Monte Carlo II

Ronald Kroening, M.S CS

Table of Contents

Monte Carlo Logic

Question: In Monte Carlo, what is used to measure what path is taken?

Monte Carlo Logic

$$UCT = Q(v) + E * \sqrt{(\ln(N(v)) / N(v\_child))}$$

UCT stands for **U**pper **C**onfidence bound for **T**rees. It is a metric that takes into account:

Q(v): Our estimated value of the current node

E: Our epsilon, or exploration value

N(v): How many times the current node was visited

N(v_child): How many times the child was visited

It was originally proposed by Levente Kocsis and Csaba Szepesvari at the Hungarian Academy of Sciences in 2006. What they found was that UCT was able to work much quicker with less data and achieve the same error rate as other algorithms.

(AB: Alpha-Beta, MC: Vanilla MC, MMMC: Minimax Value Update)

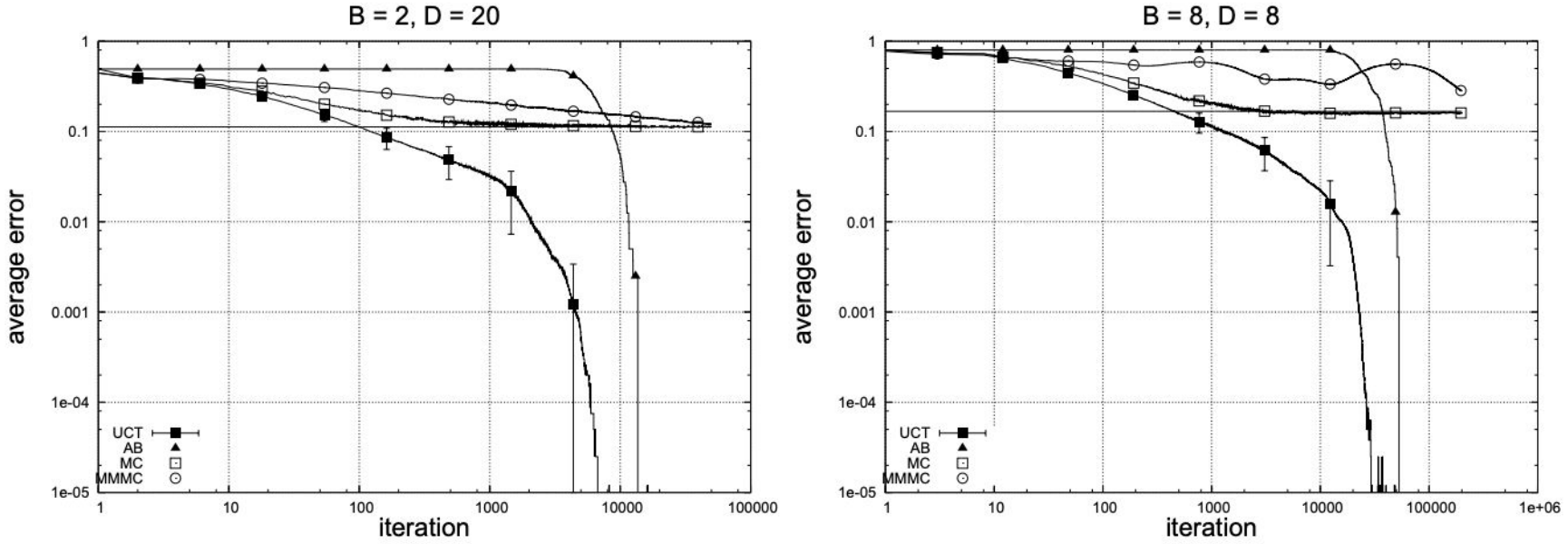


Fig. 2. Failure rate in P-games. The 95% confidence intervals are also shown for UCT.

UCT

Benefits

- Allows for further exploration of nodes by including number of visits
- Provides a flexible, general-use method for analyzing a given environment

