Reinforcement Learning Algorithms

Policy Iteration

Second step towards optimal policy…

Assumptions:

1. Reward at every state is known.
2. Transition probability is known.

In a way, the dynamics of the environment is completely known.

1 - Policy Evaluation

$$V_{i+1}^{\pi_k}(s) = \sum_{s'} P(s'|s,\pi(s))[r + \gamma V_i^{\pi_k}(s')]$$

2 - Policy Improvement

$$\pi_{k+1}(s) \leftarrow \underset{a}{\text{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$$

Algorithm:

Initialize $V(s) \in \mathbb{R}$ & $\pi(s) \in A(s)$ $\forall s$

Policy Evaluation

For each 's' in S:

$V(s) \leftarrow \sum_{s',r} P(s'|s,\pi(s))[r + \gamma V(s')]$

Policy Improvement

For each 's' in S:

$\pi_{old}(s) \leftarrow \pi(s)$

$\pi(s) \leftarrow \text{argmax}_a \sum_{s',r} P(s'|s,a)[r + \gamma V(s')]$

if $\pi(s) = \pi_{old}(s)$, then stop

The policy iteration evaluates one policy at a time which may not sweep through all the states. Therefore, to sweep through all the state there is a requirement of multiple policy evaluation.

https://www.profrkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

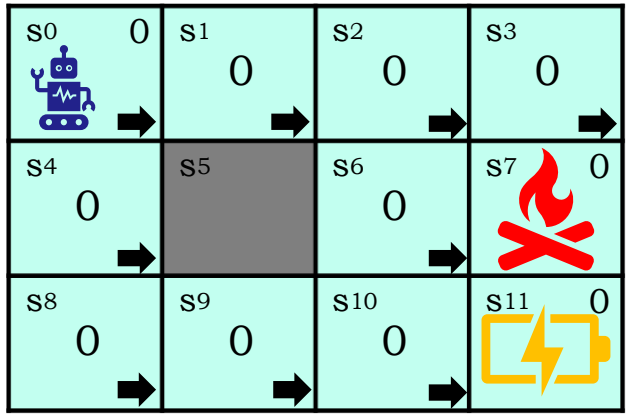
Policy Evaluation

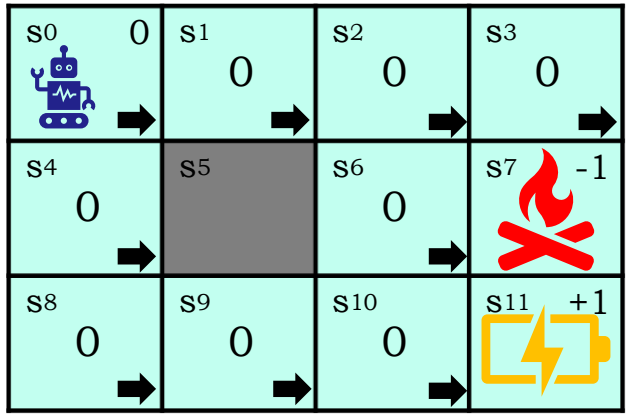
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

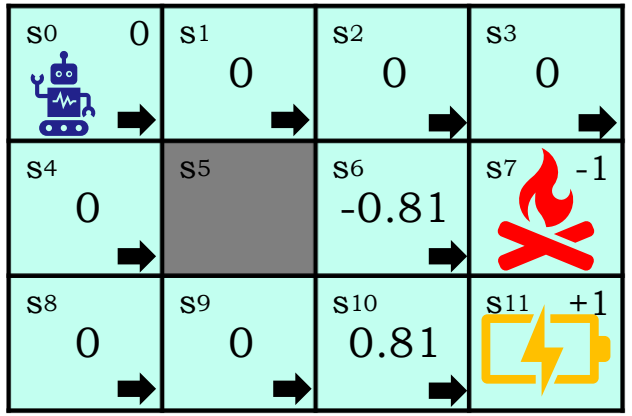
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_1(s_{10}) = 0.9[0 + 0.9\times 1] + 0.1\times[0 + 0.9\times 0] = 0.81$
  - $V^{\pi_1}_1(s_6) = 0.9[0 + 0.9\times -1] + 0.1\times[0 + 0.9\times 0] = -0.81$







