CME241 Assignment11/12

Quinn Hollister

February 2022

1 TD(λ) Prediction Algorithm for Tabular

Note that I only provide the code for the tabular case, as the whole function approximation code is already in the codebase.

```
def td_lambda_tabular_prediction(
        traces: Iterable[Iterable[mp.TransitionStep[S]]],
        approx_0,
        gamma: float,
        lambd: float
) -> Iterator[ValueFunctionApprox[S]]:

    alpha = 0.03

    func_approx = approx_0
    yield func_approx

    for trace in traces:
        el_tr = defaultdict(int)
        for step in trace:

            y: float = step.reward + gamma*func_approx[step.next_state]
            update_el_tr(el_tr, gamma, lambd)
            el_tr[step.state] += 1
            gap = y - func_approx[step.state]
            update = alpha*gap
            update_value_func(func_approx, el_tr, update, step.state)
            func_approx[step.state] += update*el_tr[step.state]
            yield func_approx


def update_el_tr(el_tr_, gamma, lambd):
    for k,v in el_tr_.items():
        el_tr_[k] = v*gamma*lambd


def update_value_func(v, el_tr_, update, state):
    for k,val in v.items():
        if k != state:
            v[k] = val + update*el_tr_[k]
```

2 MC Error vs. Discounted TD Error

We are asked to prove the equality:

$$G_t - V(S_t) = \sum_{u=t}^{T-1} \gamma^{u-t} \cdot (R_{u+1} + \gamma \cdot V(S_{u+1}) - V(S_u)) \tag{1}$$

which can easily be seen by first expanding the sum and seeing that all of the intermediate Value Function's are canceled out:

$$\begin{aligned}
&\sum_{u=t}^{T-1} \gamma^{u-t} \cdot (R_{u+1} + \gamma \cdot V(S_{u+1}) - V(S_u)) = \\
&R_{t+1} + \gamma \cdot V(S_{t+1}) - V(S_t)) + \\
&\gamma \cdot (R_{t+2} + \gamma \cdot V(S_{t+2}) - V(S_{t+1})) + \\
&\gamma^2 \cdot (R_{t+3} + \gamma \cdot V(S_{t+3}) - V(S_{t+2})) + ... + \\
&\gamma^{T-t+1} \cdot (R_T + \gamma \cdot V(S_T) - V(S_{T-1})) \\
&= R_{t+1} + ... + \gamma^{T-t-1} \cdot R_T - V(S_t) + V(S_T) \\
&= G_t - V(S_t)
\end{aligned}$$

since $V(S_T) = 0$ since we are at a terminal state. Thus, we proven the equality.

3 Value Function Comparison

True Values:
NonTerminal(state=InventoryState($on_h and = 0, on_o rder = 0$)) : -35.511,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 1)) : -27.932$,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 2)) : -28.345$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 0)) : -28.932$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 1)) : -29.345$,
$NonTerminal(state = InventoryState(on_h and = 2, on_o rder = 0)) : -30.345$

MC Prediction Values:
NonTerminal(state=InventoryState($on_h and = 0, on_o rder = 0$)) : -35.503,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 1)) : -27.923$,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 2)) : -28.336$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 0)) : -28.925$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 1)) : -29.339$,
$NonTerminal(state = InventoryState(on_h and = 2, on_o rder = 0)) : -30.342$

