Reinforcement Learning

Table of Contents

[Model-Based Learning 2](#_Toc138878237)

[Model-Free Learning 3](#_Toc138878238)

[Direct Evaluation 4](#_Toc138878239)

[Temporal Difference Learning 5](#_Toc138878240)

[Exponential Moving Average 6](#_Toc138878241)

[Active Reinforcement Learning 7](#_Toc138878242)

[Q-Learning 8](#_Toc138878243)

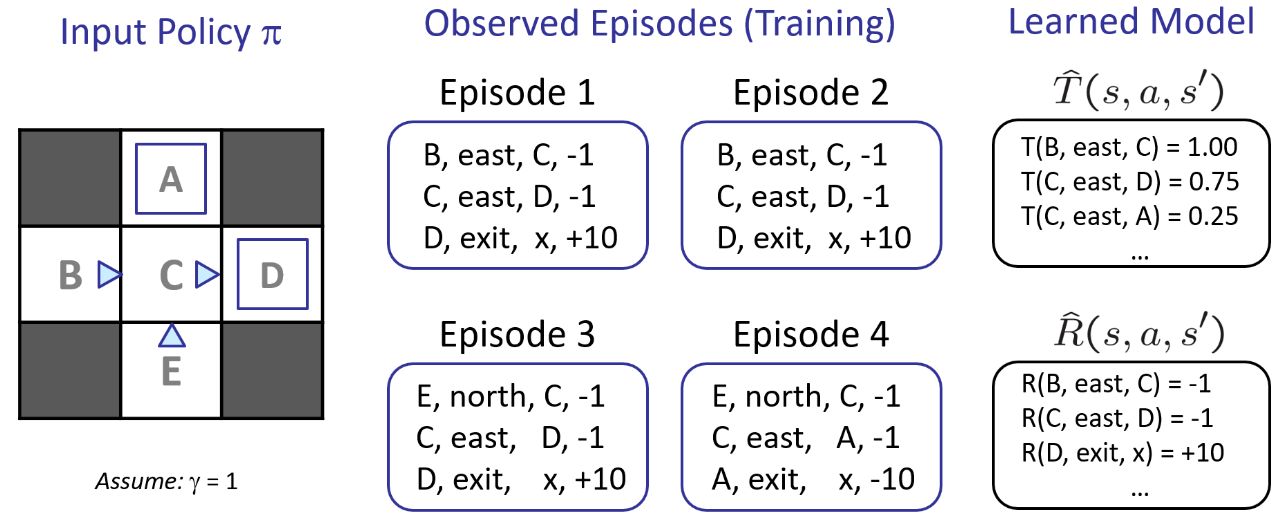
[Properties 8](#_Toc138878244)

In **Reinforcement Learning**, we do not know the transition function or the reward. As agents, we take some action from the state that we are in. The environment then takes us to a new state and also gives us some reward (not necessarily positive). We must learn to behave in a manner so as to maximize the expected reward. We do not know which states are ‘good’ and we do not know what will happen when we take an action. So we find out.

## Model-Based Learning

There are two forms of reinforcement learning, the first being **Model-Based Learning**. Here, we create a model based on our experience and then solve the problem as though it were an MDP using the model we just created.

Suppose we have some random actions decided for each state in our initial policy. We take this policy and run policy evaluation on it. Each iteration is called an **episode**, and the result of all the episodes can be used to calculate some transition probabilities and some rewards.



As an example, suppose we want to calculate the expected age of students in a class. The ideal situation would be if we knew the possible values of age and also the probability of each occurring.

If we do not know the values of (the rewards) or the probabilities with which they occur (the transition probabilities), we can instead take samples from the students and then calculate the values ourselves.

This might not be perfect at first, but if we take a large enough number of samples, it will be pretty close to exactly correct.

## Model-Free Learning

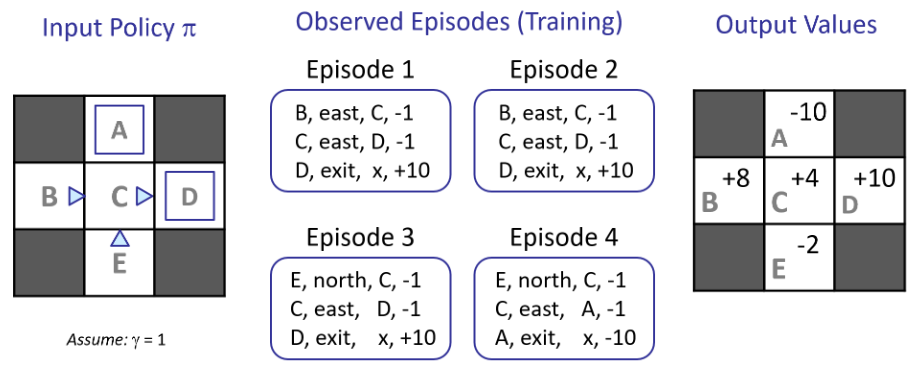
The above example, when used in Model-Free Learning, becomes as simple as taking the average value.