https://www.profrkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

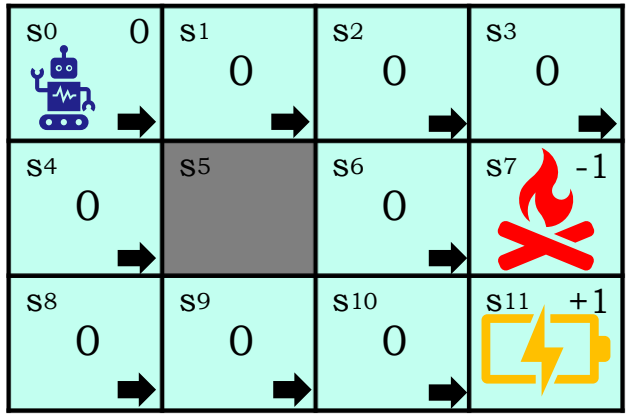
Policy Evaluation

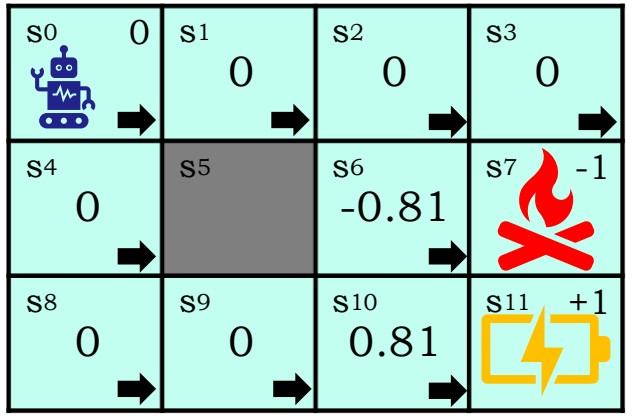
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

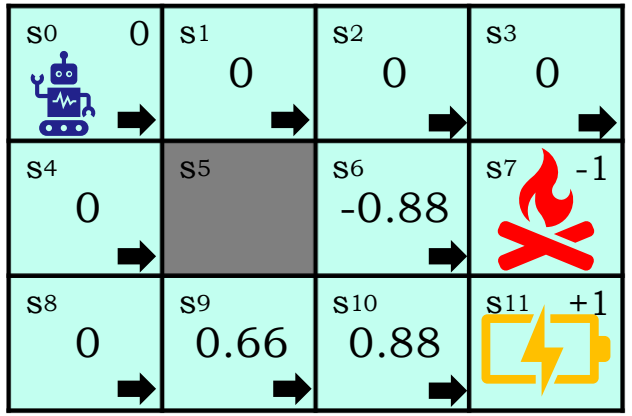
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_1(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0] = 0.81$
  - $V^{\pi_1}_1(s_6) = 0.9[0 + 0.9\times-1] + 0.1\times[0 + 0.9\times0] = -0.81$
  - $V^{\pi_1}_2(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.81] = 0.88$
  - $V^{\pi_1}_2(s_6) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times-0.81] = -0.88$
  - $V^{\pi_1}_2(s_9) = 0.9[0 + 0.9\times0.81] + 0.1\times[0 + 0.9\times0] = 0.6561$







https://www.pratikkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

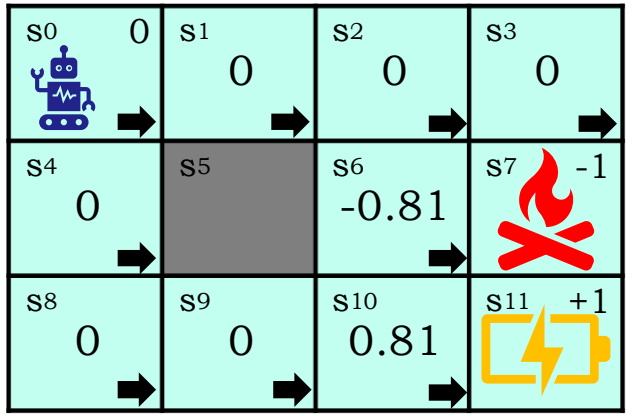
Policy Evaluation

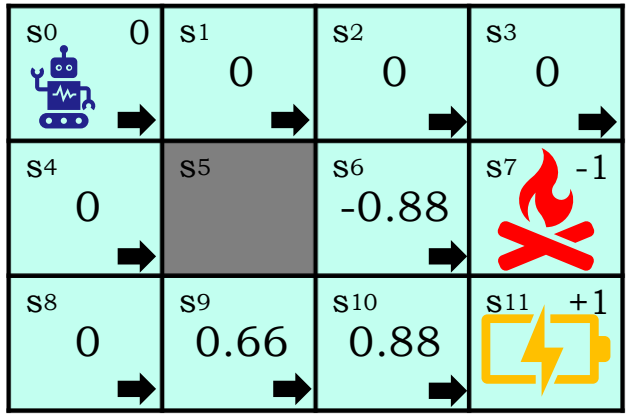
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