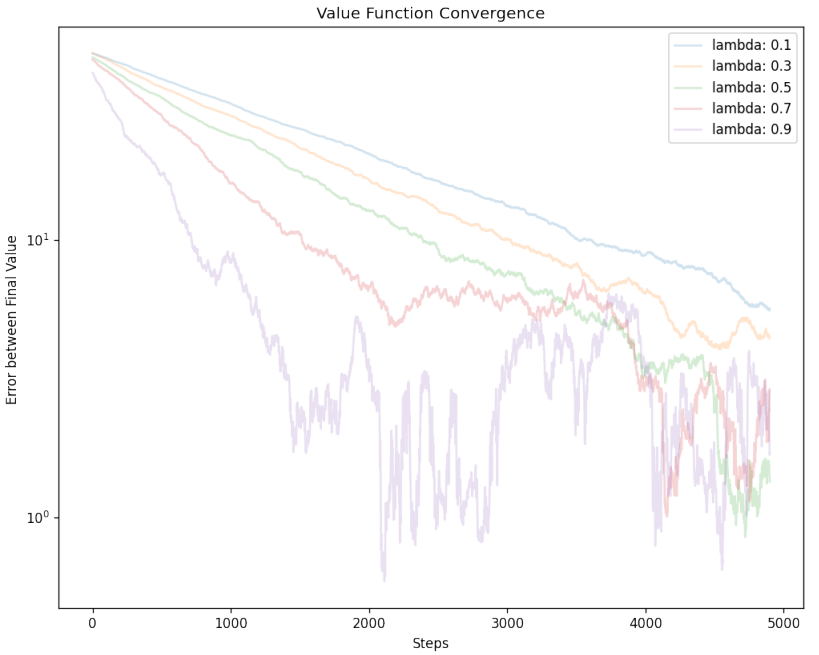
TD Prediction Values:
NonTerminal(state=InventoryState($on_h and = 0, on_o rder = 0$)) : -35.399,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 1)) : -27.942$,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 2)) : -28.245$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 0)) : -28.888$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 1)) : -29.207$,
$NonTerminal(state = InventoryState(on_h and = 2, on_o rder = 0)) : -30.205$

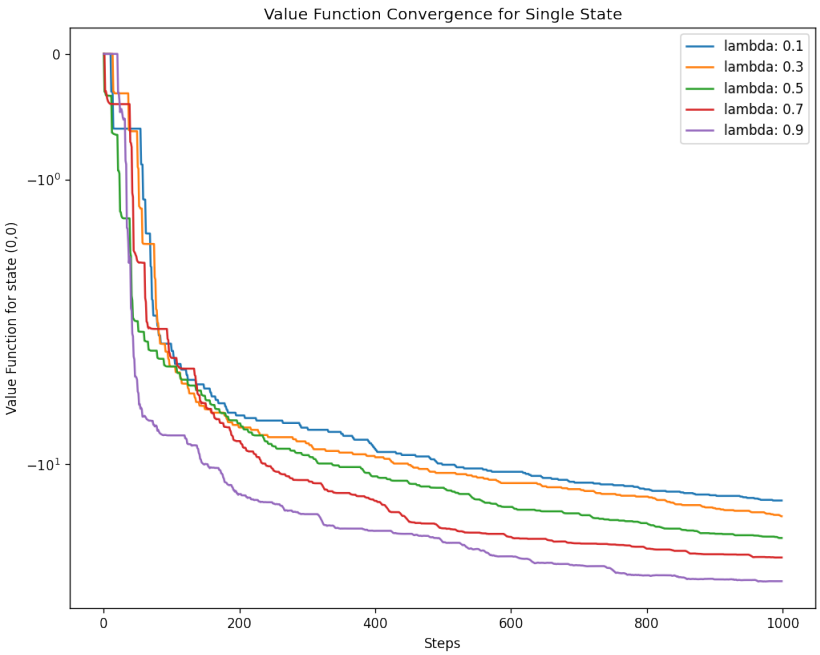
TD(λ) Prediction Values:
NonTerminal(state=InventoryState($on_h and = 0, on_o rder = 0$)) : -35.691,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 1)) : -27.934$,
$NonTerminal(state = InventoryState(on_h and = 0, on_o rder = 2)) : -28.512$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 0)) : -28.953$,
$NonTerminal(state = InventoryState(on_h and = 1, on_o rder = 1)) : -29.552$,
$NonTerminal(state = InventoryState(on_h and = 2, on_o rder = 0)) : -30.406$

As we can tell, all of these methods give generally good approximations of the value functions, out to about three orders of magnitude, while the MC will obviously give the unbiased estimate with high variance, so in this specific case, has the best approximation among the 3 approximation methods.

4 Testing on Simple Inventory

We will now look at how changing the *lambda* value affects the convergence rate of the prediction algorithm. I've provided 2 plots for 2 different convergence meanings. The first shows the convergence to the true value, and the second shows the convergence of the value function for one state.





As we can see from these plots, higher values of lambda lead to quicker conversion rates, but then the higher values hit a barrier earlier where they start to just exhibit noise, i.e. not improving their estimate

of the value function at all as it jumps around the true value. This is to be expected as higher lambda's exhibit more relation to the MC updates than TD. The lower lambda values demonstrate lower noise (i.e. lower variance) with some slight bias, again just in line with our expectations.