This process works because the probability of picking some student to check their age (which is what the expected age is giving us) is the same as the probability distribution of the ages themselves.

Thus, instead of finding the probabilities of each action occurring, in Model-Free Learning, we are directly finding the outcome. There are two variants of Model-Free Learning, Passive Reinforcement Learning and Active Reinforcement Learning.

### Direct Evaluation

In **direct evaluation**, we are given a policy, but not told about any transition functions or rewards. Our goal is to just follow the policy and obtain the state values. We do this by running **multiple episodes** and taking the average value of each state.



Notice that in the example above, the states and have different values, even though we can objectively tell that they should be the same. This issue occurs because we are not considering the connections between states.

At this point, we might start to wonder why we are not using **policy evaluation**. The reason is that policy evaluation required us to know and , which we do not in this case.

However, going back to the idea of calculating the expected age of students, we can get rid of and simply take the average of several samples.

However, there are situations where this mechanism will fail us, such as ones where it is not possible to take multiple samples, i.e., situations where the cost of failure is too high.

### Temporal Difference Learning

Since it might be difficult to take multiple samples from the same state, an alternative is to update the value of every time we experience a transition from that state. We are taking the **running average**, so the existing value of is added to the sample value. We give the new sample a low weight. Over a large number of iterations, the outcomes from the child states which have a higher probability of occurring will cause the majority of the updates, which will result in the behaviour of the transition function being replicated.

Sample of :

Update to :

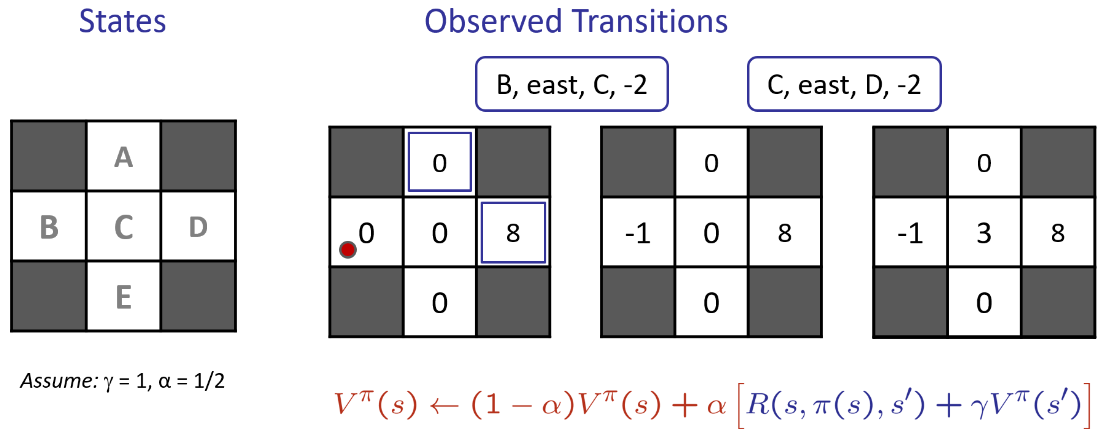
Sample update:

This process is called **Temporal Difference Learning**.

### Exponential Moving Average

Taking the running average for the value of a state makes the recent values more important. Distant values have exponentially lower weights, since those values were wrong in any case.

In addition to this, decreasing the learning rate () over time causes values to converge.



## Active Reinforcement Learning

The problem with the mechanism outlined above is that we cannot obtain a policy from it. This is because a policy involves taking the action at each state which has the maximum -Value. The -Value is still calculated in terms of and .

To get around this, we can learn the -values instead of the values. This is called **Active Reinforcement Learning**. The naming indicates an important change. If the -values are updating in real time, our actions are also updating in real time.

### Q-Learning

In -value iteration, the values are calculated using the equation below:

Similar to sample-based value iteration, we perform sample-based -value iteration by taking samples at each transition and updating the -values based on it. This is called **Q-Learning**.

### Properties

Q-Learning converges to the optimal policy, even if the actions are random. This is called **off-policy learning**. The caveats to this are that we have to make sure that our agent is exploring enough and we have to eventually make the learning rate small enough to allow it to converge, while also making sure to not decrease it too quickly.