lower γ → faster convergence

Last Time

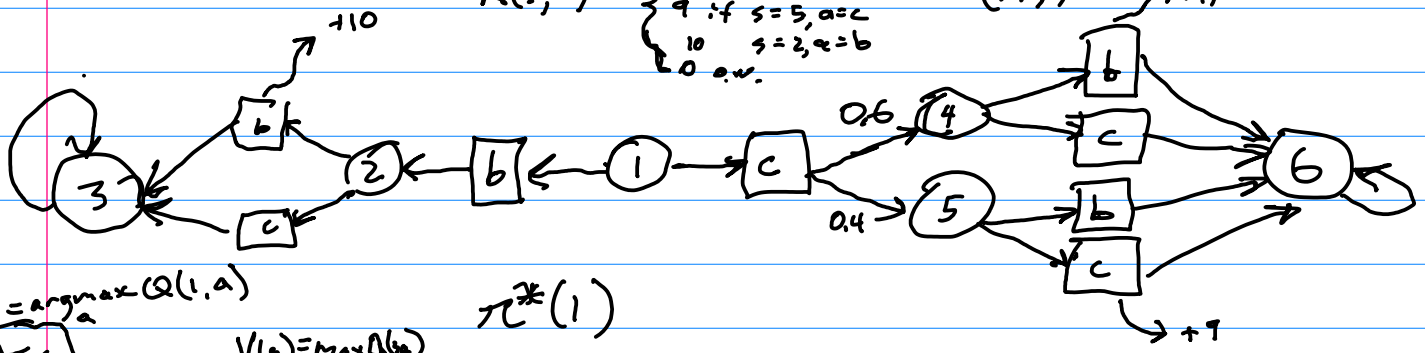
Proved VI converges
Bellman Operator, $B[V]$, is a contraction mapping in $(\mathbb{R}^{|S|}, \|\cdot\|_\infty)$
V^* is unique

This Time

→ Debugging and Perf.
→ Online Solution Methods

Breakout Rooms $S = \{1, \ldots, 6\}$ $A = \{b, c\}$ $\gamma = 1$

$$R(s,a) = \begin{cases} 11 & \text{if } s=4, a=b \\ 9 & \text{if } s=5, a=c \\ 10 & s=2, a=b \\ 0 & \text{o.w.} \end{cases} \qquad T(s'|s,a)$$



$\pi^*(1) = \arg\max_a Q(1,a)$

$\pi^*(1) = c$

$\pi^*(1)$

$V(s) = \max_a Q(s,a)$

$$Q(s,a) = R(s,a) + \sum_{s'} T(s'|s,a)\, V(s')$$

	V	Q b	Q c	b	c
1	10.2	10	10.2	$0 + 1.0 \cdot 10$	$0 + 0.6 \cdot 11 + 0.4 \cdot 9$
2	10	10	0	$10 + 0$	$0 + 0$
3	0				
4	11	11	0	$11 + 0$	$0 + 0$
5	9	0	9	$0 + 0$	$9 + 0$
6	0				

Offline

P.I. V.I.

Before Execution: find V^* / Q^*

During : $\pi^*(s) = \arg\max_a Q^*(s,a)$ "Policy Extraction"

Online

Before Execution: nothing

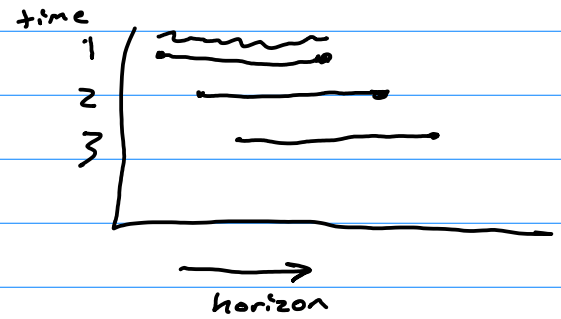
During : Everything — Consider Actions and Consequences

$|S|$ too big

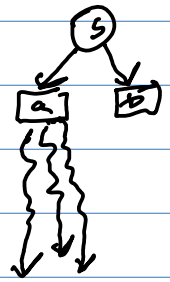
Only consider states that are reachable before γ^t gets too small

Receding Horizon

loop
- measure state s
- Calculate $a = \pi(s)$
- take a



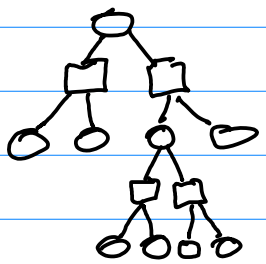
Lookahead with Rollouts



$R(s,a) + \gamma \hat{V}(s')$

$\hat{V}(s')$ = rollout with π_0

Tree Search



$\sum_{s'} T(s'|s,a) V(s')$

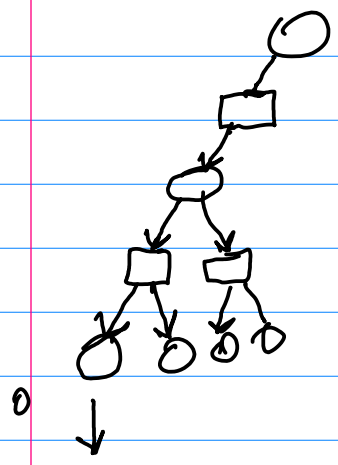
$\leftarrow Q$

$\leftarrow V(s) = \max Q(s,a)$

$\leftarrow Q(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim T(s,a)}[V(s')]$

$\leftarrow V$

Forward Search



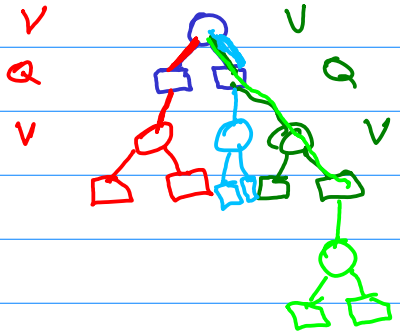
$O((|S| \times |A|)^d)$

Algorithm 4.6 Forward search

```
function SelectAction(s, d)
    if d = 0
        return (NIL, 0)
    (a*, v*) ← (NIL, −∞)
    for a ∈ A(s)
        v ← R(s, a)
        for s' ∈ S(s, a)
            (a', v') ← SelectAction(s', d − 1)
            v ← v + γ T(s' | s, a) v'
        if v > v*
            (a*, v*) ← (a, v)
    return (a*, v*)
```

Monte Carlo Tree Search (MCTS) (UCT)

Search Expand Rollout Backup




```
function simulate!(π::MonteCarloTreeSearch, s, d=π.d)
    if d ≤ 0
        return 0.0
    end
    𝒫, N, Q, c = π.𝒫, π.N, π.Q, π.c
    𝒜, TR, γ = 𝒫.𝒜, 𝒫.TR, 𝒫.γ
    if !haskey(N, (s, first(𝒜)))
        for a in 𝒜
            N[(s,a)] = 0
            Q[(s,a)] = 0.0
        end
        return rollout(𝒫, s, π.π, d)
    end
    a = explore(π, s)
    s', r = TR(s,a)
    q = r + γ*simulate!(π, s', d-1)
    N[(s,a)] += 1
    Q[(s,a)] += (q-Q[(s,a)])/N[(s,a)]
    return q
end
```

Expand ← R.O.

search

Backup

$$Q(s,a) + \underbrace{c\sqrt{\frac{\log N(s)}{N(s,a)}}}_{\text{Exploration Bonus}}$$

low $N(s,a)/N(s)$ → larg Exp Bonus