RL Week-4

L1 DP : Value Iteration

- Turning the bellman optimality eqn into an update rule.

$$v_*(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a\right]$$

$$= \max_a \sum_{s', r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

∴, write it as in terms of k.

$$v_{k+1}(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a\right]$$

$$= \max_a \sum_{s', r} p(s', r \mid s, a)\left[r + \gamma v_k(s')\right]$$

$v_0(s) = 0 \quad \forall s$

$v_1(s) =$ is calculated $v_0(s)$

$v_2(s) =$ comes from $v_1(s)$

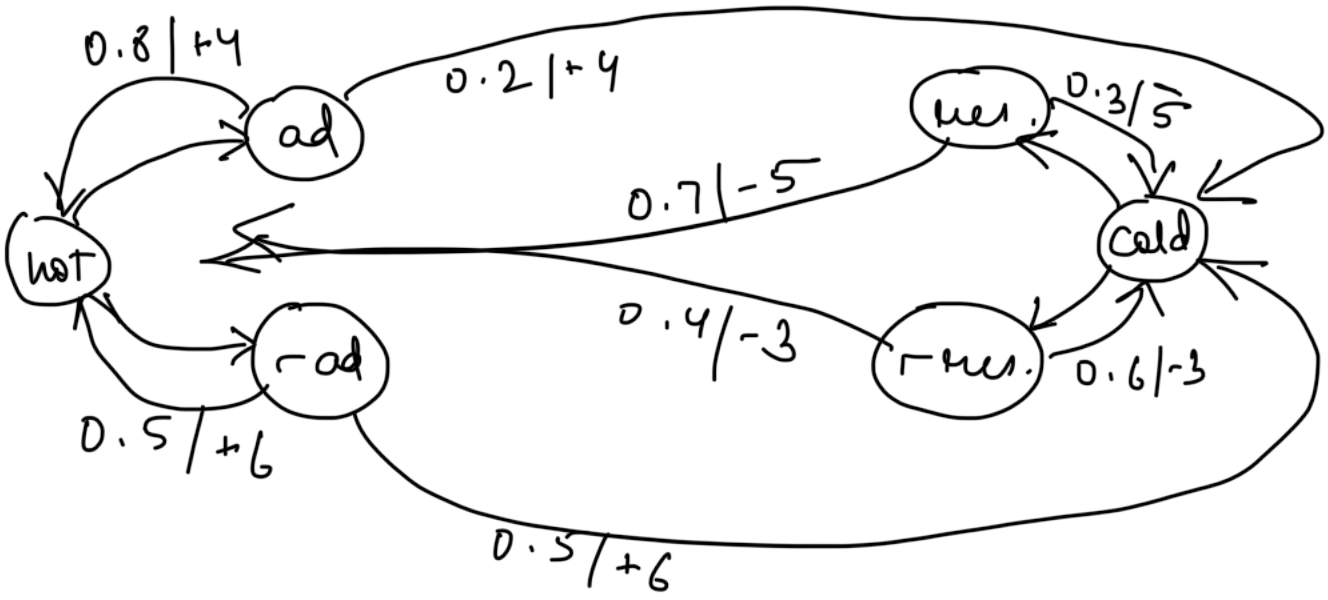
- a small threshold $\theta > 0$, that determines estimates accuracy.

Smpt. → $V(\text{terminal}) = 0$

- output of a deterministic policy, $\pi \approx \pi_*$

$$\pi(s) = \arg\max_a \sum_{s', r} p(s', r \mid s, a)\left[r + \gamma V(s')\right.$$

- going back to manufacturer problem MDP.



$V(\text{hot}) = V(\text{cold}) = 0$ [Initialise]

$A(\text{hot}) = \{ad, \neg ad\}$

$A(\text{cold}) = \{\text{res}, \neg \text{res}\}$

$v(\text{hot}) = 6$, $v(\text{cold}) = -3$

(highest of all reward in both cases)

L2 Value Iteration

$$v_{k+1}(s) = v_{\pi}^{(1)}(s)$$

further demo, convergence happens after many iterations.

L3 More on DP

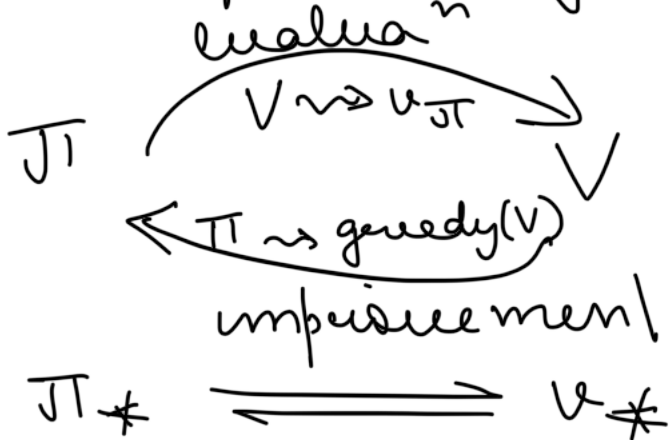
- asychronous DP

1. Old algos. → we had to do the updates over entire state set. (Problem)

2. in async. DP → the updates are not done for entire state set for each episode.
3. in async. DP, we just have to visit every state so enough that it reaches convergence.
4. in this way, order of the update can be chosen in a more flexible fashion.
5. interestingly, it takes into account real time updates from envt. & DP.
6. it helps you to update those parts which are specifically agent ~ space.

