Recitation:
RL, Projects

RL Types

- **Model based**
  - Explicitly learn $p(s_{t+1}|s_t, a_t)$, $r(s_t, a_t)$
  - Markov Decision Process (MDP) or POMDP
- **Model free**
  - Learn expected value of each state, $V(s_t)$, *given a policy*
  - Learn expected value of each state and action, $Q(s_t, a_t)$
  - Learn an optimal policy, while learning V or Q
    - Can learn on- and off-policy

State can be discrete or real, *V* and *Q* can be neural nets

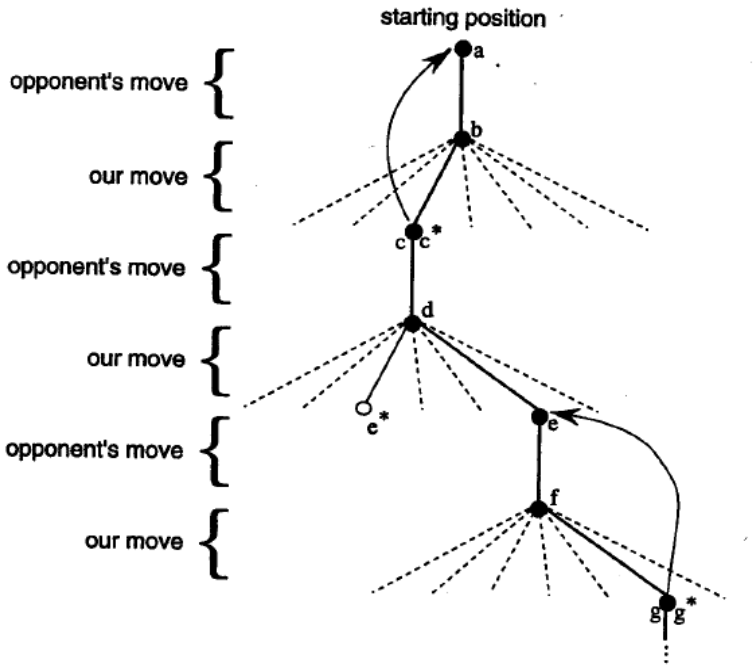
Notation summary

- s_t **state**
- $V(s_t)$ **value**
- $a_t = \pi(s_t)$ **action (and policy π)**
- γ **discount factor**
- $r(s_t,a_t,s_{t+1})$ **reward** (usually simply $r(s_t)=R_t$)
- G_t **expected discounted reward ('return')**
- $p(s_{t+1}|s_t,a_t)$ **model**

What is the relationship between $V(s_t)$ and G_t?

RL vs. search

People used to solve go and chess by search, but now solve it by RL. Why?



RL vs. Supervised Learning

One could do supervised learning of a value function $V(s)$ for a game state s by recording for each game state s who won.

Why do reinforcement learning instead?

TD(0)

One could do RL of V(s) **by updating** V(s) **at the end of each game based on who won.**

Why update *V(s)* as soon as one makes a move and sees the opponent's response?

Model based vs. Model free

- One can learn a model of the world $p(s_{t+1}|s_t,a_t)$ and use that to find an optimal policy
- Or one can learn the value $V(s_t)$ of each state--or of the action in each state $Q(s_t,a_t)$ --without a world model
- When is each better?

What is the key assumption in both cases?

A problem?

- $V(s)$ **depends on** π
- **But** π **is a function of** $V(s)$

A 0.812	B 0.868	C 0.918	Food 1.00
D 0.762		E 0.660	Shock -1.00
J 0.705	G 0.655	H 0.611	I 0.388

So how can we learn $V(s)$ **and** π^***?**

Bellman Equation (Q version)

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q(s',a)$$

Q-Learning

$$Q(s,a) := r(s,a) + \gamma \max_{a} Q(s',a)$$

We still need to figure out how to adjust the policy - on policy or off policy

Deep Q-Learning (DQL)

$$\mathit{Argmin}_{\theta} \left[Q(s,a;\theta) - \left(r(s,a) + \gamma \max_{a} Q(s',a;\theta) \right) \right]^2$$

Represent Q with a neural net

Projects

- You don’t need to do original research
- Start dumb
- Look at the data!!!