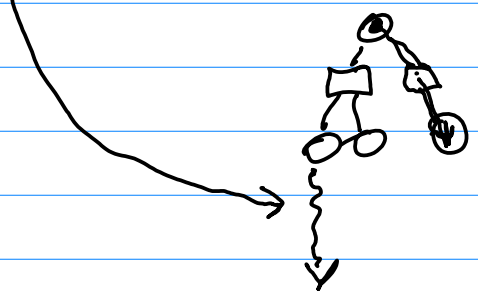
MCTS

POMDPs.simulate

simulate!



Last Time

Continuous S, A

LQR - Linear Dynamics Quad. Reward

This time

More Continuous S, A

Reinforcement Learning

Model Based Tabular

Offline

- ADP

Online

- MPC
- Sparse Tree/Progressive Widening

ADP Value Function Approx.

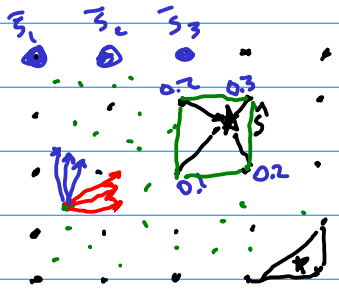
$$V(s) = \theta^T \beta(s)$$

features

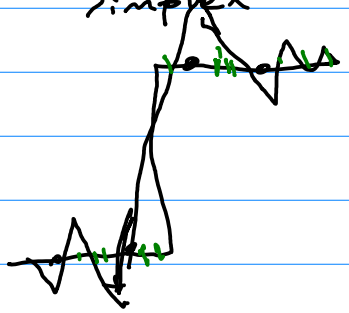
Local

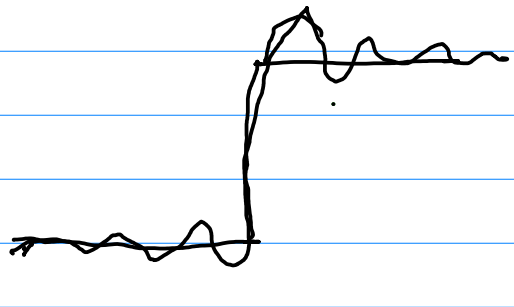
- Calculate value $\tilde{s}$
- Interpolate Between

value at $\tilde{s}$: $\theta^T \beta(\tilde{s})$



Multilinear Simplex 2^d $d+1$

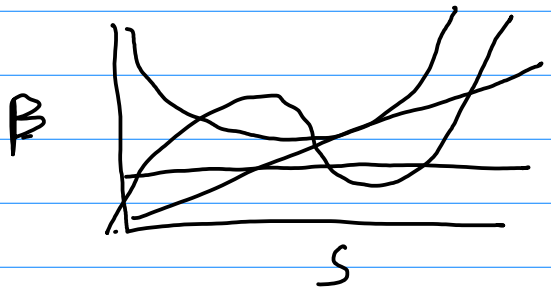




Global

$\theta^T \beta(s)$

- Fourier
- Polynomial



LSVI

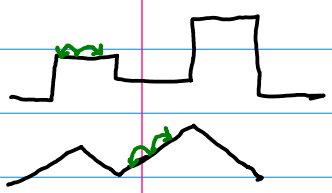
$\theta \leftarrow 0$

loop

To get rid of Gibbs in local value approx choose $\tilde{s}$ to be on interpolation points

for $\tilde{s}_i$ in sampled states

$$\tilde{v}_i \leftarrow \max_a \left(R(\tilde{s}_i, a) + \gamma \frac{1}{m} \sum_{j=1}^{m} \theta^T \beta\left(G(\tilde{s}_i, a, w_j)\right) \right)$$

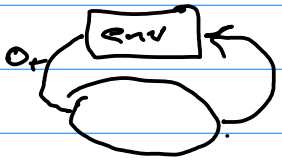


$\theta \leftarrow \text{Regress}(\tilde{s}, \beta, \tilde{v})$

→ MPC

Open Loop → Horizon

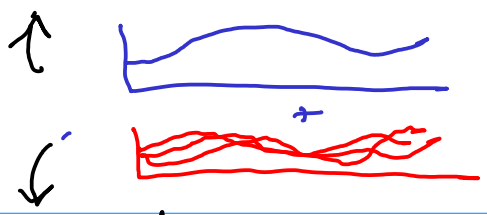
Differentiable Dynamic



$$\underset{a_{1:d}\ s_{2:d}}{\text{maximize}} \sum_{t=1}^{d} \gamma^t R(s_t, a_t)$$