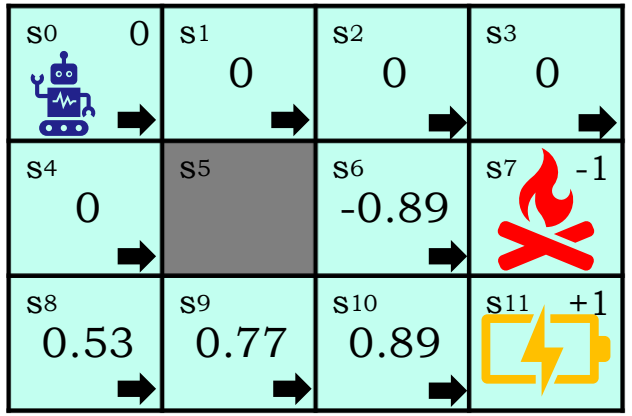
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_2(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.81] = 0.88$
  - $V^{\pi_1}_2(s_6) = 0.9[0 + 0.9\times-1] + 0.1\times[0 + 0.9\times-0.81] = -0.88$
  - $V^{\pi_1}_2(s_9) = 0.9[0 + 0.9\times0.81] + 0.1\times[0 + 0.9\times0] = 0.6561$
  - $V^{\pi_1}_3(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.88] = 0.89$
  - $V^{\pi_1}_3(s_6) = 0.9[0 + 0.9\times-0.88] + 0.1\times[0 + 0.9\times-0.88] = -0.89$
  - $V^{\pi_1}_3(s_9) = 0.9[0 + 0.9\times0.88] + 0.1\times[0 + 0.9\times0.66] = 0.77$
  - $V^{\pi_1}_3(s_8) = 0.9[0 + 0.9\times0.66] + 0.1\times[0 + 0.9\times0] = 0.53$







https://www.pratikkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

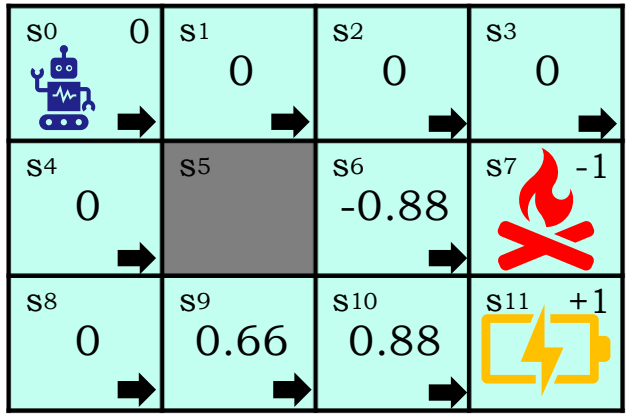
Policy Evaluation

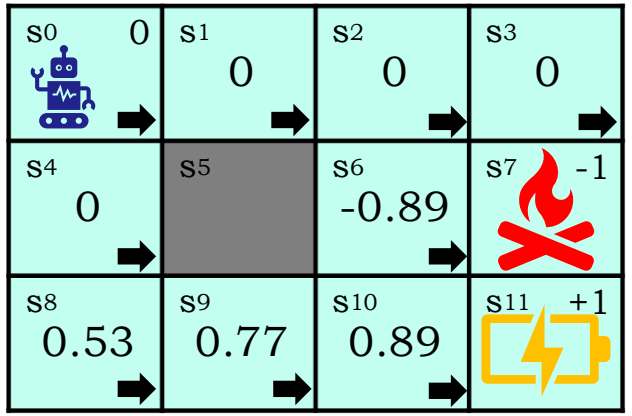
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

