Report

Requirements

Programming language: python3
You should build the Cliff Walking environment and search the optimal travel path by Sara and Q-learning, respectively.
Different settings for ε can bring different exploration on policy update. Try several ε (e.g. ε = 0.1 and ε = 0) to investigate their impacts on performances.

My Implementation

Sarsa

```
take action first, then observe the action by epsilon-greedy method at the next state.
update the Q(state,action) by calculation and then update a with a' and s with s'
```

Q-Learning

```
similar to Sarsa, just different when taking the next action.
It only takes the prediction without random.
```

Both use the same functions written in the program.

Use observe to observe the next state and the reward by the given state and action.

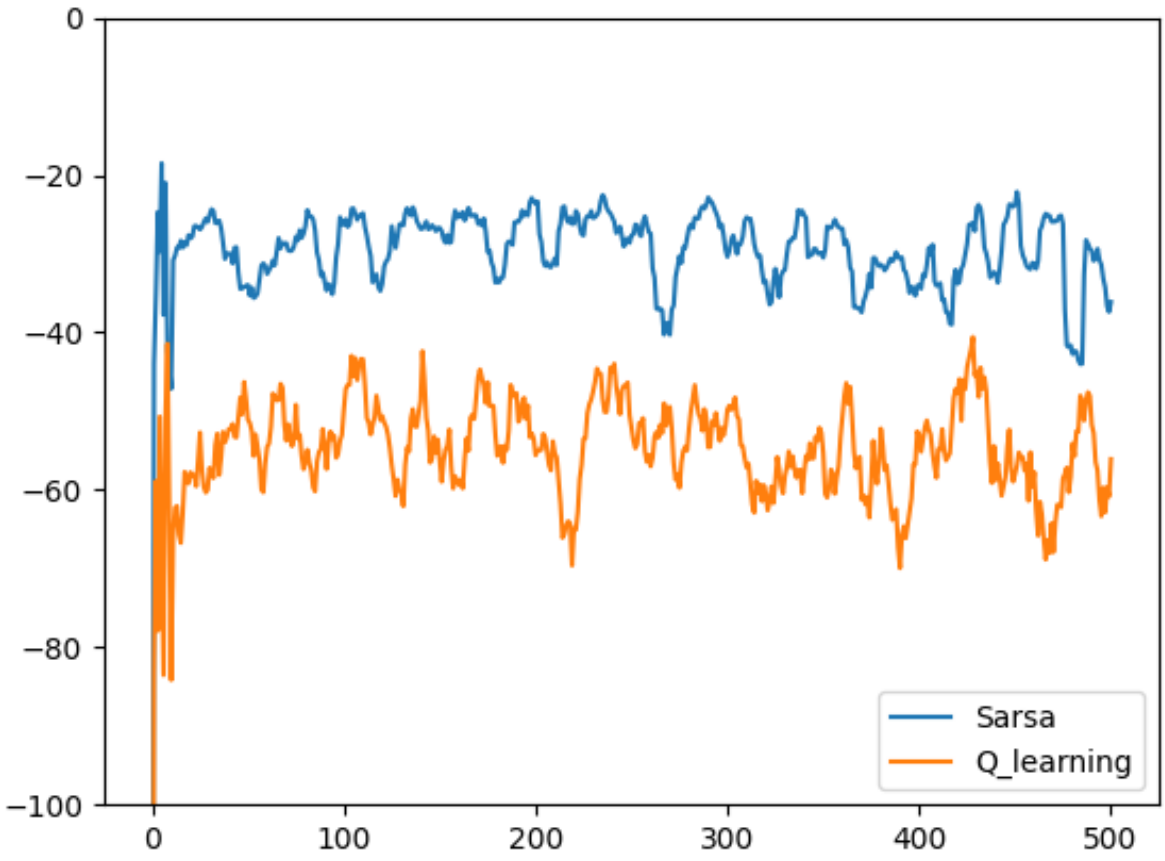
Use eg-policy to choose an epsilon-greedy action, and use predict to choose a greedy action.

