Implementação

- Algoritmo de iteração de valor

Algorithm 4.1: Value iteration algorithm

Initialization: The probability models $p(r|s,a)$ and $p(s'|s,a)$ for all (s,a) are known. Initial guess v_0.
Goal: Search for the optimal state value and an optimal policy for solving the Bellman optimality equation.

While v_k has not converged in the sense that $\|v_k - v_{k-1}\|$ is greater than a predefined small threshold, for the kth iteration, do

  For every state $s \in \mathcal{S}$, do

    For every action $a \in \mathcal{A}(s)$, do

      q-value: $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_k(s')$

    Maximum action value: $a_k^*(s) = \arg\max_a q_k(s,a)$

    Policy update: $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise

    Value update: $v_{k+1}(s) = \max_a q_k(s,a)$

Implementação

Algoritmo de iteração de política

Algorithm 4.2: Policy iteration algorithm

Initialization: The system model, $p(r|s,a)$ and $p(s'|s,a)$ for all (s,a), is known. Initial guess π_0.
Goal: Search for the optimal state value and an optimal policy.

While v_{π_k} has not converged, for the kth iteration, do
    Policy evaluation:
    Initialization: an arbitrary initial guess $v_{\pi_k}^{(0)}$
    While $v_{\pi_k}^{(j)}$ has not converged, for the jth iteration, do
        For every state $s \in \mathcal{S}$, do
$$v_{\pi_k}^{(j+1)}(s) = \sum_a \pi_k(a|s) \left[\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_{\pi_k}^{(j)}(s')\right]$$
    Policy improvement:
    For every state $s \in \mathcal{S}$, do
        For every action $a \in \mathcal{A}$, do
$$q_{\pi_k}(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_{\pi_k}(s')$$
        $a_k^*(s) = \arg\max_a q_{\pi_k}(s,a)$
        $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise

Implementação

Algoritmo de iteração de política truncada

Algorithm 4.3: Truncated policy iteration algorithm

Initialization: The probability models $p(r|s,a)$ and $p(s'|s,a)$ for all (s,a) are known. Initial guess π_0.

Goal: Search for the optimal state value and an optimal policy.

While v_k has not converged, for the kth iteration, do

Policy evaluation:

Initialization: select the initial guess as $v_k^{(0)} = v_{k-1}$. The maximum number of iterations is set as j_{truncate}.

While $j < j_{\text{truncate}}$, do

For every state $s \in \mathcal{S}$, do

$v_k^{(j+1)}(s) = \sum_a \pi_k(a|s)\left[\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_k^{(j)}(s')\right]$

Set $v_k = v_k^{(j_{\text{truncate}})}$

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

Policy improvement:

For every state $s \in \mathcal{S}$, do

For every action $a \in \mathcal{A}(s)$, do

$q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_k(s')$

$a_k^*(s) = \arg\max_a q_k(s,a)$

$\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise

Implementação

- Algoritmo: MC Básico

Algorithm 5.1: MC Basic (a model-free variant of policy iteration)

Initialization: Initial guess π_0.
Goal: Search for an optimal policy.

For the kth iteration ($k = 0, 1, 2, \dots$), do
  For every state $s \in \mathcal{S}$, do
    For every action $a \in \mathcal{A}(s)$, do
      Collect sufficiently many episodes starting from (s, a) by following π_k
      Policy evaluation:
      $q_{\pi_k}(s, a) \approx q_k(s, a) =$ the average return of all the episodes starting from (s, a)
    Policy improvement:
    $a_k^*(s) = \arg\max_a q_k(s, a)$
    $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise