Presentation:

Slide - 1

In this presentation, we will discuss the use of Temporal Difference (TD) methods in reinforcement learning. We will start with a brief introduction to TD methods, including TD(lambda) and Least Squares TD algorithms with Linear Function Approximation (LFA), and compare their strengths and weaknesses.

Next, we will introduce the Accelerated Gradient TD algorithm, which is a more recent TD method that has shown significant improvements in learning speed and sample efficiency compared to previous methods and we demonstrate its implementation on two standard reinforcement learning environments - Boyan's chain and Random walk.

(sample efficiency refers to the ability of an algorithm to learn a policy using as few samples (or interactions with the environment) as possible.)

Slide - 2

Temporal Difference (TD) methods are a class of reinforcement learning algorithms used to estimate the value function of a policy in a Markov Decision Process (MDP).

One of the main advantages of TD methods is their ability to learn from experience in an online, incremental manner, which means that they can learn from data as it arrives in a streaming fashion.

TD methods can be categorized into linear methods like TD(λ), and more sophisticated least squares methods that directly compute the TD solution.

Moreover, TD methods are computationally efficient and have been shown to perform well in a variety of reinforcement learning settings.

Slide – 3

Its just an overview of the setting that we are using these algorithms on.

The variables in these equations have their usual meanings.

Gt is the discounted sum of future rewards

X(t) is the feature vector with w as its coefficient matrix that has the weights assigned to each feature

Slide – 4

Instead of waiting till the end of trajectory, TD methods work by updating the estimated value of a state based on the difference between the expected reward for that state and the reward actually received. This difference is known as the temporal difference error (delta t), and it is used to update the estimated value of the state in a way that minimizes the error.

Zt is the eligibility trace vector

These are the update equations for TD(lambda) algorithm

(eligibility trace vector is a vector used in TD(lambda) methods to assign credit to states and actions based on their contribution to the final reward. It is updated at each time step by decaying its previous value and adding the current state-action pair, and is used to update the weights of the linear function approximator to improve the accuracy of the value function approximation.)

Slide – 5

LSTD is a model-free algorithm that estimates the value function by solving a set of linear equations

The key idea of LSTD is to use the method of least squares to minimize the mean-squared error between the predicted values and the observed returns.

Since, stationary distribution is not known, A and b are approximated as follows

Then we can take A-1 b and get the solution

We implemented LSTD by updating At and bt on incremental basis using Incremental Truncated LSTD algoritm

Slide – 6

Computational Cost: Linear TD(λ) is computationally cheaper than Least Squares TD because it updates the value function incrementally at each time step, while LFA Least Squares TD requires computing a matrix inverse to estimate the value function

Memory Requirements: Linear TD(λ) requires less memory than Linear Least Squares TD because it updates the value function incrementally and only requires storing the eligibility trace vector, while LFA Least Squares TD requires storing the entire batch of experience and the inverse of a matrix

Slide – 7

The motivation for accelerated TD learning comes from the fact that both TD(λ) and least squares TD have their strengths and weaknesses. TD(λ) can capture long-term dependencies in the data and update the value function incrementally, but it introduces bias. Least squares TD can estimate the value function using regression algorithm in quadratic computation and storage, but it can introduce high variance and require storing large amounts of data.

Accelerated TD learning aims to combine the benefits of TD(λ) ayn nd least squares TD to achieve faster convergence, improved stability, better generalization, and lower memory requirements. By combining TD(0) updates with least squares regression updates, accelerated TD learning can achieve a balance between bias and variance, capture both short-term and long-term dependencies in the data, and update the value function incrementally using a single weight vector.