Limitations

- Reliance on Exploration can hinder learning

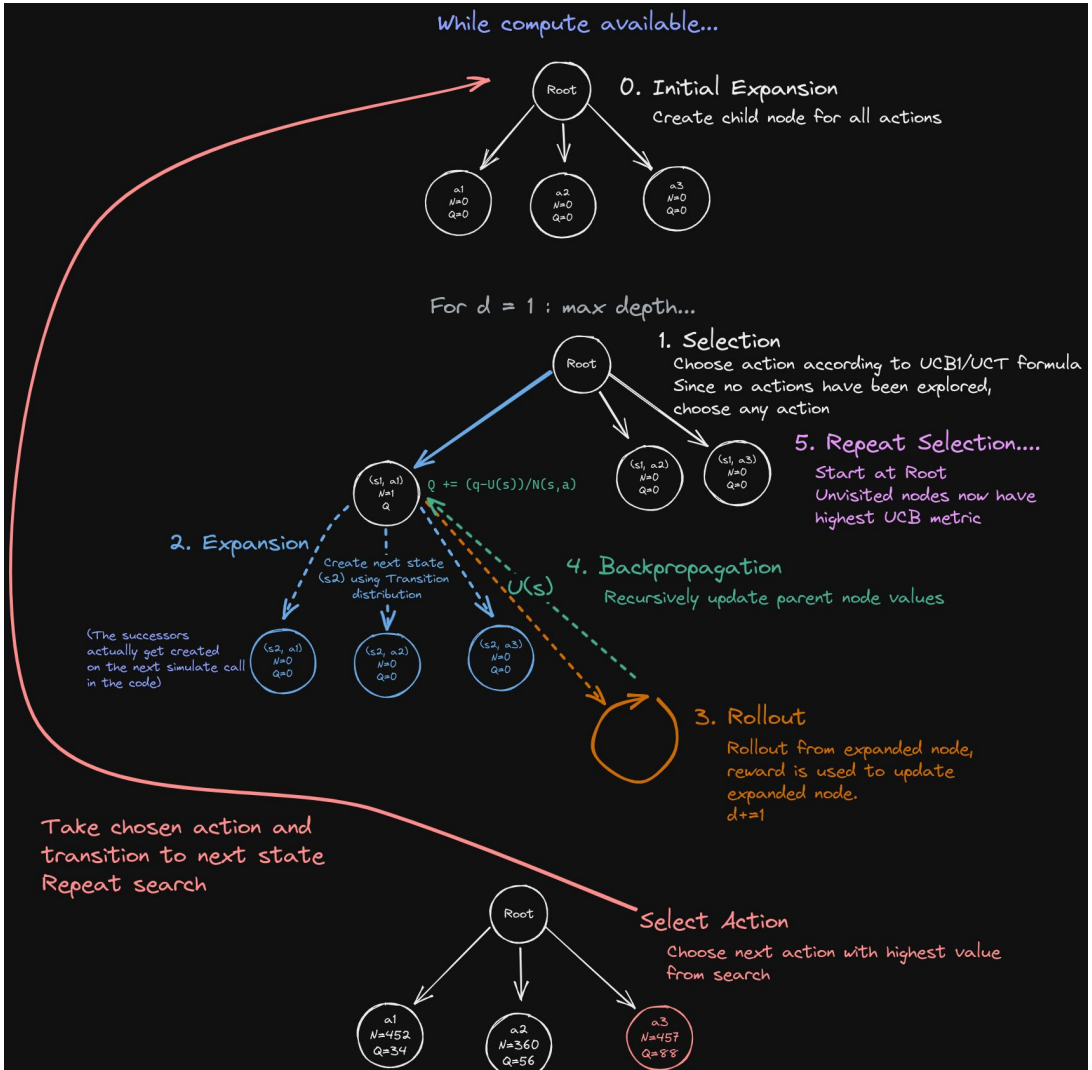
UCT Improvements

RAVE

- Improves data collection during MCTS by keeping more information on the actions available regardless if they were taken
- Allows for MCTS to learn quicker earlier on when data is scarce
- Implemented in the backpropagation function

PUCT

- Generates weights for each initial action in order to better predict which action will be optimal.
-

