Wprowadzenie do uczenia ze wzmocnieniem

część 6

Q-learning: Off-policy TD Control

Zajmijmy się ponownie problemem kontroli czyli znalezienia funkcji Q dla optymalnej polityki.

Wiemu już jak można rozwiązać ten problem przy użyciu algorytmu SARSA.

W SARSA formuła na modyfikację Q jest następująca:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \Big]$$

Modyfikacja ta jest wykonywana po każdym przejściu ze stanu nieterminalnego S_t. Jeśli S_{t+1} jest stanem terminalnym, to $Q(S_{t+1},A_{t+1})$ jest zdefiniowane jako 0.

Q-learning: Off-policy TD Control

Algorytm Q-learning znalezienia funkcji Q:

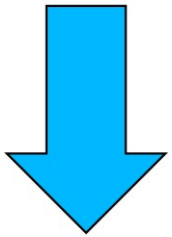
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$
    $S \leftarrow S'$
  until S is terminal

Q-learning: Off-policy TD Control

Przeanalizujmy algorytm.

Na początku wybieramy małe wartości współczynników oraz inicjujemy dowolnie wartości Q(s, a), przy czym dla stanu terminalego Q=0 (dla wszystkich akcji w tym stanie)



Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Q-learning: Off-policy TD Control

Przeanalizujmy pętlę główną algorytmu:

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    $S \leftarrow S'$
  until S is terminal

W każdym przejściu pętli agent przechodzi jeden epizod.

Q-learning: Off-policy TD Control

Zawartość pętli:

1 Initialize S
2 Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    $S \leftarrow S'$
  until S is terminal

Każdy epizod zaczyna się od stanu początkowego S. (1)

Petla wewnętrzna to przejścia agenta przez kolejne stany w epizodzie. (2)

Q-learning: Off-policy TD Control

Pętli wewnętrzna:

Loop for each step of episode:

3 Choose A from S using policy derived from Q (e.g., ε-greedy)

4 Take action A, observe R, S'

5 $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$

6 $S \leftarrow S'$

until S is terminal

W stanie S wybierana jest pewna akcja A (przy zastosowaniu polityki epsilon-zachłannej → patrz opis algorytmu SARSA). (3)

Agent wykonuje akcję A. W efekcie otrzymuje nagrodę R i przechodzi do stanu S'. (4)

Następnie modyfikowana jest wartość Q(S,A). (5)

Na końcu następuje podstawienie S=S', A=A'. (6)

Q-learning: Off-policy TD Control

Porównajmy formuły na modyfikacje wag.

SARSA

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \Big]$$

Q-learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big].$$

Jak widać różnica polega na tym, że w SARSA w stanie S_{t+1} musi być wybrana jakaś akcja (i do tego potrzebna jest polityka). Po wybraniu pewnej akcji A_{t+1} możliwe jest znalezienie wartości $Q(S_{t+1},A_{t+1})$.

Q-learning: Off-policy TD Control

Porównajmy formuły na modyfikacje wag.

SARSA

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \Big]$$

Q-learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big].$$

W algorymie Q-learning nie ma potrzeby wybierania akcji w stanie S_{t+1} (nie potrzebujemy zatem polityki – stąd 'off-policy' w nazwie algorytmu). Zamiast tego znajdujemy największą z wartości $Q(S_{t+1},a)$ po wszystkich akcjach możliwych w stanie S_{t+1}.

Koniec części 6