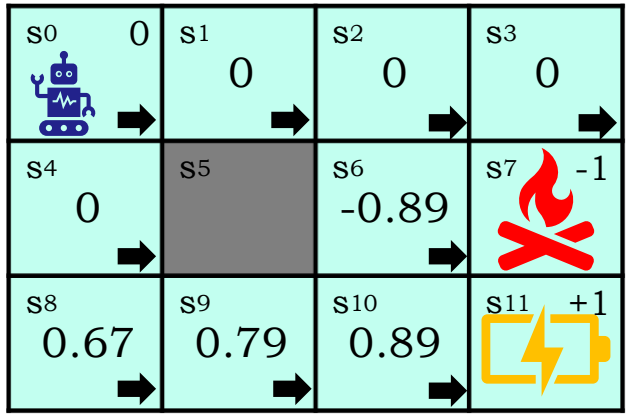
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_2(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.81] = 0.88$
  - $V^{\pi_1}_2(s_6) = 0.9[0 + 0.9\times-1] + 0.1\times[0 + 0.9\times-0.81] = -0.88$
  - $V^{\pi_1}_2(s_9) = 0.9[0 + 0.9\times0.81] + 0.1\times[0 + 0.9\times0] = 0.6561$
  - $V^{\pi_1}_3(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.88] = 0.89$
  - $V^{\pi_1}_3(s_6) = 0.9[0 + 0.9\times-0.88] + 0.1\times[0 + 0.9\times-0.88] = -0.89$
  - $V^{\pi_1}_3(s_9) = 0.9[0 + 0.9\times0.88] + 0.1\times[0 + 0.9\times0.66] = 0.77$
  - $V^{\pi_1}_3(s_8) = 0.9[0 + 0.9\times0.66] + 0.1\times[0 + 0.9\times0] = 0.53$







https://www.pratikkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

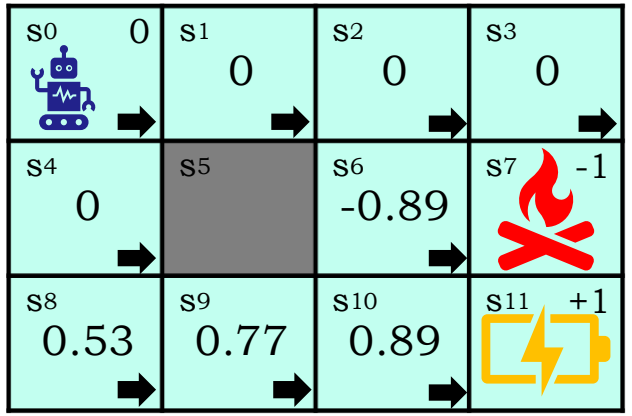
Policy Evaluation

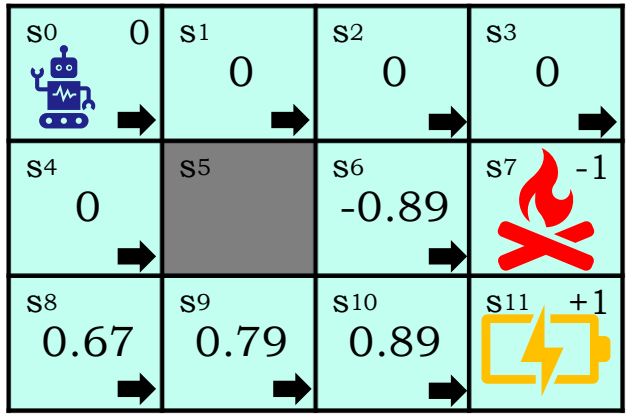
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

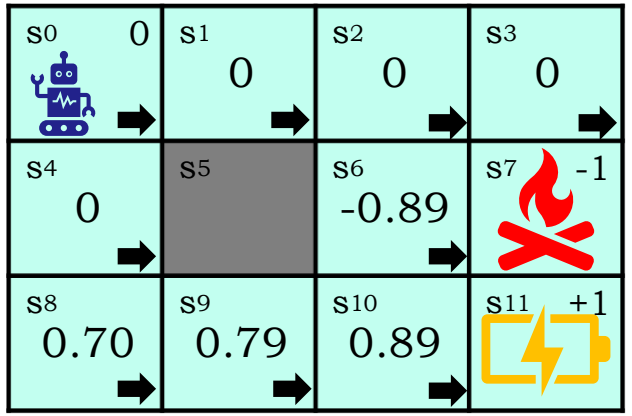
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_2(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.81] = 0.88$
  - $V^{\pi_1}_2(s_6) = 0.9[0 + 0.9\times-1] + 0.1\times[0 + 0.9\times-0.81] = -0.88$
  - $V^{\pi_1}_2(s_9) = 0.9[0 + 0.9\times0.81] + 0.1\times[0 + 0.9\times0] = 0.6561$
  - $V^{\pi_1}_3(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.88] = 0.89$
  - $V^{\pi_1}_3(s_6) = 0.9[0 + 0.9\times-0.88] + 0.1\times[0 + 0.9\times-0.88] = -0.89$
  - $V^{\pi_1}_3(s_9) = 0.9[0 + 0.9\times0.88] + 0.1\times[0 + 0.9\times0.66] = 0.77$
  - $V^{\pi_1}_3(s_8) = 0.9[0 + 0.9\times0.66] + 0.1\times[0 + 0.9\times0] = 0.53$







https://www.pratikkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s,a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a)[r + \gamma V^*_k(s')]$

