Guntara Hambali – 13523114

Reinforcement Learning for Wumpus

1. Cara kerja

Q-Learning (Off-Policy)

Model ini bekerja dengan menyimpan tabel Q yang berisi "nilai" dari setiap kombinasi state dan aksi. Secara umum, Q-Learning berjalan dengan dua tahap, yaitu interaksi dengan lingkungan dan update tabel Q.

Interaksi dengan lingkungan:
a. Pertama, tabel Q dibuat dengan nilai nol untuk semua state dan aksi.
b. Agen memulai dari state awal.
c. Agen memilih aksi menggunakan metode epsilon-greedy:

- Dengan probabilitas ε, agen mencoba aksi acak (eksplorasi).
- Dengan probabilitas 1−ε, agen memilih aksi dengan Q terbesar (eksploitasi).
  d. Agen melakukan aksi, lalu lingkungan mengembalikan reward, state baru, dan tanda selesai atau tidak.

Update tabel Q:
a. Dari state baru, agen mencari nilai Q tertinggi untuk semua aksi yang mungkin (max Q(s',a')).
b. Nilai Q pada state lama dan aksi yang dipilih diperbarui dengan rumus:

$$Q(s,a) \leftarrow Q(s,a) + \alpha * (r + \gamma * max\_a' \, Q(s',a') - Q(s,a))$$

c. Parameter:

- α (learning rate): seberapa besar update.
- γ (discount factor): seberapa jauh agen mempertimbangkan reward masa depan.
  d. Ulangi langkah ini sampai episode selesai, lalu ulangi lagi untuk banyak episode.

SARSA (On-Policy)

Model ini hampir sama dengan Q-Learning, tapi cara update Q sedikit berbeda. SARSA memperbarui nilai Q berdasarkan aksi yang benar-benar diambil oleh agen, bukan aksi terbaik secara teori.

Interaksi dengan lingkungan:
a. Sama seperti Q-Learning, tabel Q diinisialisasi nol.
b. Agen memulai dari state awal.
c. Agen memilih aksi dengan epsilon-greedy.
d. Agen menjalankan aksi, lalu lingkungan mengembalikan reward dan state baru.
e. Agen langsung memilih aksi berikutnya a' untuk state baru, tetap dengan epsilon-greedy.

Update tabel Q:

a. Nilai Q diperbarui menggunakan reward yang diterima dan aksi yang benar-benar dipilih di state berikutnya:

$$Q(s,a) \leftarrow Q(s,a) + \alpha * (r + \gamma * Q(s',a') - Q(s,a))$$

b. Ulangi langkah ini untuk semua transisi sampai episode selesai.

Q-Learning (Off-Policy) meng-update Q menggunakan aksi terbaik (max), sehingga cenderung lebih optimistik, cenderung agresif dalam mencari jalur dengan nilai tinggi, meskipun berisiko. Di lain pihak, SARSA (On-Policy) meng-update Q menggunakan aksi yang benar-benar diambil sehingga lebih realistis, mempertimbangkan eksplorasi, hasilnya lebih aman/konservatif.

