Reinforcement Learning 強化学習

Q-Learning vs SARSA

Agent States エージェント状態

NOTE: Grid program merupakan vertical flip dari grid di spek sehingga “up” di program sama dengan “down” di grid yang ada di spek.

1. row
2. col
3. is_breeze
4. is_stench
5. is_glitter
6. has_reward

Q-Learning　Q 学習

Q-Learning adalah sebuah metode dalam Reinforcement Learning (RL) yang bekerja dengan cara mengambil action paling optimal pada saat ini, dengan asumsi action yang diambil di langkah selanjutnya paling optimal juga.

Procedure　方法

1. **Buat sebuah Q-Table kosong.**
   Q-Table adalah sebuah tabel yang menyimpan pasangan state dan action sebagai key, dan perkiraan reward yang didapatkan sebagai value. Pada kasus ini, Q-Table disini berbentuk sebuah Python Dictionary, dan jika get() tidak menemukan key-nya, maka akan bernilai 0.
2. **Pilih action yang diambil dengan *epsilon-greedy dan lakukan*.**
   Action yang diambil pada sebuah state dipilih dengan *epsilon-greedy* yaitu, dari sebuah nilai epsilon diantara 0 dan 1 yang diinisialisasi diawal, jika hasil random 0 sampai 1 jatuh diatas nilai epsilon, maka action yang dipilih adalah action yang memiliki Q-value tertinggi di state saat ini. Jika hasil random jatuh dibawah nilai epsilon, maka akan dipilih sebuah action random.
3. **Update Q-value saat ini.**
   Agent melakukan aksi dan pindah ke state berikutnya dengan t+1, Q-value di t di-update dengan rumus dibawah ini:
   $$Q(s_t, a_t) = Q(s_t, a_t) + \alpha * (r_t + \gamma * max_a\, Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$
   Dimana:

- Q(s,a) = Q-value pada s dan a
- s = state
- a = action
- t = saat ini
- α = alpha / learning rate
- r = reward
- γ = gamma / discount factor terhadap Q-value masa depan
- max = action dengan Q-value terbesar pada state selanjutnya

Dapat dilihat bahwa dengan Q-learning, estimasi nilai reward bergantung pada asumsi agent selalu memilih action yang paling optimal di masa depan, walaupun mungkin hal tersebut tidak selalu terjadi akibat adanya random dari epsilon.

4. **Ulangi langkah 2 dan 3 hingga episode berakhir, lalu ulangi hingga batas jumlah episode.**
   Sebuah episode berakhir jika agent masuk kedalam state terminal, yaitu menabrak wumpus, masuk jurang, atau memanjat keluar. Setelah episode diulang hingga batas jumlah episode, Q-table yang dihasilkan dapat digunakan untuk menentukan action paling optimal pada setiap state.

SARSA

SARSA yaitu singkatan dari State Action Reward State Action, juga adalah metode Reinforcement Learning, yang berbeda sedikit dibandingkan Q-Learning. Perbedaanya adalah ia tidak mengasumsi agent akan selalu mengambil aksi yang paling optimal, melainkan dia akan memilih dulu aksi apa yang akan dilakukannya di masa depan, sehingga estimasi Q-value berdasarkan aksi yang sebenarnya dilakukan.

Procedure 方法

1. **Buat sebuah Q-Table kosong.**
2. **Pilih action yang diambil dengan *epsilon-greedy*.**
3. **Lakukan action yang telah dipilih**
4. **Pilih lagi action yang akan diambil di state selanjutnya dengan epsilon-greedy**
   Berbeda dengan Q-Learning yang menggunakan asumsi action selanjutnya adalah paling optimal, SARSA akan memilih action selanjutnya sebagai komitmen yang pasti dilakukan.
5. **Update Q-value saat ini.**
   Q-value di t di-update dengan rumus dibawah ini:

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

Perhatikan bahwa tidak seperti Q-Learning, tidak ada max a disini karena action yang akan diambil sudah dipilih sebelumnya, tidak peduli optimal atau tidak (jika random akibat epsilon).

6. **Ulangi langkah 3 hingga 5 hingga episode berakhir, lalu ulangi hingga batas jumlah episode.**
   Sebuah episode berakhir jika agent masuk kedalam state terminal, yaitu menabrak wumpus, masuk jurang, atau memanjat keluar. Setelah episode diulang hingga batas jumlah episode, Q-table yang dihasilkan dapat digunakan untuk menentukan action paling optimal pada setiap state.

Comparison 比較

Off-Policy vs On-Policy

Perbedaan paling dasar adalah Q-Learning bersifat Off-Policy dan SARSA bersifat On-Policy. Policy disini berarti aksi yang sesungguhnya diambil. Q-Learning bersifat Off-Policy karena rumus dalam update Q-value menggunakan asumsi aksi paling optimal akan dipilih, tidak peduli akhirnya diambil atau tidak. Sebaliknya, SARSA bersifat On-Policy karena rumus dalam update Q-value menggunakan aksi yang sebenarnya diambil.

Results Table 結果テーブル

	Method	*Total Score*	*First Win Episode*
1	Q-Learning	904381.0	1
	SARSA	905603.0	19
2	Q-Learning	919500.0	24
	SARSA	897155.0	14
3	Q-Learning	910414.0	22
	SARSA	874231.0	18
4	Q-Learning	905506.0	29
	SARSA	860714.0	26
5	Q-Learning	919492.0	18
	SARSA	900868.0	20

Q-Learning rata-rata mendapatkan perolehan total score lebih tinggi dibanding SARSA. Rata-rata first win episode Q-Learning adalah 18.8 disini, dan SARSA adalah 19.4 sehingga Q-Learning disini rata-rata menang sebelum SARSA. Path yang ditemukan kedua algoritma sama dari pilihan kedua path menang yang paling optimal yaitu :

(['down', 'right', 'down', 'grab', 'up', 'left', 'up', 'climb'], 994.0) atau

(['right', 'down', 'down', 'grab', 'up', 'left', 'up', 'climb'], 994.0)

Secara teori, Q-Learning lebih cenderung mengambil resiko dibandingkan SARSA karena selalu mengasumsikan langkah optimal akan diambil di masa depan. Namun pada kasus ini tidak ada perbedaan karena tidak ada rute yang lebih beresiko dibanding rute lainnya.