$Q^*(s,a) = \sum_{s'} p(s'|s,a)\left[r + \gamma \max_{a'} Q^*(s',a')\right]$

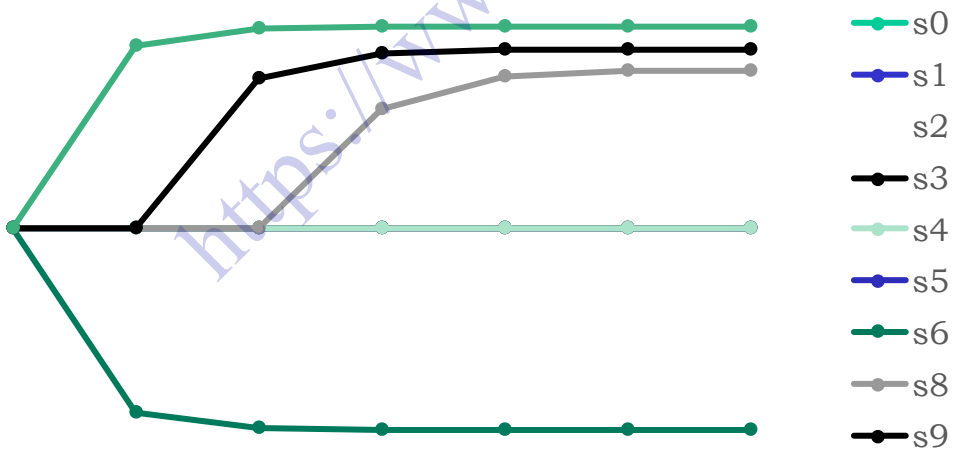
Policy Evaluation

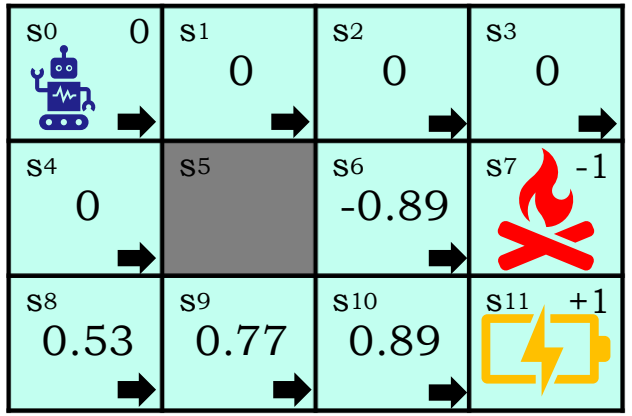
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s,\pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

