Multi-Armed Bandit

Given unknown probability distributions $R_1, \ldots, R_K$ with means $\mu_1, \ldots, \mu_K$ $\qquad \mu^* = \max_{1 \le i \le k} \mu_i$

Choose $i_1, i_2, i_3, \ldots$ to maximize total payout

Regret = difference between what you got and best possible expectation

$$\rho_T = T \cdot \mu^* - \sum_{t=1}^{T} \hat{r}_t \qquad \hat{r}_t = \text{reward at time } t$$

"optimal" means $P\left(\lim_{T\to\infty} \frac{\rho_T}{T} = 0\right) = 1$

Ex:

Arm 1 prob	Arm 1 payout	Arm 2 prob	Arm 2 payout	Arm 3 prob	Arm 3 payout
$\frac{1}{3}$	2	$\frac{1}{4}$	3	$\frac{1}{100}$	200
$\frac{2}{3}$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{99}{100}$	0
		$\frac{1}{2}$	0		
$\mu_1 = \frac{2}{3}$		$\mu_2 = \frac{7}{8}$		$\mu_3 = 2 = \mu^*$	

uniform rotation: cycle through each arm

1 2 3 1 2 3 $\underbrace{1\ 2\ 3}$, . . .

expected regret over 3 plays $= \frac{4}{3} + \frac{9}{8} + 0 = \frac{59}{24}$

$$\lim_{T\to\infty} \frac{\rho_T}{T} = \frac{59}{72} \quad \left(= \frac{59}{24} \cdot \frac{1}{3}\right)$$

greedy : play each once, then highest observed payout forever

1 2 3 1 1 1 1 1 1 . . . $\qquad \lim_{T\to\infty} \frac{\rho_T}{T} = \frac{4}{3}$ prob > 0

1 2 3 2 2 2 2 2 . . . . $\qquad \lim_{T\to\infty} \frac{\rho_T}{T} = \frac{9}{8}$ prob > 0

1 2 3 3 3 3 . . . . . . $\qquad \lim_{T\to\infty} \frac{\rho_T}{T} = 0$ prob < 1

ε-greedy: play 1 round, then play best observed reward w/ prob $1-\varepsilon$, random arm w/ prob ε

$$\lim_{T\to\infty} \frac{\rho_T}{T} = \underbrace{\frac{\varepsilon}{k}}_{>0} \cdot \sum_{i=1}^{k} \underbrace{(\mu^* - \mu_i)}_{>0 \text{ for non-optimal}} > 0$$

(assuming not all optimal)

zero regret

tunable parameter

total plays

zero regret

Choose arm j that maximizes $\bar{r}_j + \sqrt{\dfrac{z \cdot \ln T}{n_j}}$

tunable parameter

total plays

UCB
(upper confidence bound)

avg observed reward for j

$\#$ times arm j played

exploitation

exploration

$$P\left(\lim_{T\to\infty} \frac{\rho_T}{T} = 0\right) = 1$$

Monte Carlo Techniques

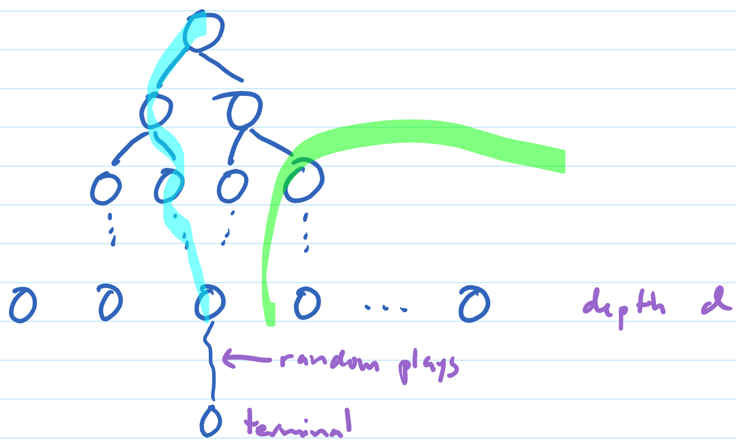
Flat Monte Carlo: for each action simulate to terminal using random play

choose action w/ highest observed average

Scrabble, Bridge

Combine with UCB: choose action to max $\bar{r}_j + \sqrt{\frac{2 \ln T}{n_j}}$

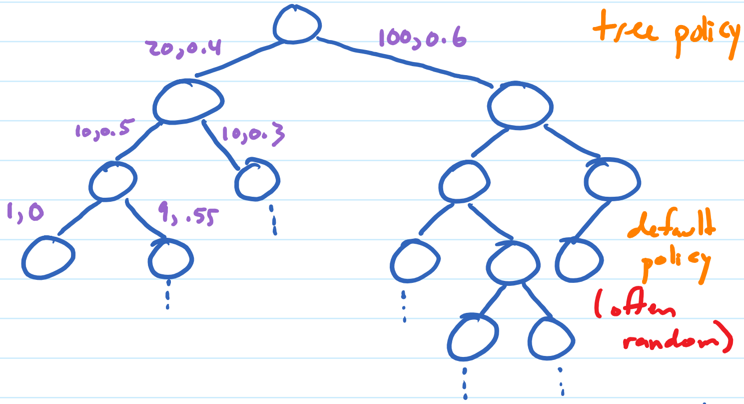
Combine with tree search: build tree to depth d
(Flat UCB)
traverse to leaf using UCB at each level
at leaf, play randomly to end
propagate stats back along path



Grow Tree asymmetrically: Monte Carlo Tree Search

↳ explore good parts deeper than bad parts

Monte Carlo Tree Search



while time left

tree policy traverse tree root → expandable node or terminal node

some missing children

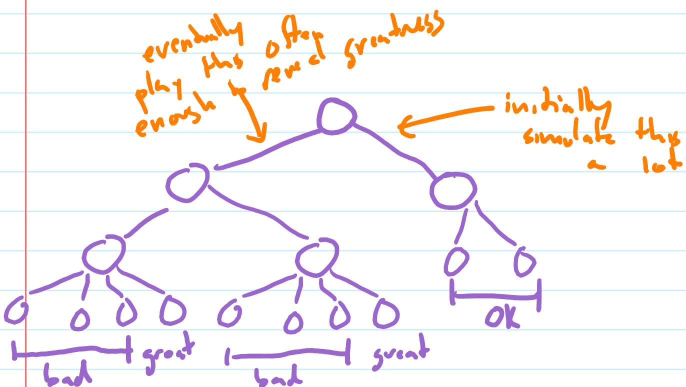
add child

playout from child

backpropagate result

return best child of root

highest observed reward
(or played most often)
(or keep going until same)



UCT = MCTS + UCB as tree policy

advantages: convergent — converges to minimax

anytime — can suggest move after any iteration (compare to iterative deepening)

no domain knowledge — no heuristic

easily parallelized

leaf parallel — multiple parallel playouts from leaf

root parallel — separate tree on each CPU
combine results in end

tree parallel — parallel traversals in same tree
(but needs locking)
if playout time >> traversal time
waits for locks insignif

disadvantages: no domain knowledge

some games not amenable

Tree vs DAG

default policy: random

move-averaged sampling technique (MAST)

PAST (predicate averaged sampling techniques)

		n	r
P_1 :	A has more in store	100	0.52
P_2 :	B has more in store	50	0.4
P_3 :	A has no pit w/ >2 seeds	50	0.38
P_4	$\sim P_3$	100	0.55
P_5 :	A has empty pits	50	0.7
P_6 :	$\sim P_5$	100	0.3

MCTS tree data → playability of game?