Subject to $s_{t+1} = \text{mean}\left(T(s_t, a_t)\right) \quad \forall t$

off the shelf solver

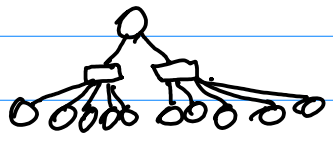


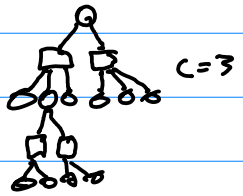
$$\underset{a_{1:d}}{\text{maximize}} \quad \sum_{t=1}^{d} \sum_{i=1}^{m} \gamma^t R(s_t^i, a_t)$$

$$\text{subject to} \quad s_{t+1}^i = G(s_t^i, a_t, w_t^i) \quad \forall_t$$

Sparse Tree Search

continuous S, discrete A





$C = S$ ∞

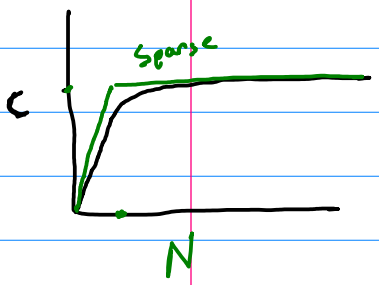
DPW

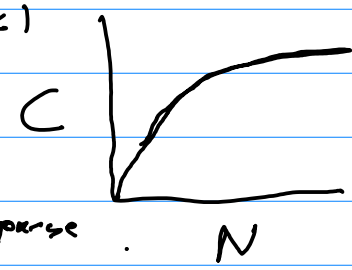
Progressive Widening

cont. S, A

$k = 8$ $\alpha = \frac{1}{30}$

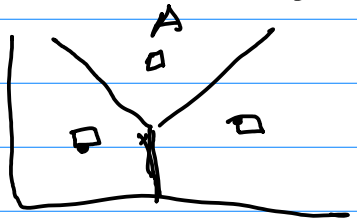
add new branch if $\cdot C < kN^\alpha$ $\alpha < 1$





For states: ends up being close to sparse

For action: need a good heuristic Voronoi

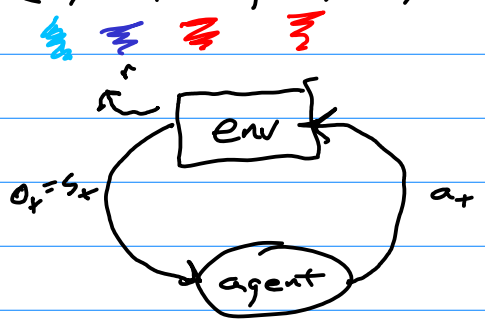


Reinforcement Learning

Course

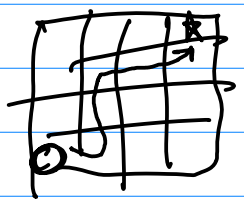
Immediate vs Future Rewards ✓
Unknown Models ←
Partial Observability
Other Agents

(S, A, T, R, γ)




```
actions(env)
reset!(env)
s', r, done ← step!(env, a)
```

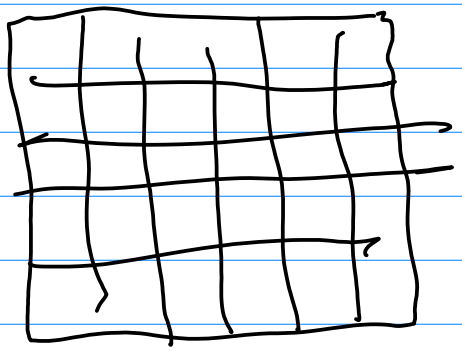
$s', r \leftarrow G(s, a, w)$



Breakout Rooms

S, A

How would you solve this problem



→ Genetic Algorithm
$\pi(s) = f(\theta, s)$
$\pi[s]$

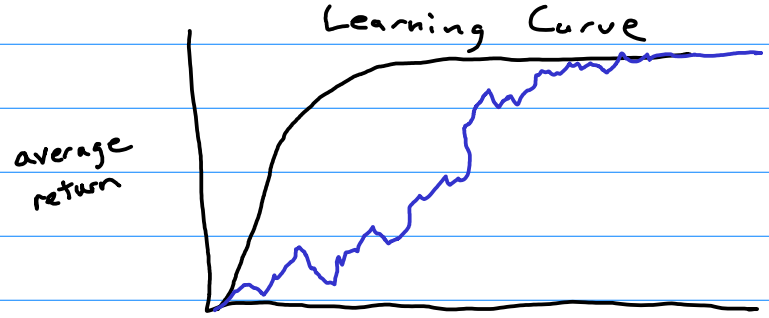
→ Learn Q values
gradually take fewer random actions

→ Understand R
trade off exploration/exploit

→ Q-learning
-Thompson Sampling
T, R

Challenges

1. Exploitation vs Explore
2. Credit Assignment
3. Generalization



Episode: starting state → terminal state

Learn Q, π

Model Based / Model Free

↑ Learn T, R

Tabular Max. Likelihood Model-Based RL

$N[a][s,s']$

$N \leftarrow$ zeros

$\rho \leftarrow$ zeros $\quad \rho[a][s]$

$s \leftarrow s_0$

$\pi \leftarrow$ random policy

loop

$a \leftarrow$ rand(A) w.p. ϵ , $\pi(s)$ o.w.

$s', r \leftarrow$ step!(env, a)

$N[a][s,s'] \mathrel{+}= 1$

$\rho[a][s] \mathrel{+}= r$

M.L. T, R {

$$T[a][s,s'] \leftarrow \frac{N[a][s,s']}{\sum_{s'} N[a][s,s']} \quad \forall\, s,a,s'$$

$$R[a][s] \leftarrow \frac{\rho[a][s]}{\sum_{s'} N[a][s,s']}$$

π {

$\pi \leftarrow$ solve (T, R)

$s \leftarrow s'$