- if policy evaluation is stopped after 1 update, then we have to apply greedification to that particular state. We get value iteration.

- GPI (Generalized Policy Iteration)
↳ in this, we allow policy evaluation & improvement interact with each other, independently.



- almost RL methods are kind of GPI.
Eg. Policy iteration runs the evaluation toward completion, after this improvement starts. In value iteration, 1 step of evaluation is done before improvement. But in GPI, async DP, the two methods work together.

L4 MC (Monte Carlo) Methods

- as per DP,

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]$$

- as per MC RL,

$$v_\pi(s) = E_\pi\left[G_t \mid S_t = s\right] \quad \forall s \in S$$

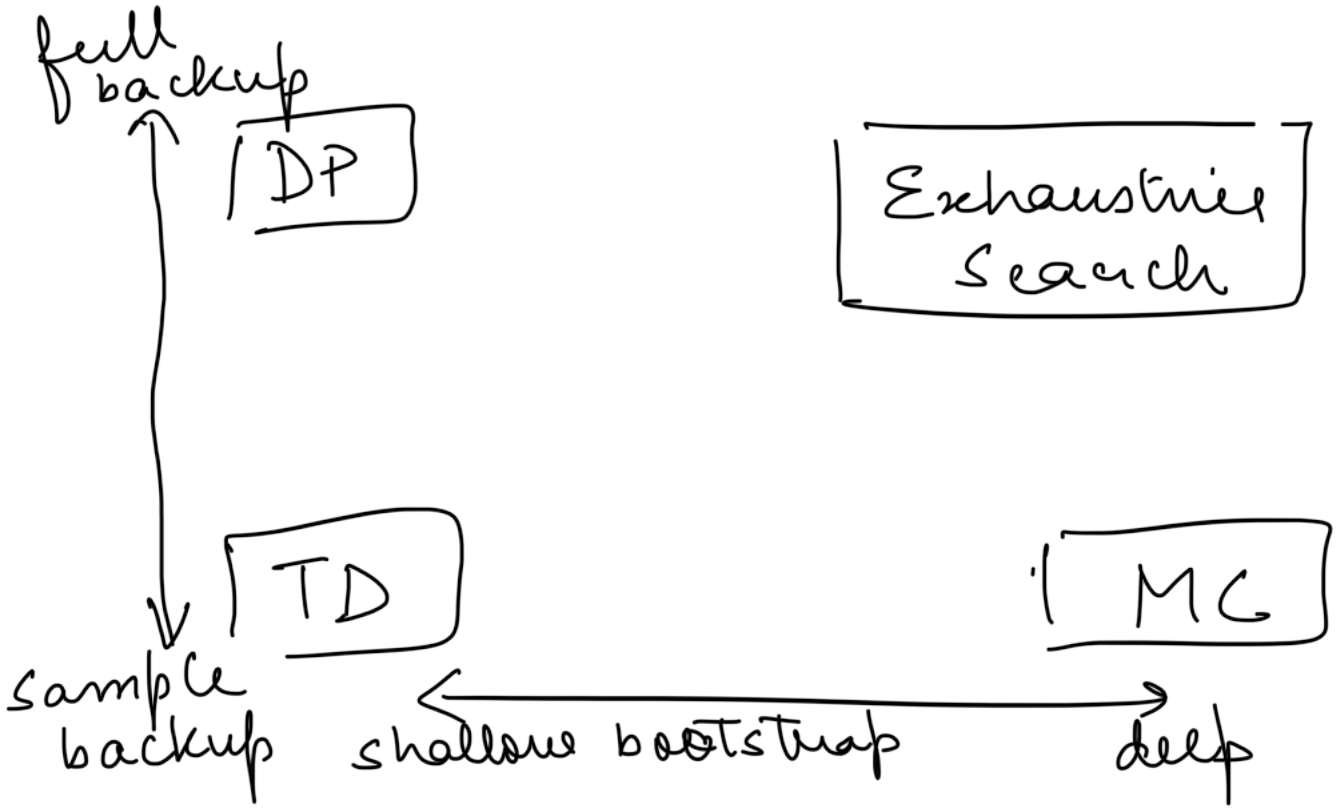
1. learning directly from sample episodes of experience.
2. not req. (full model) and is model free.
3. no need of bootstrapping
4. value func^n is calculated as the mean of discounted returns. (G_t)

- MC Predic^{ns} —
  1. 1st visit → V(s) is the avg. of returns, first visit to s.

2. every visit → returns the avg. following all visits to s.

L5 Comparison of DP, MC & TD

- Bootstrapping
  ↳ update using an estimate
  1. DP bootstraps
  2. MC does not bootstrap
  3. TD bootstrap
- Sampling → update calculated using samples w/o model.
  1. MC samples & TD
  2. DP does not sample.



Useful table —

	Bootstrap	Sampling
1. DP	✓	X
2. MC	X	✓
3. TD	✓	✓