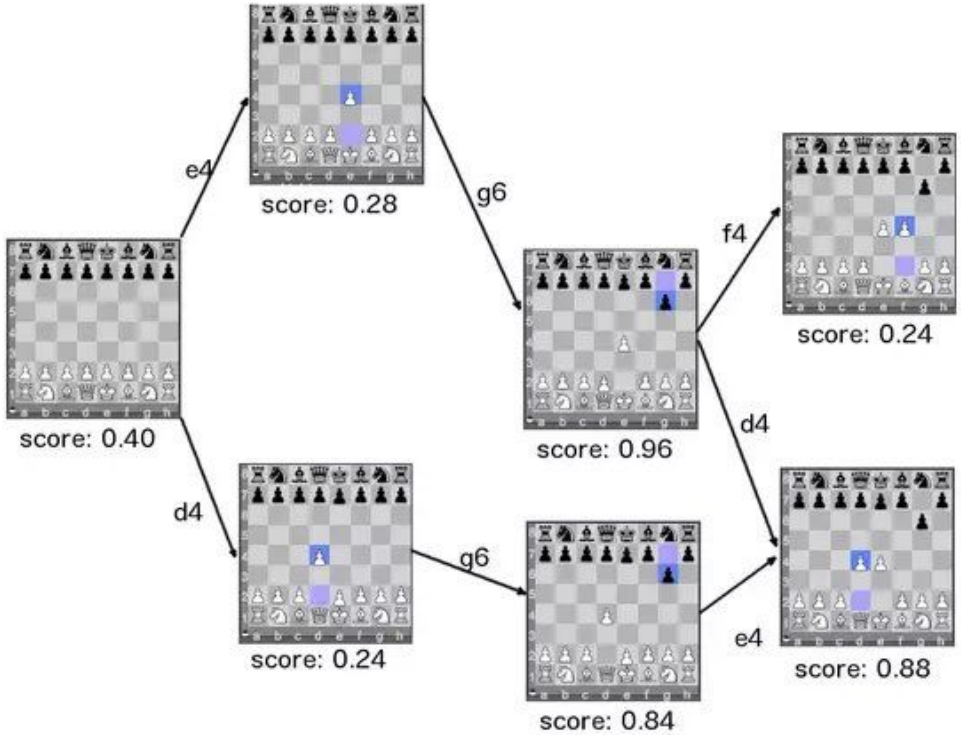

While compute available...
Root
0. Initial Expansion
Create child node for all actions
a1
N=0
Q=0
a2
N=0
Q=0
a3
N=0
Q=0
For d = 1 : max depth...
Root
1. Selection
Choose action according to UCB1/UCT formula
Since no actions have been explored,
choose any action
5. Repeat Selection....
Start at Root
Unvisited nodes now have
highest UCB metric
(s1, a1)
N=1
Q
Q += (q-U(s))/N(s,a)
(s1, a2)
N=0
Q=0
(s1, a3)
N=0
Q=0
2. Expansion
Create next state
(s2) using Transition
distribution
U(s)
4. Backpropagation
Recursively update parent node values
(The successors
actually get created
on the next simulate call
in the code)
(s2, a1)
N=0
Q=0
(s2, a2)
N=0
Q=0
(s2, a3)
N=0
Q=0
3. Rollout
Rollout from expanded node,
reward is used to update
expanded node.
d+=1
Take chosen action and
transition to next state
Repeat search
Root
Select Action
Choose next action with highest value
from search
a1
N=452
Q=34
a2
N=360
Q=56
a3
N=457
Q=88

Case Study I: Chess

Q1: Just looking at this board, what would 100 be a reference to?

Q2: In a game of chess, how would you assess the effectiveness of a board?

Case Study I: Chess



Case Study I: Chess

- Terminal State:
  - King has no moves left and is in check
  - no actions are possible
- Actions
  - Moving a pawn
- Rewards
  - 1 if win
  - -1 if lose
  - 0 if draw
  - Generates legal moves only so no need to worry about punishing illegal moves

Case Study II: Tic Tac Toe

https://ai-boson.github.io/mcts/

Required Functions for Running MCTS

- Node representation that can store
  - The parent node
  - Action taken from the parent
  - How many times it was visited
  - The rewards from its children
  - List of Actions from that node that have not been tried
  - The children possible from the node (future states)
- As actions are taken, new nodes are generated.
- All MCTS actions take place in this node class

Further Reading

- AlphaGo and AlphaZero papers
- “A Survey of MCTS Methods” published in IEEE’s Transactions on Computational Intelligence and AI in Games, Vol. 4, No. 1, March 2012
  http://www.incompleteideas.net/609%20dropbox/other%20readings%20and%20resources/MCTS-survey.pdf

Works Cited

“Bandit Based Monte-Carlo Planning.” Kocsis and Szepesvari, *CITESEERX*, citeseerx.ist.psu.edu/doc/10.1.1.102.1296. Accessed 3 Oct. 2023.

K. Matsuzaki, “Empirical analysis of PUCT algorithm with evaluation functions of different quality,” *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2018. doi:10.1109/taai.2018.00043

“MCTS Algorithm Tutorial with Python Code for Students ” Monte Carlo Tree Search (MCTS) Algorithm Tutorial and It’s Explanation with Python Code., ai-boson.github.io/mcts/. Accessed 3 Oct. 2023.