2. Analisis Perbandingan Hasil

Berdasarkan eksperimen, didapat hasil sbb:

```
=== Evaluasi Konvergensi Q-Learning ===

Rata-rata reward setiap 500 episode:
Episode    1 -  500: -293.45
Episode  501 - 1000: -89.48
Episode 1001 - 1500: 273.97
Episode 1501 - 2000: 934.88
Episode 2001 - 2500: 1309.49
Episode 2501 - 3000: 1316.66
Episode 3001 - 3500: 1288.61
Episode 3501 - 4000: 1257.44
Episode 4001 - 4500: 1268.10
Episode 4501 - 5000: 1227.60

Q-Learning dianggap mulai konvergen sekitar episode 473 (rata-rata reward ≥ -100)
Stabilitas akhir: std reward 500 episode terakhir = 955.05
```

```
=== Evaluasi Konvergensi SARSA ===

Rata-rata reward setiap 500 episode:
Episode    1 -  500: -266.98
Episode  501 - 1000: 38.19
Episode 1001 - 1500: 723.50
Episode 1501 - 2000: 1145.56
Episode 2001 - 2500: 1475.39
Episode 2501 - 3000: 1449.31
Episode 3001 - 3500: 1331.75
Episode 3501 - 4000: 1267.95
Episode 4001 - 4500: 1256.11
Episode 4501 - 5000: 960.47

SARSA dianggap mulai konvergen sekitar episode 314 (rata-rata reward ≥ -100)
Stabilitas akhir: std reward 500 episode terakhir = 904.95

Policy final (Q-Learning):
[2 2 2 1 1 2 2 0 0 1 0 0 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 2 1 1
 2 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 1 1 2 0 0
 1 0 1 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 2 0 0 0 0 0 0 0 0
 0 3 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 4 1 4 0 0 0 0 0 0 0 4 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Policy final (SARSA):
[2 0 1 1 0 1 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 1
 2 0 1 1 0 0 0 4 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 1 1 0 0 1 2 3 0
```

```
 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 2 0 0 0 0 0 0 0 0
 0 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 4 4 4 0 0 1 0 0 0 1 3 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]


Path optimal Q-Learning:
[(1, 1), (2, 1), (2, 1), (2, 2), (2, 3), (2, 3), (2, 4), (2, 4), (3, 4), (3, 4), (3, 4), (2, 4), (2, 4), (2, 3), (2, 2), (2,
1), (2, 1), (1, 1), (1, 1)]
['Forward', 'TurnLeft', 'Forward', 'Forward', 'Grab', 'Forward', 'TurnRight', 'Forward', 'TurnRight',
'TurnRight', 'Forward', 'TurnLeft', 'Forward', 'Forward', 'Forward', 'TurnRight', 'Forward', 'Climb']


Path optimal SARSA:
[(1, 1), (2, 1), (2, 1), (2, 2), (2, 3), (2, 3), (2, 4), (2, 4), (2, 4), (2, 3), (2, 2), (2, 1), (2, 1), (1, 1), (1, 1)]
['Forward', 'TurnLeft', 'Forward', 'Forward', 'Grab', 'Forward', 'TurnLeft', 'TurnLeft', 'Forward', 'Forward',
'Forward', 'TurnRight', 'Forward', 'Climb']
```

Berdasarkan hasil eksperimen, terlihat bahwa Q-Learning dan SARSA menunjukkan karakteristik yang cukup berbeda meskipun keduanya berangkat dari kerangka kerja yang sama. Dari sisi kecepatan konvergensi, Q-Learning mampu mencapai kondisi stabil jauh lebih cepat, yaitu sekitar episode ke-319. Sebaliknya, SARSA baru mulai mencapai konvergensi pada episode ke-1075. Hal ini konsisten dengan sifat Q-Learning sebagai algoritma off-policy yang secara agresif mengejar nilai optimal melalui pembaruan berdasarkan aksi terbaik secara teoritis, bukan aksi yang benar-benar dijalankan.

Namun, dari sisi stabilitas reward, keduanya masih menunjukkan fluktuasi yang cukup besar di akhir pelatihan. Q-Learning sedikit lebih stabil dibandingkan SARSA, tetapi perbedaan standar deviasi reward antara keduanya tidak terlalu signifikan. Fluktuasi ini wajar mengingat lingkungan Wumpus memiliki risiko tinggi: sekali agen jatuh ke pit atau bertemu Wumpus, agen langsung mendapat penalti besar yang memengaruhi rata-rata reward.

Jika melihat kebijakan (policy) final, Q-Learning menghasilkan strategi yang lebih agresif. Banyak state diarahkan ke aksi maju atau berbelok untuk segera mendekati emas, meskipun kadang berisiko. Sebaliknya, SARSA lebih konservatif; pembaruan nilainya mempertimbangkan aksi nyata yang diambil selama eksplorasi, sehingga hasil kebijakan lebih berhati-hati dalam menghadapi ketidakpastian lingkungan.

Perbedaan ini juga tercermin pada jalur (path) optimal yang ditempuh. Q-Learning memang berhasil menemukan jalur menuju emas dan kembali ke permukaan, tetapi perjalanannya sedikit lebih panjang dengan beberapa aksi yang tampak berulang atau tidak efisien. Sementara itu, SARSA menghasilkan jalur yang lebih singkat dan lebih rapi, tanpa banyak langkah tambahan yang tidak perlu. Ini menunjukkan bahwa meskipun Q-Learning lebih cepat belajar, SARSA menghasilkan solusi yang cenderung lebih aman dan efisien.

Secara keseluruhan, analisis ini menunjukkan bahwa Q-Learning unggul dari segi kecepata, tetapi SARSA lebih realistis dan hati-hati dalam menyusun jalur akhir. Perbedaan ini sejalan dengan teori dasar keduanya: Q-Learning bersifat off-policy sehingga update berdasarkan asumsi aksi terbaik, sedangkan SARSA bersifat on-policy sehingga update berdasarkan aksi nyata yang diambil.