Assessment 6

SIT 796

Introduction

This article introduces temporal differences building on top of Monte Carlo methods and goes on to explain the relationship they have with eligibility traces.

Temporal Differences and Eligibility Traces

Monte Carlo method learns optimal behavior from experience i.e., from agent's interaction with the environment. This type of learning is called online learning, where the agent learns without the knowledge of probability transitions.

In a Monte Carlo policy evaluation, the state value function is updated as:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

- $V(S_t)$ is the state value,
- α -> learning rate or step size
- G_t -> total discounted reward for S_t

Where G_t is calculated as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T$$

The final reward or the total discounted reward G_t is used to update values for each state. This denotes the strategy update after every episode.

Temporal Differences build upon the concept of Dynamic Programming and Monte Carlo methods and can be considered a blend of the two methods, they do not need the episode to end which is the major difference from Monte Carlo method. The TD method waits only until the next step to determine the new value for $V(S_t)$. This update is denoted by:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

- $V(S_t)$ is the state value,
- α -> learning rate or step size
- R_{t+1} is the reward observed for time t+1
- $YV(S_{t+1})$ is the discounted value function at t+1

This allows the use of DP, as the state value function can be represented in terms of successive states. The replacement of $G_t \rightarrow R_{t+1} + \gamma V(S_{t+1})$ shows how TD methods update every step. Using these temporal methods an agent applies what it learns in the next step to the policy in current step.

TD and MC both use a forward view i.e., the update steps depend on rewards that are from the following time step (future time step). This makes the update step complex. This is where **Eligibility traces** offer a more practical way to achieve the same task, which is looking backwards instead of looking for future returns. Eligibility traces hold decaying value of value function V(s) through a function of state action. This storing of decaying value of V(s) is what allows us to look backwards. They also provide a bridge between Monte-Carlo method and temporal differences.

Formally, eligibility trace is defined as:

$$e_t(S_t) = \begin{cases} \gamma\lambda e_{t-1}(s) & if\ s \neq S_t \\ \gamma\lambda e_{t-1}(s) + 1 & if\ s = S_t \end{cases}$$

- γ is the discount rate
- λ is trace-decay parameter, [0,1]
- $e_t(S_t)$ is the eligibility trace for state *s* and time *t*

Here trace signifies the stored state, and it decays by $\gamma\lambda$ for all states and is incremented by 1 for the state that is visited. This is the reason why this is also known as accumulating trace. If $\lambda = 0$, then the method will behave as TD (0) and if $\lambda = 1$, then the method will behave as Monte Carlo, thus providing a bridge between two methods.

Conclusion

Eligibility traces can be used as a common framework that allows a way to move from a Monte Carlo method to TD method and vice-versa, by modifying the parameter λ. As the value ranges from [0,1] we can assign different weight which results in a method that is a mixture of the two. This provides flexibility to move along the spectrum with TD (0) at one end and Monte Carlo at another.

References

[1] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," IEEE Trans. Neural Netw., vol. 9, no. 5, pp. 1054–1054, 1998.

[2] http://incompleteideas.net/book/RLbook2020.pdf