

Wprowadzenie do RL 6

Zadanie 1

Napisz program implementujący poniższy **algorytm iteracji wartości** dla środowiska *Frozen Lake* w celu znalezienia optymalnej polityki π^*

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\text{terminal}) = 0$

Loop:

```
|  $\Delta \leftarrow 0$   
| Loop for each  $s \in \mathcal{S}$ :  
|    $v \leftarrow V(s)$   
|    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$   
|    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$   
until  $\Delta < \theta$ 
```

Output a deterministic policy, $\pi \approx \pi_*$, such that
 $\pi(s) = \arg\max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$

Zadanie 2

Napisz program implementujący poniższy **algorytm obliczenia polityki** dla środowiska *Frozen Lake* w celu znalezienia wartości $V(s)$.

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
 $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in S_0, S_1, \dots, S_{t-1} :

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Przyjmij, że **polityka stochastyczna** π określa jednakowe prawdopodobieństwo dla każdej akcji: $\forall a \pi(a|s) = 1/4$. Przetestuj program dla różnych wartości parametru γ np. 1, 0.9, 0.5. Wykorzystaj notatnik: `FrozenLake_MC_generate_episode.ipynb`.