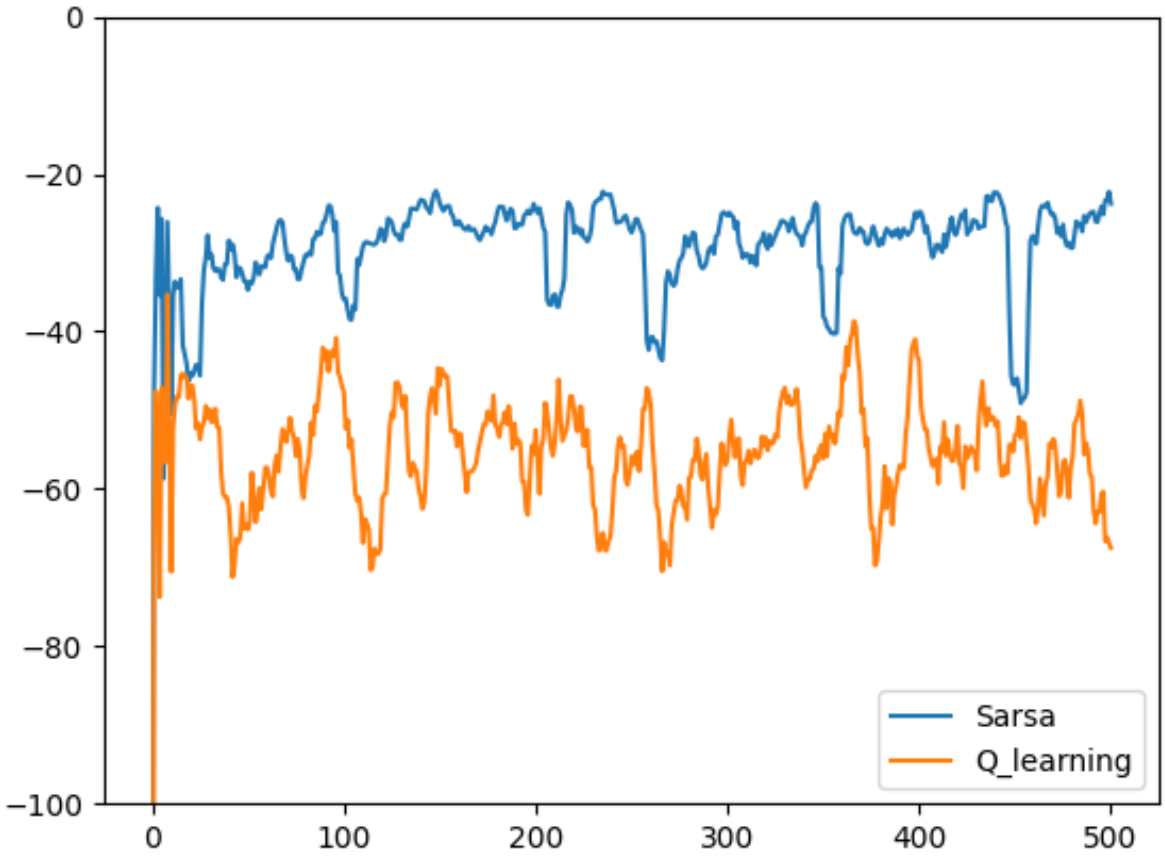
Result

ε = 0.1

Gets the result as expected. The Sarsa and Q-learning chooses different path because Q-learing use absolute greedy policy when choosing the next action, which permits that when Q reaches convergence it won't take actions that will fall down the cliff, while Sarsa use epsilon-greedy method which is effected by the cliff even when Q reaches convergence. Therefore, Sarsa takes the actions far away from the cliff.

```
Sarsa path:
→  →  →  →  →  →  →  →  →  →  →  ↓
↑  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  E
Q-Learning path:
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  E
```