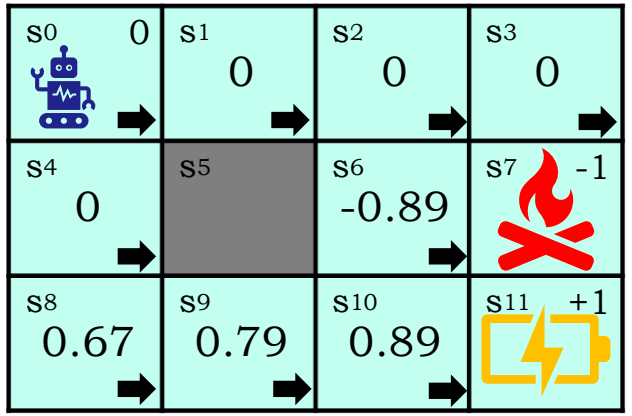
Policy Improvement

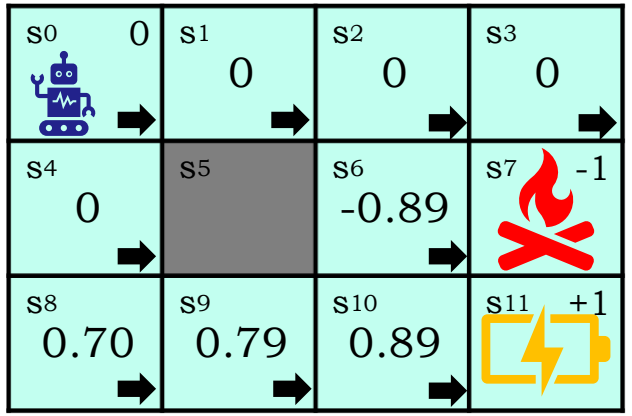
$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s,\pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$
  - $V^{\pi_1}_2(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.81] = 0.88$
  - $V^{\pi_1}_2(s_6) = 0.9[0 + 0.9\times-1] + 0.1\times[0 + 0.9\times-0.81] = -0.88$
  - $V^{\pi_1}_2(s_9) = 0.9[0 + 0.9\times0.81] + 0.1\times[0 + 0.9\times0] = 0.6561$
  - $V^{\pi_1}_3(s_{10}) = 0.9[0 + 0.9\times1] + 0.1\times[0 + 0.9\times0.88] = 0.89$
  - $V^{\pi_1}_3(s_6) = 0.9[0 + 0.9\times-0.88] + 0.1\times[0 + 0.9\times-0.88] = -0.89$
  - $V^{\pi_1}_3(s_9) = 0.9[0 + 0.9\times0.88] + 0.1\times[0 + 0.9\times0.66] = 0.77$
  - $V^{\pi_1}_3(s_8) = 0.9[0 + 0.9\times0.66] + 0.1\times[0 + 0.9\times0] = 0.53$









https://www.pratikkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s,a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a)[r + \gamma V^*_k(s')]$

$Q^*(s,a) = \sum_{s'} p(s'|s,a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

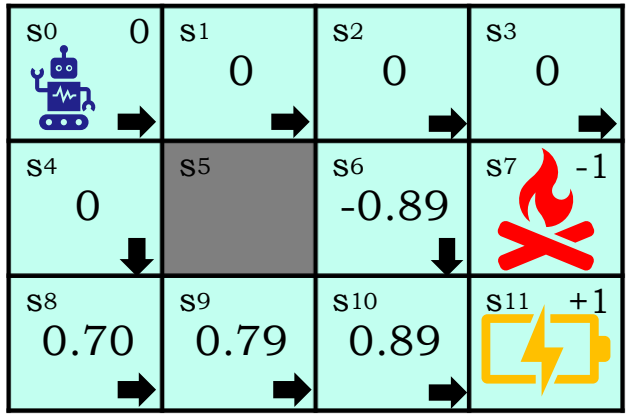
Policy Evaluation

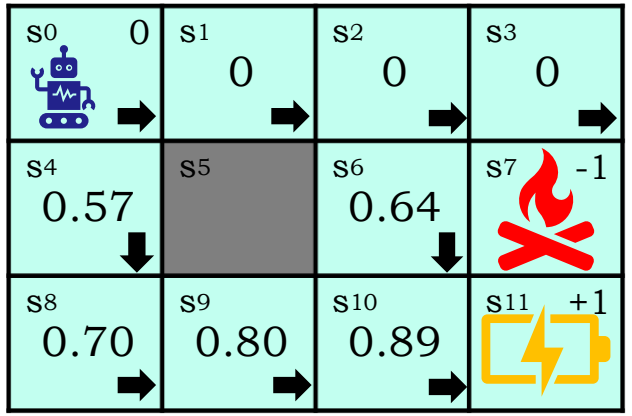
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

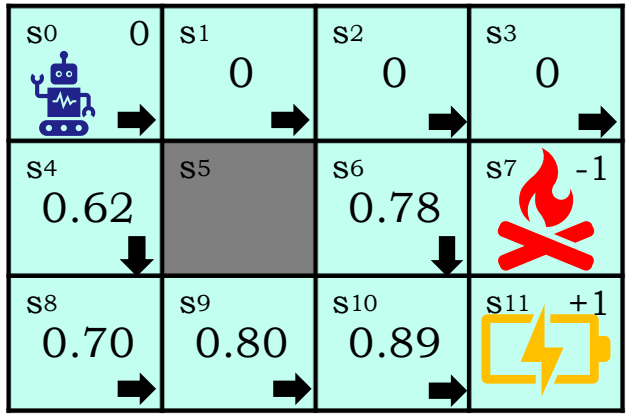
Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$
- Policy Improvement
  - $\pi_2(s) = \underset{a}{\operatorname{argmax}}\big(\sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_1}(s')]\big)$
  - $\pi_2(s^0) = \underset{a}{\operatorname{argmax}}(0_{\rightarrow}, 0_{\downarrow}) = \rightarrow/\downarrow$
  - $\pi_2(s^1) = \underset{a}{\operatorname{argmax}}(0_{\rightarrow}, 0_{\leftarrow}) = \rightarrow/\leftarrow$
  - $\pi_2(s^2) = \underset{a}{\operatorname{argmax}}(0_{\rightarrow}, 0_{\leftarrow}, -0.89_{\downarrow}) = \rightarrow/\leftarrow$
  - $\pi_2(s^3) = \underset{a}{\operatorname{argmax}}(-1_{\downarrow}, 0_{\leftarrow}) = \leftarrow$
  - $\pi_2(s^4) = \underset{a}{\operatorname{argmax}}(0_{\uparrow}, 0.7_{\downarrow}) = \downarrow$
  - $\pi_2(s^6) = \underset{a}{\operatorname{argmax}}(-1_{\rightarrow}, 0_{\uparrow}, 0.89_{\downarrow}) = \downarrow$
  - $\pi_2(s^8) = \underset{a}{\operatorname{argmax}}(0.79_{\rightarrow}, 0_{\uparrow}) = \rightarrow$
  - $\pi_2(s^9) = \underset{a}{\operatorname{argmax}}(0.89_{\rightarrow}, 0.7_{\leftarrow}) = \rightarrow$
  - $\pi_2(s^{10}) = \underset{a}{\operatorname{argmax}}(1_{\rightarrow}, 0.79_{\leftarrow}, -0.89_{\uparrow}) = \rightarrow$







