不围棋 AI 文档

董海辰 518030910417

2018 年 11 月 11 日

目录

1 基本信息

AI 代号 White Album 2(下简称 WA2)

代码长度 10088B

关键算法 Monte Carlo 树搜索及其优化

2 设计思路

在传统的 Monte Carlo 树搜索 (MCTS) 中, 将棋盘状态以落子转移保存为树结构, 对于一个节点进行一定次数的随机模拟, 来判断一个节点及其父节点的价值 (获胜概率).

对于一个局面随机模拟的庞大计算量限制了模拟次数, 导致在每次估价过程中没有足够的数据来判断价值, 成为 MCTS 算法的主要瓶颈, 因而 WA2 的主要设计思路即是增加每个节点的样本数量.

3 具体实现

3.1 MCTS

每次沿着当前搜索树, 每次选择期望胜率最高的节点找到叶子 (未被扩展的节点), 进行随机模拟, 并对整课树进行更新. 一个节点的模拟次数达到一定阈值后则进行下一层的扩展.

3.2 随机模拟

算法 对每个同色联通块进行 BFS, 若仅有一口气且为对手颜色则不可行 (导致吃子), 或者是某空格四方的最大己方的气为 1 时不可行 (导致自杀). 每次等概率选择可以落子的位置, 直到游戏结束.

复杂度 $O(n)$(其中 n 为棋盘大小).

3.3 更新节点

算法 对于一次对叶子节点局势随机模拟的结果, 传统的 MCTS 算法会更新从叶子到根所有节点的模拟次数 (n) 和胜利局数 (w).WA2 在此基础上新加了属性估计模拟次数 (n_1) 和估计

胜利局数 (w_1). 将模拟胜率 $\frac{w}{n}$ 和估计胜率 ($\frac{w_1}{n_1}$) 相结合来评估当前局势的价值:

$$v = (1-\lambda)\frac{w}{n} + \lambda\frac{w_1}{n_1}$$

其中 λ 是与 n 相关的系数, 随着 n 的增大而趋近于 1, 因为显然真实的模拟次数越多, 其结果越是可信的, 而模拟次数越小, 就越需要通过其他途径获得的预估胜率. 在 WA2 中:

$$\lambda = \sqrt{\frac{1000}{1000+3n}}$$

显然在 $n > 1000$ 时, 倾向模拟胜率, 在 $n < 1000$ 时, 倾向估计胜率.

估计方法 考虑到在中前期, 落子的顺序对局势的影响不大, 也不会改变最终的局面. 所以考虑对叶子到根路径上的每一个节点, 不仅更新在本次模拟中实际落子位置带来的影响:

$$n = n + 1$$

$$w = w + [win == nodePlayer]$$

也考虑所有后继操作 (包括本次) 中本方落子位置带来的估计影响:

$$n_1 = n_1 + 1$$

$$w_1 = w_1 + [win == nodePlayer]$$

在代码中, 即对每个节点, 找出所有距离为偶数的祖先节点, 更新其对于本次落子位置的后继状态 (如果存在), 为节约空间 (偏后期时, 一个节点的子节点数量很少), 子节点用 std::map 维护.

复杂度 $O(k^2 \log n)$(其中 k 为模拟深度,n 为棋盘大小)

3.4 其他

3.4.1 节点存储

- unsigned long long a[3] : 状态压缩储存棋盘, 每个位置为 0/1(B)/2(W).
- int color,fa,exd,mvx,mvy : 当前玩家 (上一步落子方的颜色), 父节点, 是否扩展 (extended), 得到当前局势的上一步落子位置 (mvx,mvy).
- int lson,rson : 由于使用数组方法存储搜索树, 讲每个节点的子节点储存为连续的一段, 以方便找到最优子节点时的遍历.

3.4.2 关于 UCT

信心上限树 UCT(Upper Confidence bound applied to Tree) 通过引入参数 c 使得模拟次数更多的节点置信程度更高. 但在 WA2 的实际操作中, 发现加入

$$c = k\sqrt{\frac{\ln N}{n}}(k \text{ 参数}, N\text{为总模拟数}, n\text{为当前节点模拟数})$$

后战斗力大量下降, 推测原因可能是预估的胜利次数与信心无关, 引入参数后也不利于开拓更多的节点, 故最终舍弃了 UCT, 仅适用传统 MCTS 的估价方法.

3.4.3 其他

由于没有刻意针对某个 AI 也没有考虑什么搞一些奇怪的 trick, 唯一的特判即是若四个角未被占据, 则在棋盘角落子, 感性认为这样可以让对手更容易吃子.

4 一点感想

为了 (至少开始很) 有趣的 NoGo 去熬夜, 去翻论文和 wiki, 去肝代码真的一件快乐的事, 也是大学生活的船新体验. 大作业的意义绝不是对战网站上跳动的数字和不断刷新的红红绿绿, 也不仅仅是几个博弈算法和奇技淫巧, 而是许多人为了一个目标不断奋 (e) 斗 (jing) 的充实的快感和成就感.

5 参考资料

1. https://en.wikipedia.org/wiki/Monte_Carlo_tree_search
2. Yuxia Sun, Ziyang Zhang, Xiaoyan Wang, The Research of UCT and Rapid Action Value Estimation in NoGo Game, 2016