Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. Which of the following statements is true about function approximation in reinforcement learning? (**Select all that apply**)

1 / 1 point

- [x] It can be more memory efficient.

Correct
Correct! We cannot enumerate and store all states in a table for large or continuous state spaces. By using function approximation, we can use fewer parameters to represent the value function.

- [x] It allows faster training by generalizing between states.

Correct
Correct! Function approximation allows the agent to generalize to unseen but similar states, and can learn the value function more quickly. Furthermore, in continuous state/action spaces the agent may never see the same state twice and we need such generalization to accurately estimate the values.

- [ ] We only use function approximation because we have to for large or continuous state spaces. We would use tabular methods if we could, and learn an independent value per state.
- [x] It can help the agent achieve good generalization with good discrimination, so that it learns faster and represent the values quite accurately.

Correct
Correct. Recall the 2D plot of generalization and discrimination. Tabular methods discriminate between different states perfectly but with no generalization. Alternatively, one could treat all states as the same, with each update generalizing to all states but with no discrimination. Ideal function approximation methods achieves both good generalization and good discrimination.

2. We learned how value function estimation can be framed as supervised learning. But not all supervised learning methods are suitable. What are some characteristics of reinforcement learning that can make it harder to apply standard supervised learning methods?

1 / 1 point

- [ ] Data is available as a fixed batch.
- [x] When using bootstrapping methods like TD, the target labels change.

Correct
Correct. Targets depend on our own estimates, and these estimates change as learning progresses.

- [x] Data is temporally correlated in reinforcement learning.

Correct
Correct!

3. Value Prediction (or Policy Evaluation) with Function Approximation can be viewed as supervised learning mainly because _________. [choose the most appropriate completion of the proceeding statement]

1 / 1 point

- (●) Each state and its target (used in the Monte Carlo update, TD(0) update, and DP update) forms an input-output training example which we can use to train our approximation to the value function
- ( ) We use stochastic gradient descent to learn the value function.
- ( ) We can learn the value function by training with batches of data obtained from the agent's interaction with the world.

Correct
They can be seen as an (input, output) training example with (S_t, G_t) for Monte Carlo update, $(S_t, R_{t+1} + \gamma V_\pi(S_{t+1}))$ for TD(0) update, and $(s, E_\pi(R_{t+1} + \gamma V_\pi(S_{t+1})|S_t = s))$ for DP update. Each of these updates makes the output of value function of S_t more like the target value.

4. Which of the following is true about using Mean Squared Value Error ($\overline{VE} = \sum \mu(s)[v_\pi(s) - \hat{v}(s,w)]^2$) as the prediction objective?

1 / 1 point

$\mu(s)$ represents the weighted distribution of visited states

(Select all that apply)

- [x] This objective makes it explicit how we should trade-off accuracy of the value estimates across states, using the weighting μ.

Correct
Correct. $\mu(s)$ is a weighting of how much we care about the error in state s, and we usually choose $\mu(s)$ to be the fraction of time we spend in state s.

- [x] The agent can get zero MSVE when using a tabular representation that can represent the true values.

Correct
Correct. In fact, in the tabular setting, we did not define an objective because we did not need to. With

Correct. In fact, in the tabular setting, we did not define an objective because we did not need to. With a table of values, we can represent the true value function exactly. So we do not need an objective to help specify how to trade-off accuracy.

- [x] Gradient Monte Carlo with linear function approximation converges to the global optimum of this objective, if the step size is reduced over time.

Correct
Correct. There are stronger theoretical guarantees with linear function approximation than with non-linear function approximation.

- [ ] Even if the agent uses a linear representation that **cannot represent** the true values, the agent can still get zero MSVE.

5. Which of the following is true about $\mu(S)$ in Mean Squared Value Error? (**Select all that apply**) 1 / 1 point

- [x] It has higher values for states that are visited more often.

Correct
Correct.

- [x] It is a probability distribution.

Correct
Correct.

- [ ] If the policy is uniformly random, $\mu(S)$ would have the same value for all states.
- [x] It serves as a weighting to minimize the error more in states that we care about.

Correct
Correct.

6. The stochastic gradient descent update for the MSVE would be as follows. 1 / 1 point

Fill in the blanks (A), (B), (C) and (D) with correct terms. (**Select all correct answers**)

$$\mathbf{w_{t+1}} \doteq \mathbf{w_t}\ (A)\ \tfrac{1}{2}\alpha\nabla[\ (C)\ -\ (D)\ ]^2$$

$$= \mathbf{w_t}\ (B)\ \alpha[\ (C)\ -\ (D)\ ]\nabla\hat{v}(S_t, \mathbf{w_t})$$

$(\alpha > 0)$

- [x] $-, -, \hat{v}(S_t, \mathbf{w_t}), v_\pi(S_t)$

Correct
Correct! stochastic gradient descent makes update to $\mathbf{w_t}$ proportional to the negative gradient of the squared error.

- [ ] $+, +, \hat{v}(S_t, \mathbf{w_t}), v_\pi(S_t)$
- [x] $-, +, v_\pi(S_t), \hat{v}(S_t, \mathbf{w_t})$

Correct
Correct! stochastic gradient descent makes update to $\mathbf{w_t}$ proportional to the negative gradient of the squared error.

