CMPUT 366 Reading-Writing Exercise 9

Name: Minghan Li
Student ID: 1561234
CCID: minghan4

After we obtain the knowledge of how to combine function approximation and reinforcement learning algorithms for policy evaluation in Chapter 9, the next thing we want to do is apply the knowledge on control problems. Most ideas would remain the same: we can still use semi-gradient descent to update the weights of our function approximator, and we still can use TD target or actual return as our target. However, we need to pay attention to the continuing problems with function approximation, for the classic discounting setting kind of break in this situation.

One of the most classic on-policy control algorithms in tabular case is *Sarsa*. With function approximation, we can use the same *semi-gradient* method to update the parameters just like in Chapter 9 except in this time we try to estimate $q(s, a)$, and actions can be treated as either input of the function or we can specify several action heads if the actions are discrete. In the Mountain Car example we can see the linear approximation with tile-coding forms a linear expansion over the state-space and sarsa can still find the optimal policy with appropriate step-size. Similarly, we can also incorporate n-step sarsa with function approximation as well. The sensitivity analysis over different n with respect to the step-size is similar to the one in tabular case.

Perhaps the most important concept in this chapter is the *average reward*. Although we have seen this kind of setting quite often in dynamic programming but it rarely shows up in reinforcement learning problem. One reason is that the discount setting has already allowed us to consider long-term rewards in certain horizon, i.e. $\frac{1}{1-\gamma}$. However, in continuing problem and with function approximation, this setting is deprecated and we have to use *reward rate* instead of discounted return. In average reward setting, we would expect the markov chain has to be *ergodic*, which means that the agent can visit every state aperiodically. It also means that where the MDP starts or any early decision made by the agent can have only a temporary effect.

To show why the discounting setting breaks in continuing problems with function approximation, the authors points out that the discounted return is actually proportional to the average reward, which means that the ranking of policies in discounting setting would be exactly the same as the average reward setting even we set $\gamma = 0$. Although we can still use discounted return and maybe achieve the similar performance as using average reward, we have to know that it is actually a *solution*, but using discounting setting here for continuing *problem* just makes less sense than average reward setting, for which in the approximate case most policies cannot be represented by a value function. The arbitrary policies that remain need to be ranked, and the scalar average reward $\mathrm{r}(\pi)$ provides an effective way to do this.

In the end, the author also discuss how to incorporate n-step TD in average reward setting with function approximation. The only modification we have to make is instead discounting the q-value of the next state-action pair, we use the difference between the immediate reward and the difference between the $q(s, a)$ and $q(s', a')$ as our temporal difference. To sum up, with function approximation we can do control in much more complex situations but we also have to be careful about choosing the probelm setting, e.g. discounting setting or average reward setting, for each specific task.