Dynamic Programming

Optimal Substructure

The Optimal Solution to **all subproblems** produces the optimal solution to the original problem. In other words, if you find the **optimal policy for all states** in a problem, you find the optimal policy for **the whole problem**.

Thereforce, DP turns Bellman equations into Update Rules:

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

$v(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v(s')]$

You need to know in advance how the state changes and what rewards we get from performing each action in each state: $p(s', r \mid s, a)$.

Value Iteration

Update v iteratively to get better approximation of optimal value at every iteration:

$v(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v(s')]$

where $v(s')$ an old value of this current value table for the next state.

In value iteration, you change the policy only once at the end of all the iterations. Throughout the iterations, the policy is always constant (usually random uniform), and for every iteration, you evaluate the value for **every** action.

Policy Iteration

While value iteration iterates through all actions evenly, policy iteration iterates values and then update the policy (Policy Improvement) to carry out another "value iteration" using the newly set policy (Policy Evaluation):

$v(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v(s')]$

You evaluate for only **one** action that is given by your current policy, and thus the $\sum_a \pi(a \mid s)$ expression instead of $\max_a$.

In policy iteration, you change the policy after every value iteration, so the policy is being updated (Improvement) across all the value iterations (Evaluation).

Policy Improvement Theorem

If the new value of the state after choosing action with new policy, $q_\pi(s, \pi'(s))$, is greater than the original value, $v_\pi(s)$, then the new state value, $v_{\pi'}(s)$ is greater than original state value, $v_\pi(s)$:

$q_\pi(s, \pi'(s)) \geq v_\pi(s) \Rightarrow v_{\pi'}(s) \geq v_\pi(s)$

where $q_\pi(s, a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma v(s')]$.

Then, the policy is updated to the new argument (action) that gives better value for the state:

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