Comparing n-step Q(sigma) with other n-step methods

Zhuoyu Wang (260454547)

In this exercise, I implement the SARSA(0), n-step SARSA, n-step Expected SARSA, n-step tree backup and n-step Q(sigma) on a grid game.

**The Environment**

The example is simplified from Example 11.8 on <http://artint.info/html/ArtInt_262.html>. The original setting of the environment is available online. I tried the complex version of the game but it took too much time for the agent to learn it, so I modified the setting to simplify it. The following is the simplified version of it.

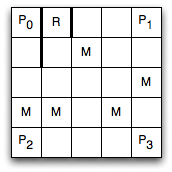


Figure 1. The grid game used in this exercise

As shown in Figure 1, there are 25 grid locations the agent could be in, and in each episode the agent starts from P3. A prize is on the up-left corner, P0. When the agent lands on a prize, it receives a reward of *10* and the game stop. There are five monsters appear at the locations marked *M*. If the agent lands on these locations, it receives a reward of *-10*.

In this example, the state consists of two components: *⟨X,Y⟩*, where *X* is the *X*-coordinate of the agent, *Y* is the *Y*-coordinate of the agent. There are thus *5×5 = 25* states.

The agent has four actions: *up*, *down*, *left*, and *right*. These move the agent one step in the direction indicated by the name. If the agent crashes into an outside wall or one of the interior walls (the thick lines near the location *R*), it remains where was and receives a reward of *-1*.

**The experiments**

The settings of the experiments are as follow:

Discount rate ϒ= 0.90;

Greedy policy ε=0.10;

Step size α=0.25 and 5;

n in the n-step methods n=2, 4, 6, 8;

sigma=0.5 in Q(sigma);

Considered methods: SARSA(0), n-step SARSA, n-step Expected SARSA, n-step tree backup and n-step Q(sigma).

For each setting and each method, training the agent for 50 times and get the average of the time steps and the average returns of each time step before the agent converges. The stop criterion is that for 20 sequential episodes, the optimal action matrix doesn’t change.

In some case, tree backup may stick in some states and cannot jump out from them even if the step time is over 100000 in the episode. For example, it may repeatedly switch between two locations. To avoid this, the agent stops when it achieves a maximum of 5000 steps. Only results of the first 25 time steps are plotted since there is not too much change after 25 time steps.

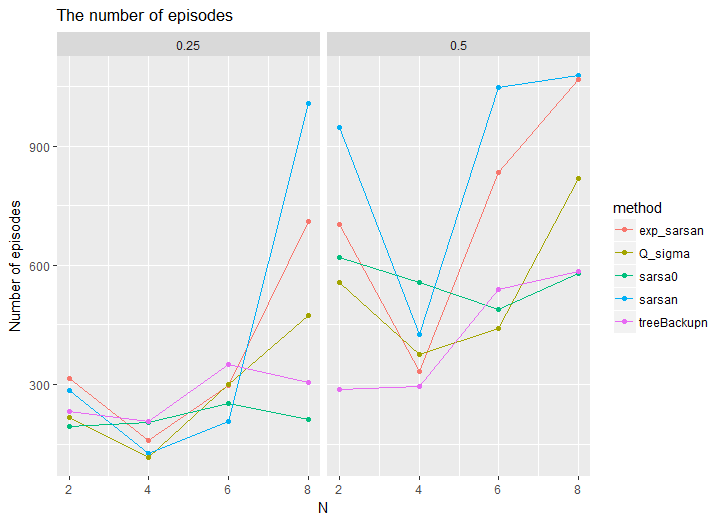


Figure 2. The number of episodes before convergence when n=2, 4, 6, or 8 and alpha=0.25 or 0.5.

Figure 2 presents the number of episodes before convergence when n=2, 4, 6, or 8 and alpha=0.25 or 0.5. It seems the SARSA(0) is the most stable method and converges within a reasonable time steps. For other methods, they works better when n=4 (how about n=3 and n=5?) than other values. Q(Sigma) works best among all n-step methods when n=4 and alpha=0.25, and treeback works best when n=8 for both alpha = 0.25 and a = 0.5 settings. For both alpha = 0.25 and a = 0.5 settings, the number of episodes increase dramatically when n increases from 4 to 6 and 8.

Figure 3 shows the mean return of each time step averaging over 50 episodes. Here we plot the returns within 25 time steps. After 25 time steps, the average returns are almost flat. For all settings, SARSA(0) has the highest returns. Among all the n-step methods, Q(Sigma) has the highest returns.