- [ ] $+, -, v_\pi(S_t), \hat{v}(S_t, \mathbf{w_t})$

7. In a Monte Carlo Update with function approximation, we do stochastic gradient descent using the following gradient: 1 / 1 point

$$\nabla[G_t - \hat{v}(s, \mathbf{w})]^2 = 2[G_t - \hat{v}(s, \mathbf{w})]\nabla(-\hat{v}(S_t, \mathbf{w}_t))$$

$$= (-1) * 2[G_t - \hat{v}(s, \mathbf{w})]\nabla\hat{v}(S_t, \mathbf{w}_t)$$

But the actual Monte Carlo Update rule is the following:

$$\mathbf{w_{t+1}} = \mathbf{w_t} +\ \alpha[G_t - \hat{v}(S_t, \mathbf{w}_t)]\nabla\hat{v}(S_t, \mathbf{w}_t), \qquad (\alpha > 0)$$

Where did the constant -1 and 2 go when α is positive? (**Choose all that apply**)

- [ ] We assume that the 2 is included in $\nabla\hat{v}(S_t, \mathbf{w}_t)$.
- [x] We assume that the 2 is included in the step-size.

Correct
Correct. It is equivalent to use α or 2α, because we select α. If we want to use an α of 0.1 for the gradient with a 2 in front, then it is equivalent to use an α of 0.2 without a 2 in front of the gradient.

- [ ] We are performing gradient ascent, so we subtract the gradient from the weights, negating -1.
- [x] We are performing gradient descent, so we subtract the gradient from the weights, negating -1.

Correct
Correct.

8. When using stochastic gradient descent for learning the value function, why do we only make a small update 1 / 1 point

towards minimizing the error instead of fully minimizing the error at each encountered state?

- (selected) Because we want to minimize approximation error for all states, proportionally to μ.
- Because the target value may not be accurate initially for both TD(0) and Monte Carlo method.
- Because small updates guarantee we can slowly reduce approximation error to zero for all states.

Correct
Correct! With function approximation, the agents have limited capacity and minimizing the approximation error for one state invariably increases the error for other states. We want to make small updates so that the error is reduced across states, proportionally to the weighting μ.

9. The general stochastic gradient descent update rule for state-value prediction is as follows: 1 / 1 point

$$\mathbf{w_{t+1}} \doteq \mathbf{w_t} + \alpha[U_t - \hat{v}(S_t, \mathbf{w_t})]\nabla\hat{v}(S_t, \mathbf{w_t})$$

For what values of U_t would this be a semi-gradient method?

- $v_\pi(S_t)$
- G_t
- (selected) $R_{t+1} + \hat{v}(S_{t+1}, w_t)$
- $R_{t+1} + R_{t+2} + \ldots + R_T$

Correct
Correct. This is the typical TD(0) bootstrapping target, which depends on the current weight vector $\mathbf{w_t}$. It will not produce a true gradient estimate, because its expected value is not equal to true v_π.

10. Which of the following statements is true about state-value prediction using stochastic gradient descent? 1 / 1 point

$$\mathbf{w_{t+1}} \doteq \mathbf{w_t} + \alpha[U_t - \hat{v}(S_t, \mathbf{w_t})]\nabla\hat{v}(S_t, \mathbf{w_t})$$

(**Select all that apply**)

- [x] Semi-gradient TD(0) methods typically learn faster than gradient Monte Carlo methods.

Correct
Correct! Similar to the tabular case, Semi-gradient TD(0) methods learn faster than gradient Monte Carlo methods.

- [x] Using the Monte Carlo return as target, and under appropriate stochastic approximation conditions, the value function will converge to a local optimum of the Mean Squared Value Error.

Correct
Correct! Monte Carlo return (G_t) is an unbiased estimate of $v_\pi(S_t)$. It converges to a stationary point, which under mild conditions, will be a local optimum of the MSVE.

- [ ] Stochastic gradient descent updates with Monte Carlo targets always reduce the Mean Squared Value Error at each step.
- [x] Using the Monte Carlo return or true value function as target results in an unbiased update.

Correct
True. The stochastic update with either target is an unbiased estimate of the gradient of the MSVE.

- [x] When using $U_t = R_{t+1} + \hat{v}(S_{t+1}, \mathbf{w_t})$, the weight update is not using the true gradient of the TD error.

Correct
Correct! When computing the gradient of the TD error, we do not consider the effect of changing the weight vector $\mathbf{w_t}$ in the bootstrapped target U_t.

11. Which of the following is true about the TD fixed point? 1 / 1 point

(**Select all correct answers**)

- [ ] The weight vector corresponding to the TD fixed point is a local minimum of the Mean Squared Value Error.
- [x] Semi-gradient TD(0) with linear function approximation converges to the TD fixed point.

Correct
Correct! This is the definition of TD fixed point.

- [ ] The weight vector corresponding to the TD fixed point is the global minimum of the Mean Squared Value Error.
- [x] At the TD fixed point, the mean squared value error is not larger than $\frac{1}{1-\gamma}$ times the minimal mean squared value error, assuming the same linear function approximation.

Correct
Correct! See Equation (9.14) from the textbook.

12. Which of the following is true about Linear Function Approximation, for estimating state-values? (Select all that apply) 1 / 1 point

- [x] The gradient of the approximate value function $\hat{v}(s, \mathbf{w})$ with respect to $\mathbf{w}$ is just the feature vector.

Correct
Correct. In linear function approximation, the value function is a linear combination of the weight vector and the feature vector. $\hat{v}(s, \mathbf{w}) = \mathbf{w}^T \mathbf{x}(s)$. By taking the gradient with respect to $\mathbf{w}$, the gradient is the feature vector $\mathbf{x}(s)$.

- [x] State aggregation is one way to generate features for linear function approximation.

Correct
Correct.

- [ ] The size of the feature vector is not necessarily equal to the size of the weight vector.