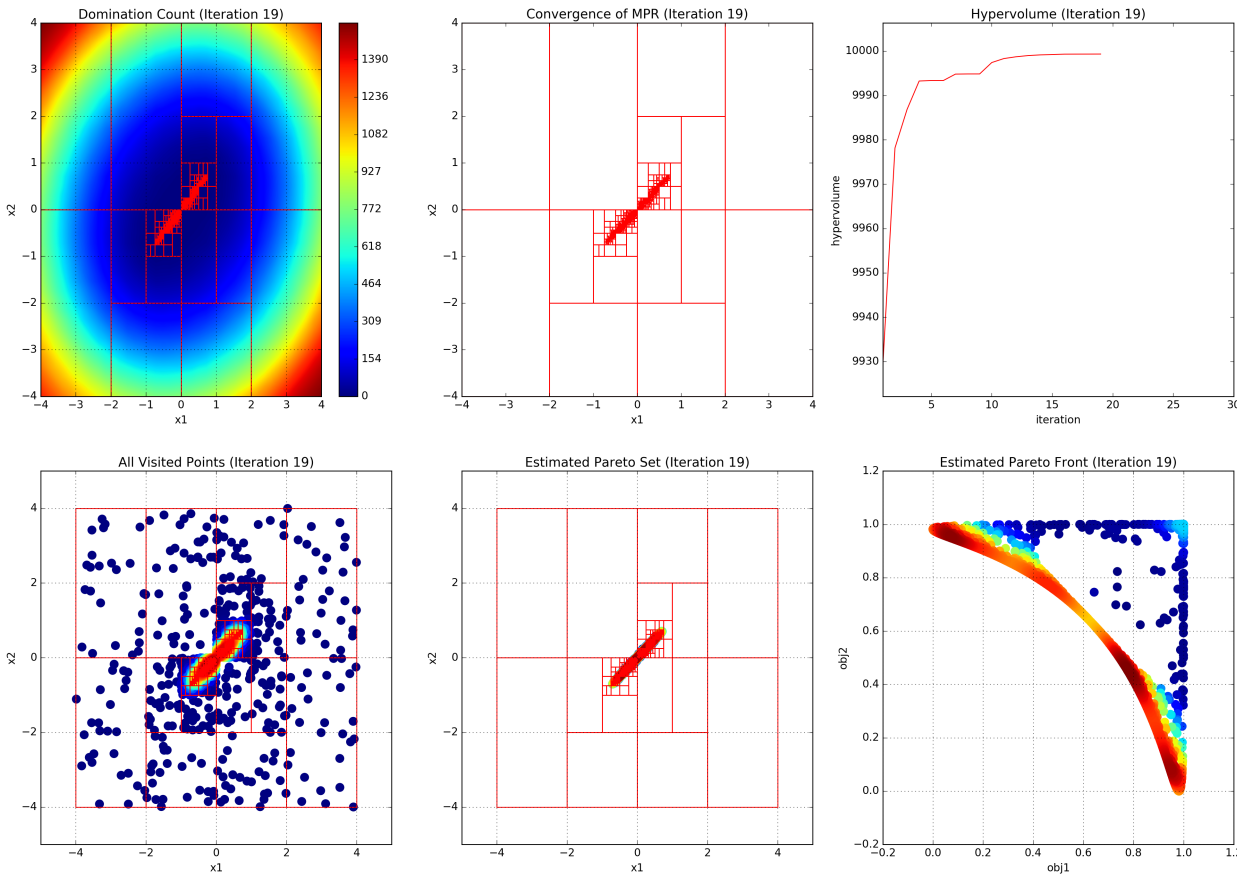
这一问题相对Ranking & Selection，一般是选择数量过多，然而针对每一个选择的仿真时间很短，因此重点在于如何进行搜索。下图给出了这一领域的文献分类，

Table 1: Categorization of Literature for Random Search

Methodology	Single Objective / Deterministic	Single Objective / Stochastic	Multi-objective / Deterministic	Multi-objective / Stochastic
Model-based	Costa et al. (2007)	Sun et al. (2014)	Hale & Zhou (2015)	Gutjahr (2003)
P2P-based	Kirkpatrick et al. (1983)	Alrefaei & Andradóttir (1999)	Bandyopadhyay et al. (2008)	Gutjahr (2005)
Population-based	Holland (1992)	Chang (2015)	Deb et al. (2002)	Li et al. (2015)
Partition-based	Shi & Ólafsson (2000)	Shi et al. (2000)	Sourd & Spanjaard (2008)	This Work

Discrete_Optimization_via_Simulation_Literature.png

类似于蒙特卡洛树搜索的算法在该领域称之为Nested Partition或者Partition-based Random Search，即给定一个高维实数域上的连续搜索空间，通过不断的对搜索空间进行细分形成树状结构进行有偏采样（biased sampling）。具体可参考侍乐媛老师的文章，[Nested Partitions Method for Global Optimization](#)。下图是目前我正在做的利用Partition-based Random Search来解决多目标随机优化问题的一个例子。