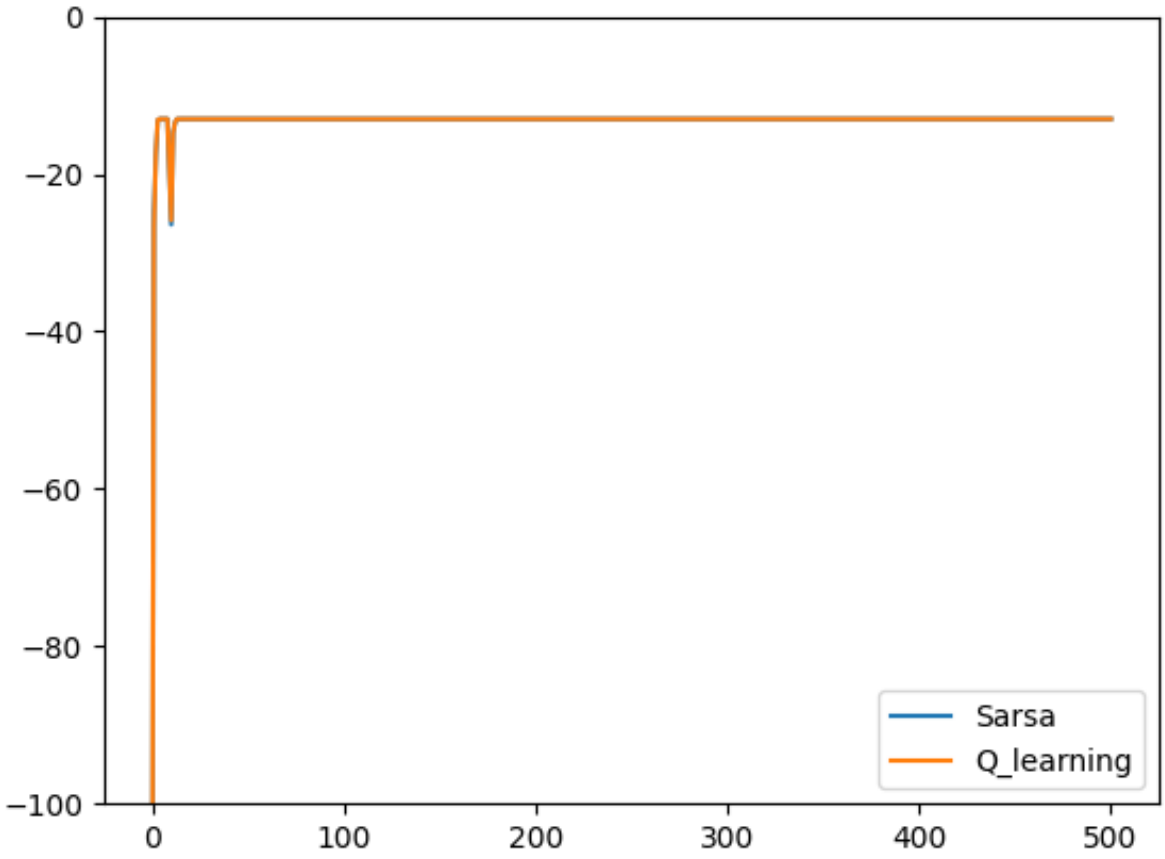




Sarsa reaches convergence slower than Q-Learning, while it results better in average rewards as it chooses a relatively safer route.

ε = 0

Obviously, when ε = 0 Sarsa and Q-Learning are the same. They all become TD(0)



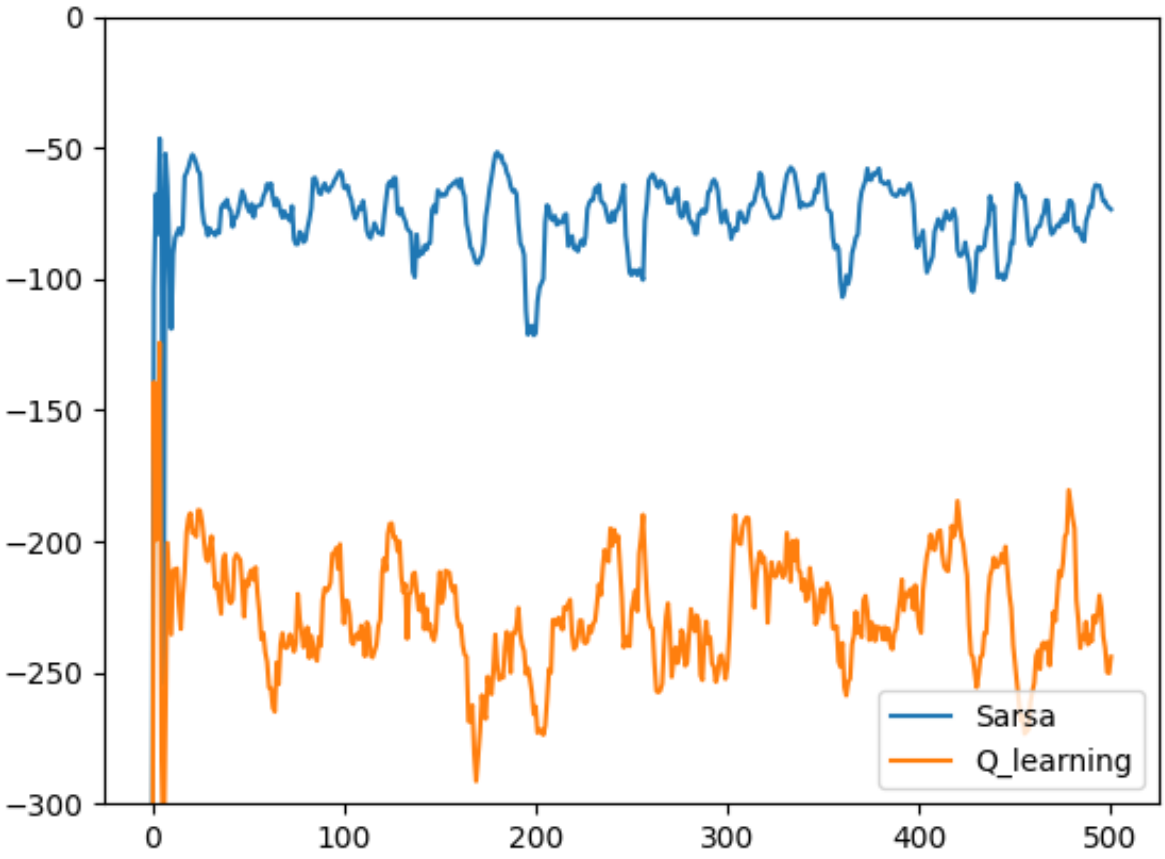
ε = 0.3

Sarsa observes several path with different attempts. Q-Learning is the same as before. The rewards becomes smaller due to the larger ε and it becomes more instable especially when using Sarsa.

```
Sarsa path:
→  →  →  →  →  →  →  →  →  →  →  ↓
↑  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  E
Q-Learning path:
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  E
```

```
Sarsa path:
→  →  →  →  →  →  ↓  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  E
Q-Learning path:
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  E
```

```
Sarsa path:
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  E
Q-Learning path:
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  E
```


0
−50
−100
−150
−200
−250
−300
0
100
200
300
400
500
Sarsa
Q_learning