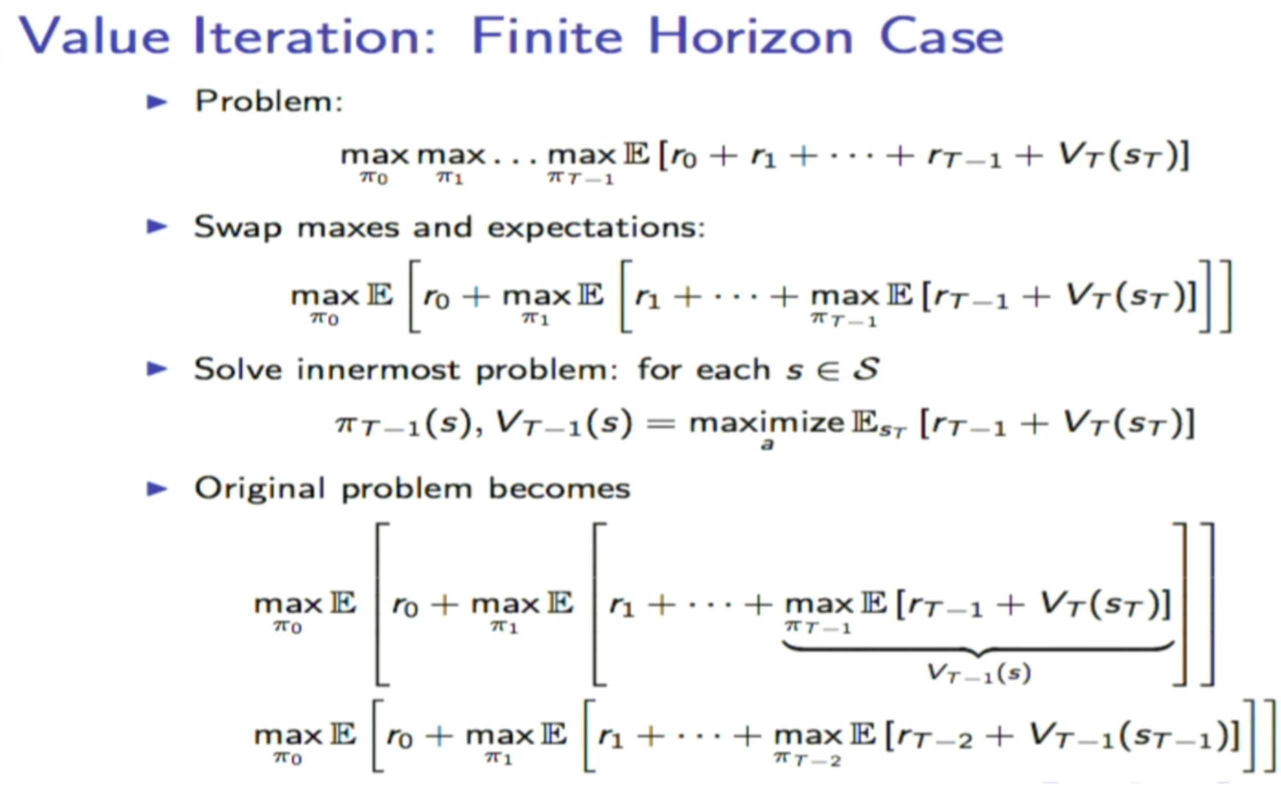
MDPs, value and policy iteration

- this is the basic formalisms of reinforcement learning in simple analytical, discrete settings.
- value and policy iteration will be the foundations for Q-learning and policy gradient methods respectively
- refer to the notes on reinforcement learning in the intro machine learning (cs 229) notes for a primer on the basic mathematical formulation here and value and policy iteration algorithms.
- An MDP can be trivially mapped onto a POMDP (by making observations = state). this is the difference between fully observable and partially observable environments.
- A POMDP can also be mapped onto a MDP by making each state represent the history of all states (this leads to an explosion in number of states). This is kinda what RNNs do by trying to remember everything that matters in a vector.
- Other than the discounted (γ-based) reward we saw in our intro RL lecture from cs 229, john schulman introduces a other kind of reward: finite horizon rewards, which don't necessarily have to have discount factors → they don't have to bound the reward since they only go for a finite number of steps.
- in the finite horizon setting, the rewards are not stationary anymore → reward function is also time dependent.
- value iteration in the finite horizon case:



Value Iteration: Finite Horizon Case

- Problem:

$$\max_{\pi_0}\max_{\pi_1}\dots\max_{\pi_{T-1}}\mathbb{E}\left[r_0+r_1+\cdots+r_{T-1}+V_T(s_T)\right]$$

- Swap maxes and expectations:

$$\max_{\pi_0}\mathbb{E}\left[r_0+\max_{\pi_1}\mathbb{E}\left[r_1+\cdots+\max_{\pi_{T-1}}\mathbb{E}\left[r_{T-1}+V_T(s_T)\right]\right]\right]$$

- Solve innermost problem: for each $s\in\mathcal{S}$

$$\pi_{T-1}(s), V_{T-1}(s) = \operatorname*{maximize}_{a}\mathbb{E}_{s_T}\left[r_{T-1}+V_T(s_T)\right]$$

- Original problem becomes

$$\max_{\pi_0}\mathbb{E}\left[r_0+\max_{\pi_1}\mathbb{E}\left[r_1+\cdots+\underbrace{\max_{\pi_{T-1}}\mathbb{E}\left[r_{T-1}+V_T(s_T)\right]}_{V_{T-1}(s)}\right]\right]$$

$$\max_{\pi_0}\mathbb{E}\left[r_0+\max_{\pi_1}\mathbb{E}\left[r_1+\cdots+\max_{\pi_{T-2}}\mathbb{E}\left[r_{T-2}+V_{T-1}(s_{T-1})\right]\right]\right]$$

- Notice the similarity in problem statement of LQR and value iteration! in both cases, we solve the innermost problem first and then backup until we get to finding the best action at $t = 0$
- note that the above is strictly for the finite horizon case only. the way we solved it only works for finite horizon case.

- for the infinite case we still need the γ. but we can't work backwards like above since there are an infinite number of timesteps.
- for the infinite case, the algorithms for value and policy iteration and proof of convergence is given in the RL notes for cs 229 class notes. since there are a finite number of policies, the policy will eventually converge to the optimal policy.
- can we combine the benefits of the cheap update step in value iteration (not having to solve a linear system) with the benefit of fewer iterations in policy iteration?
- Yes! Modified policy iteration. Remember that in value iteration we just did a $k = 1$ look ahead which made the update step very fast. in policy iteration we looked all the way ahead and solved the system of linear equation exactly to get the value for each state based on current policy. In modified policy iteration, we look k steps ahead (k bellman backup updates), where k is some small integer. $k = 1$gives value iteration and $k = \infty$ gives policy iteration. Shulman doesn't prove why modified policy iteration works, says the proof is not so simple as for value and policy iteration ¯_(ツ)_/¯.