http://www.prefrkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)(r + \gamma V^*(s'))$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

Policy Evaluation

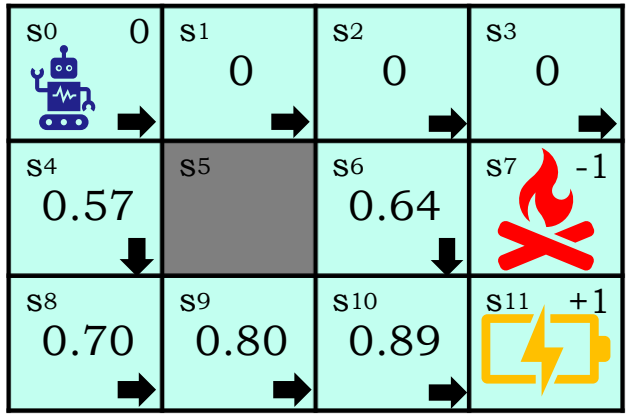
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\left[r + \gamma V^{\pi_k}_i(s')\right]$

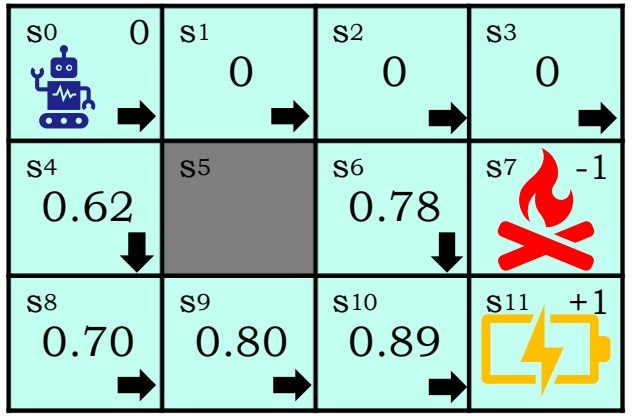
Policy Improvement

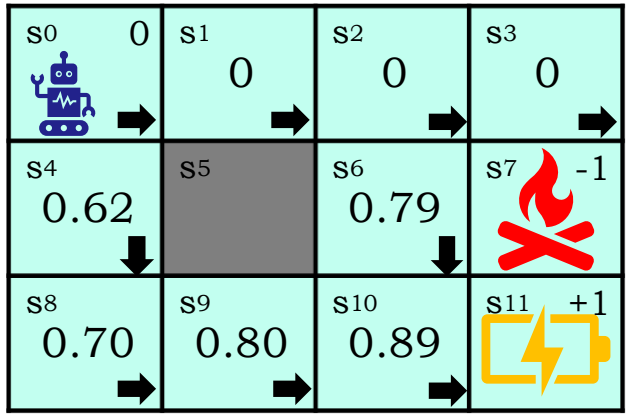
$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\left[r + \gamma V^{\pi_k}_i(s')\right]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\left[r + \gamma V^{\pi_k}_i(s')\right]$







