Markov Decision Process: (S, A, P, R) — states (positions), actions (moves) (or $A(s)$), rewards (scores)

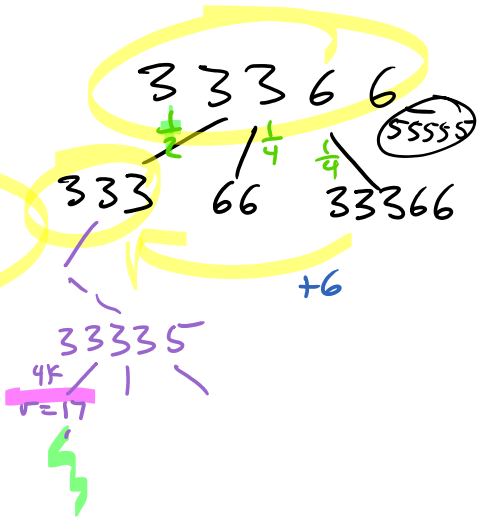
$P: S \times A \times S \times R \to [0,1]$ — current state, action, next state, reward

$p(s', r \mid s, a)$ = prob. of reaching s' w/ reward r given current state s, action a

$p(s' \mid s, a) = \sum_r p(s', r \mid s, a)$

expected reward $r(s, a) = \sum_r r \cdot \sum_{s'} p(s', r \mid s, a)$

$r(s, a, s') = \sum_r \frac{r \cdot p(s', r \mid s, a)}{p(s' \mid s, a)}$



initial state, terminal state

Episode: $(S_0, a_0, S_1, R_1), (S_1, a_1, S_2, R_2), \ldots, [(S_{T-1}, a_{T-1}, S_T, R_T)]$

$G = R_1 + R_2 + R_3 + \cdots$ Goal: maximize sum of rewards

$= R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots = \sum_{t=1}^{\infty} \gamma^{t-1} R_t$

$0 < \gamma \leq 1$

for finite process, can take $\gamma = 1$

Policy: $\pi: A \times S \to [0,1]$ s.t. $\sum_a \pi(a \mid s) = 1$ → prob action a taken in state s

for deterministic policy $\pi(a \mid s) = 1$ for one a for any given s

write $\pi(s) = a$ when a is

Value: $v_\pi(s) = \sum_a \pi(a \mid s) \cdot \sum_{s'} \sum_r p(s', r \mid s, a) \cdot (r + \gamma v_\pi(s'))$

$q_\pi(s, a)$

value of states under policy π = expected future reward from state s following π

0.5

all new states s', rewards r; immediate reward; future rewards from new state s'

$q_\pi(s, a_1) < q_\pi(s, a_2)$

0.5	0.5
1.0	0.0
0.0	1.0

optimal value

$v^*(s) = \max_\pi v_\pi(s)$

$q^*(s, a) = \max_\pi q_\pi(s, a)$

Bellman Eq

$\pi^*(s) = \operatorname{argmax}_a q^*(s, a)$

$$q^*(s,a) = \max_\pi q_\pi(s,a)$$

$$\pi^*(s) = \arg\max_a q^*(s,a)$$

which π gives max? π that gives max $v_\pi(s')$; which gives $v^*(s')$

Bellman Eq

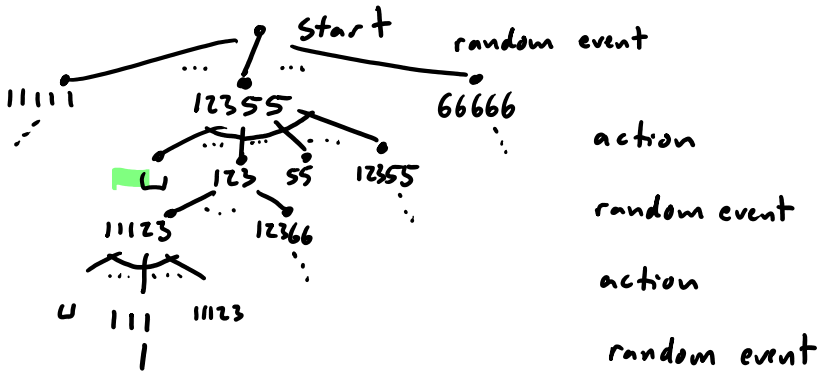
$$v^*(s) = \max_{a \in A(s)} q^*(s,a) = \max_a \max_\pi q_\pi(s,a)$$

$$= \max_a \sum_{s',r} p(s',r|s,a) \cdot \left[r + \gamma v^*(s') \right]$$

Some deterministic π gives max (for q^* too)

for a finite process, this is a recurrence (base cases for terminal states)

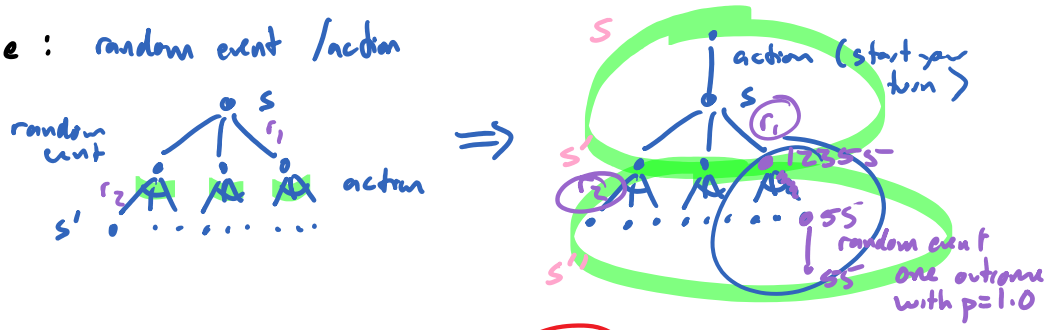
Alternative Views



Traditional Markov Decision Process: action / random event

$$v^*(s) = \max_a \sum_{s',r} p(s',r \mid s,a) \cdot [r + \gamma \cdot v^*(s')]$$

~~Typical~~ Many Game: random event / action



for random event (in original): $v^*(s) = \max_a \sum_{s',r} p(s',r \mid s,a) \cdot [r + \gamma \cdot v^*(s')]$

$$= \sum_{s',r_1} p(s',r_1) \cdot [r_1 + \gamma \cdot v^*(s')]$$

for player action: $v^*(s') = \max_a \sum_{s'',r} p(s'',r \mid s',a) \cdot [r + \gamma \cdot v^*(s'')]$

assume deterministic rewards

$$= \max_a r(s',a) + \gamma v^*(n(s',a))$$

$n(s',a)$: state s'' that results from taking action a in state s'

Yahtzee

anchor : start of turns

component : states between start of one turn and next

number of anchors : attempt 1 (all in bb)

ones... sixes	7^6
3K, 4K	28^2
C	27
FH, LS, SS, Y	3^4
Yahtzee bonus	13

$\geq$ 2 trillion anchors

- 1800 states / anchor
$\approx$ 4 quadrillion states

50 days @ 1 billion / sec

very fast

ones ... sixes	used/unused	2^6
upper total	0...63	64
3K,... C	used/unused	2^6
Y	used 10/50	3

$\frac{3}{4}$ million

~1 minute (Java code)

Yahtzee Graph