Session 3

Reward and Policy

Markov Decision Process

- States - s
- Actions - a(s), a
- Model - T(s,a,s') ~ Pr(s'|s,a)
- Rewards - R(a), R(s), R(s,a)

- Policy - π(s) -> a

Markov Property

Only the present matters.

Pr(s’|s,a) = Pr(s’|s1,s2,s3,s4…..sn , a)

Assumption: Things are stationary. Rules don’t change with respect to time. Dealing with a time changing world is a little more complex.

What if a state isn’t Markov? You can always fold history into a state and make it Markov.

What we need? A solution. A policy.

- A policy that maps $\pi(s) \rightarrow a$. It says take this action, it is a command.
- π^* is the optimal policy that maximises the long term reward.
- It isn't a plan. Instead it queries one step at a time.

A bit more on Reward Dynamics.

- Rewards are
    - Delayed.
    - Minor changes matter.
- Zero reward at a point. But at the end of the game you win.
- Rewards are the critics for your action.

Difference between supervised and reinforcement Learning.

- Play a game of chess.
- One bad move at turn 3. Then you play a beautiful game, but still lose.

- In supervised, the algorithm can’t learn which one of these moves caused your downfall. Just learns the sequence caused it. In reinforcement learning, the algorithm is able to pick where you erred.
- Credit assignment problems.

Policy

- A policy is the agent's **behaviour**
- It is a map from **state** to **action**, e.g.
- **Deterministic** policy: $a = \pi(s)$
- **Stochastic** policy: $\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$

Grid world Example

- R(s) = 0
- R(s) = -.04
- R(s) = -2
- R(s) = +2

All rewards are for non-terminal states. The blocked state cannot be reached. The goal state has a reward +1. The trap state has a reward -1. The game does not end until one of the terminal states are reached.

Assumptions

- World is stationary. The physics (rules) of the world does not change with time.
- Infinite Horizons. Game won't end until I hit a terminal state.
    - If the game isn't an 'infinite horizons' game, the policy will change with respect to time. The game will get more aggressive if we said you have only 3 turns left.
- Utility of sequences is maintained.
    - If U(s0,s1,s2,s3,s4……) > U(s0,s1',s2',s3',………)
    - Then U(s1,s2,s3,s4) > U(s1',s2',s3'………)
    - Also called as stationarity of preference. If I prefer something today, I will prefer the same tomorrow.

Is utility the sum of rewards?

U(s0,s1,s2,......) = sigma(t=0:infinity) {R(s_t)}

The above gives a result that is highly "unusable."

An infinite series of positive rewards will give you the same utility (infinity), even though some sequence is better than the other.

New definition

U(s0,s1,s2,......) =

sigma(t=0:infinity) {<gamma>**t * R(s_t)}

0 <= {<gamma>} < 1

By tuning gamma, we can set how far into the future we want to see.

Optimum Policy

$$\pi^* = \underset{\pi}{\arg\max}\; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, \pi\right]$$

Policy which when followed, the Expectation of rewards is maximum.

A new definition of $U^{pie}(s)$

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \,\middle|\, \pi,\ s_0 = s\right]$$

The utility of the state is the expectation of the reward, provided you start from that state and follow the policy <pie>. So utility is now policy dependent.

So what is the optimum policy

$$\pi^*(s) = \arg\max_{a} \sum_{s'} T(s,a,s')\, U^{\pi^*}(s')$$

$$U(s) = R(s) + \gamma \max_{a} \sum_{s'} T(s,a,s')\, U(s')$$

A better definition.

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')\, U(s')$$

N states, N linear equations. N unknowns. Can be solved right?

NO. The “max over a” makes this system non-linear.

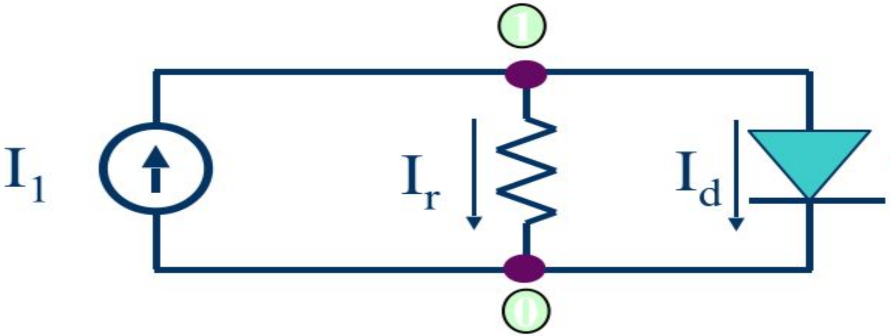
Instead we use an iterative process.

<VALUE ITERATION>

1. Start with a random utility
2. Update utility based on neighbours.
3. Repeat and Repeat
4. Wait for convergence
5. Once you know Utility, you know the policy

- Similar to gradient descent or any other supervised learning algorithm.
- Works because at each step, you add “truth to an assumption.” Eventually your assumption will become true. And the fact that <gamma> is less than one is required to prove mathematical convergence. {MATH STUFF}
- Method is called as “VALUE ITERATION”

DC Analysis of Nonlinear Circuits - Example



$$I_d - I_s(e^{V_d/V_t} - 1) = 0$$

Need to Solve

$$\boxed{I_r + I_d - I_1 = 0}$$

$$\frac{1}{R}e_1 + I_s(e^{\frac{e_1}{V_t}} - 1) - I_1 = 0 \quad \Rightarrow \quad g(e_1) = I_1$$

How to find policies directly?

1. Start with a random policy π_0
2. Given π_t calculate $U_t = U^{<pie>t}$. What we evaluate is how good that policy is.
3. Improve π_t to get π_{t+1}. Recompute Utility for your new policy.
4. Rinse and repeat.
5. For more information about this: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/kaelbling96a-html/node20.html