

Reinforcement Learning

1. Jelaskan cara kerja dari algoritma **Q-Learning** dan **SARSA**, terutama perbedaan fundamental antara keduanya (*on-policy* vs *off-policy*).

Q-Learning bersifat off policy. Q-Learning akan belajar tentang jalur terbaik tanpa mepedulikan aksi yang ia lakukan saat ini karena ia akan berasumsi untuk aksi selanjutnya ia akan memilih aksi yang terbaik (walaupun ada kemungkinan random dari eksplorasi).

Update table Q-Learning akan menggunakan rumus ini:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

Diagram labels for the Q-Learning update formula:

- Old Q Value: $Q(s, a)$ (before update)
- New Q Value: $Q(s, a)$ (after update)
- Learning Rate: α (0 ~ 1)
- Reward: r
- Discount Rate: γ (0 ~ 1)
- Maximum Q value of transition destination state: $\max_{a'} Q(s', a')$
- TD error: $r + \gamma \max_{a'} Q(s', a') - Q(s, a)$

Sedangkan, SARSA bersifat on policy. SARSA akan belajar berdasarkan aksi yang benar-benar akan ia ambil di langkah berikutnya, sesuai policy saat ini.

Update table SARSA akan menggunakan rumus ini:

$$Q(s, a) = Q(s, a) + \alpha [r + \gamma Q(s', a') - Q(s, a)]$$

Diagram labels for the SARSA update formula:

- Updated Q-value: $Q(s, a)$ (after update)
- Current Q-value: $Q(s, a)$ (before update)
- Target Q-value: $r + \gamma Q(s', a')$
- Current Q-value: $Q(s, a)$ (before update)

2. Bandingkan hasil dari kedua algoritma tersebut dalam konteks **Wumpus World** ini.

Untuk kasus tugas ini, tidak ada perbedaan secara signifikan. Keduanya memiliki waktu eksekusi yang mirip dan jalur eksekusi yang sama persis. Ini karena tempat permainan cukup sempit dan tidak ada ruang untuk SARSA mengambil jalur yang lebih “aman”. Namun, episodes yang diperlukan untuk mencapai tujuan seharusnya lebih cepat Q-Learning karena SARSA akan lebih sulit belajar ke state yang lebih “risky” yaitu posisi gold yang terletak diantara dua tile pembunuh.