https://www.profkumar.co.in/

Policy Iteration: Dynamic Programming

$State, s \in S$

$Action, a \in A(s)$

$Reward, r \in R(s, a, s')$

$Transition\ Probability, P(s, a, s')$

$Optimal\ Policy, \pi^*(s)$

$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$State\ value, V^{\pi}(s)$

$Action\ value, Q^{\pi}(s, a)$

$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$

$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$

Policy Evaluation

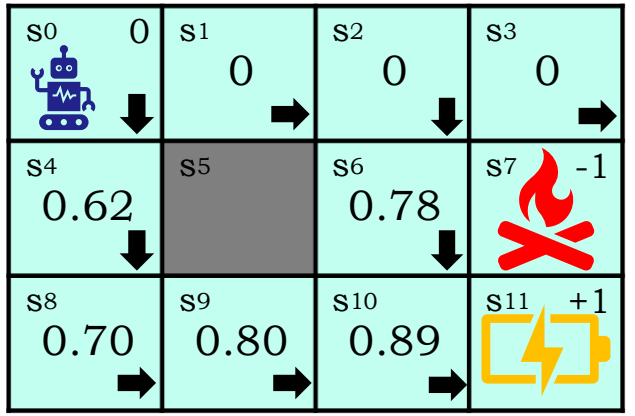
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$

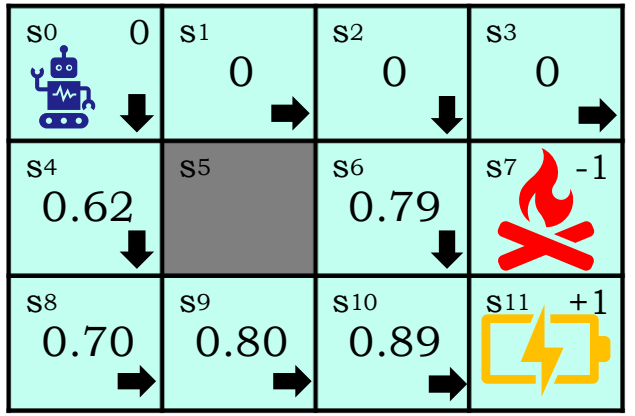
Policy Improvement

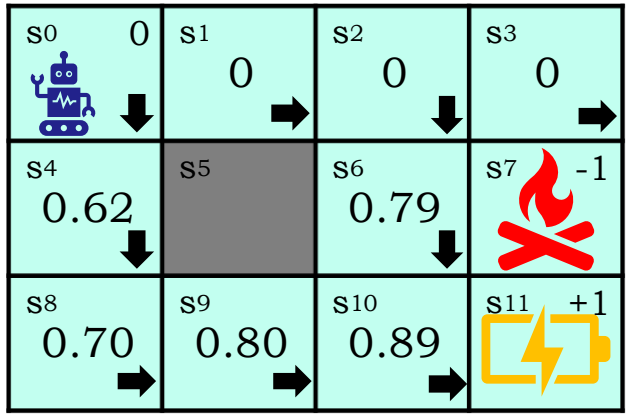
$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\big[r + \gamma V^{\pi_k}_i(s')\big]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

- Policy Improvement - II
  - $\pi_3(s) = \underset{a}{\operatorname{argmax}}\big(\sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_2}(s')]\big)$







https://www.profkumar.co.in/

Comparison of solution approaches of MDPs

$State, s \in S$
$Action, a \in A(s)$
$Reward, r \in R(s, a, s')$
$Transition\ Probability, P(s, a, s')$
$Optimal\ Policy, \pi^*(s)$
$Return, G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$
$State\ value, V^{\pi}(s)$
$Action\ value, Q^{\pi}(s, a)$
$V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$
$V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$
$Q^*(s, a) = \sum_{s'} p(s'|s, a)\left[r + \gamma \max_{a'} Q^*(s', a')\right]$
Policy Evaluation
$V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\left[r + \gamma V^{\pi_k}_i(s')\right]$
Policy Improvement
$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

Value Iteration

- Bellman's optimal value function
  - $V^*(s) = \max_a \sum_{s'} P(s'|s, a)\big(r + \gamma V^*(s')\big)$
- Bellman update
  - $V^*_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a)[r + \gamma V^*_k(s')]$

Policy Iteration

- Policy Evaluation
  - $V^{\pi_k}_{i+1}(s) = \sum_{s'} P(s'|s, \pi_k(s))\left[r + \gamma V^{\pi_k}_i(s')\right]$
- Policy Improvement
  - $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s, a)[r + \gamma V^{\pi_k}(s')]$

Most natural problems have unknown dynamics or a complicated model of environments.

- Value Iteration
  - Each iteration updates both return value and policy.
- Policy Iteration
  - Updates returns for a policy: Policy evaluation.
  - Updates the policy: Policy iteration.
  - Two loops. Faster outer loop (involving policy improvement)
- Benefits:
  - Simple algorithm.
  - Faster convergence.
- Limitations:
  - Requires the knowledge of complete dynamics of the environment a priori.

https://www.profrkumar.co.in/