Reinforcement Learning

Salman Hanif - 13523056

Cara Kerja Algoritma Q-Learning dan SARSA

Algoritma Q-Learning dan SARSA memiliki fundamental policy yang masih sama, yaitu dengan mengacu kepada Q-Table dalam pengambilan keputusan,. Keduanya bekerja dengan mengolah dan memperbarui Q-table nya sendiri, dengan menyimpan nilai untuk setiap state-action pair. Policy ini didasari oleh reward numerik yang didapat agent pada setiap action di state tertentu. Proses pembaharuan dari Q-Table oleh agent itu sendiri bertujuan untuk menghasilkan policy yang optimal di lingkungan yang tidak diketahui.

Tabel numerik Q-Table , jumlah baris dan kolomnya tergantung kepada jumlah state/kondisi (baris) dan available action/step (kolom) yang dapat dilakukan pada state tersebut. Pemodelan matematis yang mengisi setiap kolom tabel pada Q-Table didasari oleh Markov Decision Processes (MDPs), di mana setiap nilai mewakili ekspektasi *return* dari aksi yang dilakukan pada *state* terkait.

Perbedaan fundamental Q-Learning & Sarsa (off-policy vs on-policy)

Perbedaan fundamental dari on-policy (Sarsa) dan off-policy (Q-Learning) adalah pada metode pembelajaran dan pembaharuan Q-Table-nya, yaitu apakah mereka memperhitungkan kebijakan yang sedang dimiliki saat ini atau tidak.

Sarsa menggunakan policy yang dimiliki saat ini untuk memperbarui nilainya (belajar&mempertimbangkan tindakan yang benar-benar akan diambil), sedangkan Q-Learning tidak, dia mempelajari tindakan optimal/terbaik yang bisa diambil di masa depan, terlepas dari aksi yang benar-benar diambil.

Secara bahasa mungkin membingungkan saat dibaca, tapi dengan rumus matematis Q-Table berikut dapat lebih mudah dipahami.

Sarsa

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

Q-Learning

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(R_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

Sarsa memperbarui nilai Q-Table dengan mempertimbangkan action yang diambil di step selanjutnya (at+1) (next action diambil didasari Q-Table saat ini), sedangkan Q-Learning mengambil nilai max / optimal yang tersedia di next step. Jadi bisa dikatakan Q-Learning lebih optimis dan cepat, sedangkan Sarsa lebih konservatif.

Perbandingan Hasil Kedua Algoritma di Wumpus World

Kecepatan konvergensi (jumlah episode yang dibutuhkan untuk belajar)	Algoritma Q-Learning yang mengedepankan langkah optimal pada pembaruan tabelnya lebih cepat (konvergensi) untuk mencapai policy optimalnya dibandingkan dengan Sarsa. Dengan alpha/learning rate senilai 0.15 dan epsilon decay 0.999, Q-learning lebih cepat 2.5 kali dari Sarsa. Pada nilai variabel lain, contohnya epsilondecay 0.995, Q-learning bahkan 3 kali lebih cepat. Lebih cepat di sini artinya, jumlah episode yang diperlukan untuk mencapai langkah paling optimal masing-masing agen.
Kebijakan (policy) final yang dihasilkan	Pada peninjauan dari nilai Q-Table, nilai dari setiap action di Q-learning memiliki nilai yang tinggi-tinggi dan cenderung tidak berbeda jauh antar action. Berbeda dengan Sarsa yang nilai action optimalnya jauh melebih action lainnya. Hal ini karena berdasarkan sifat matematisnya, Sarsa lebih mempertimbangkan action at+1 yang sesuai policy sehingga lebih realistis dan berhati-hati, nilai Q-Table lebih optimis, tetapi tetap lebih efisien secara hasil dan kecepatan konvergensi.
Jalur (path) yang ditempuh berdasarkan risiko	Ini adalah poin paling terlihat dan menarik yang membedakan Q-learning dan Sarsa. Dikarenakan selama pembaruan Q-Table sarsa memperhitungkan langkah selanjutnya, maka langkah yang berisiko cenderung dia hindari dan lebih berhati-hati. Walaupun tidak optimal, tetapi path yang ditempuhnya memang meminimalisasi risiko kekalahan. Paling terlihat adalah setelah grab gold, karena di kanan dan kiri kolom gold ada Wumpus dan PIT, Sarsa memilih maju satu langkah baru kemudian putar balik ketimbang langsung turn left/right untuk menghindari risiko masuk ke sana dan kalah.