FON_Deterministic.png

结语

虽然，看完本文章，该选择困难的你还是继续『选择』选择『困哪』』』，毕竟哪有那么多资源去一一尝试，那就贪心一点吧，执念于当前最好的，就行啦！少即是多，无即是有，做一个极简主义者，断舍离！

关于【深度增强学习】系列的说明

对于我自己而言，写【深度增强学习】这一系列文章，除了自己的兴趣之外，其实增强学习和我的研究方向（仿真优化）也略微相关，希望能从中获取些新知识和新想法。初步打算本系列文章以[David Silver的公开课](#) 以及 [UC Berkeley的CS294](#)为蓝本，着重在增强学习领域，

陆续会补充深度学习的相关探讨。记录自己的所学所思，力图抓住主要核心。毕竟吾生也有涯，而知也无涯。以有涯随无涯，殆已！但学习是一辈子的事情，所以时不时也会重新补充或者修改这些文章。本人初涉深度增强学习领域，还希望各位学界业界大牛多多指正文章中不当之处，互相切磋，谢谢！

延伸阅读

- [Wiki - Multi-armed Bandit Problem](#)
- [Wiki - Monte Carlo Tree Search](#)
- [Dr. Sebastien Bubeck's Blog](#)
- [Dr. Sebastien Bubeck's Publications](#)
- [Finite-time Analysis of the Multiarmed Bandit Problem](#)
- [A Survey of Monte Carlo Tree Search Methods](#)
- [From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning](#)
- [谷歌Analytics: 多手柄老虎机实验](#)
- [Chapter 2, Reinforcement Learning \(2nd edition\)](#)
- [第十六章，《机器学习》](#)

Footnotes:

1

也称作Boltzmann distribution，最早是描述分子运动的

2

softmax distribution的均值随其参数变化，将包含观测值从最小到最大的所有可能取值

3

详见[Algorithm Design的第六章：Dynamic Programming](#)

4

这是某次听港科大齐向彤老师讲的

5

详见[Monte carlo search algorithm discovery for single-player games](#)



本作品采用[知识共享署名 2.5 中国大陆许可协议](#)进行许可。欢迎转载，但请注明来自[Mount Greenwich](#)的文章[《深度增强学习【2】从多臂赌博机问题到蒙特卡洛树搜索》](#)，并保持转载后文章内容的完整与无歧义。本人保留所有版权相关权利。

一键订阅 | Greenwich's Newsletter

输入邮箱，为您科普 人工智能|量化交易|算法之美

订阅

据说爱打赏的人运气都不会差
谢主隆恩



微信打赏



支付宝打赏

- [上一页](#)
- [首页](#)
- [下一页](#)

留言

0条评论


Greenwicher

1 登录

推荐

分享

按评分高低排序



开始讨论.....

来做第一个留言的人吧！

在 GREENWICHER 上还有

深度增强学习【1】走向通用人工智能之路

2条评论 • 6天前 • 刘威志 — 谢谢,拖延了好久,希望最近有时间继续更新该系列

工欲善其事，必先利其器

3条评论 • 9天前 • 刘威志 — 贱人你来啦,[阴险]

📧 订阅  在您的网站上使用 Disqus添加 Disqus添加  隐私

新大陆被发现了1332次！
Dec 24 2016

-
- [如何创造AI2](#)
- [人工智能2](#)
- [仿真优化1](#)
- [增强学习2](#)
- [多臂赌博机问题1](#)
- [深度增强学习2](#)
- [蒙特卡洛树搜索1](#)
- 1. [多臂赌博机问题](#)
 - 1. [问题描述](#)
 - 2. [探索（ Exploration ）vs 利用（ Exploitation ）](#)
 - 3. [基本算法](#)
 - 1. [ε— greedy算法](#)
 - 2. [Softmax算法](#)
 - 3. [Bayes Bandit算法](#)
 - 4. [Upper Confidence Bound算法](#)
 - 5. [其他算法](#)
 - 4. [Demo展示](#)
- 2. [蒙特卡洛树搜索](#)
 - 1. [问题描述](#)
 - 2. [决策树结构](#)
 - 3. [蒙特卡洛方法](#)
 - 4. [Demo展示](#)
- 3. [作为同构的仿真优化](#)
 - 1. [Ranking & Selection](#)
 - 2. [Discrete Optimization via Simulation](#)

- 4. [结语](#)
- 5. [关于【深度增强学习】系列的说明](#)
- 6. [延伸阅读](#)
- 7. [Footnotes:](#)

© 2017 LIU Weizhi with help from [Hexo](#) and [Twitter Bootstrap](#). Theme by [Freemind](#).
本站共迎来了4390个小伙伴，总计8420次站点流量.

[友荐云推荐](#)