Markov Decision Processes II

Table of Contents

[Problems with Value Iteration 2](#_Toc138878274)

[Policy Evaluation 2](#_Toc138878275)

[Policy Extraction 3](#_Toc138878276)

[Policy Iteration 4](#_Toc138878277)

[Double Bandits 4](#_Toc138878278)

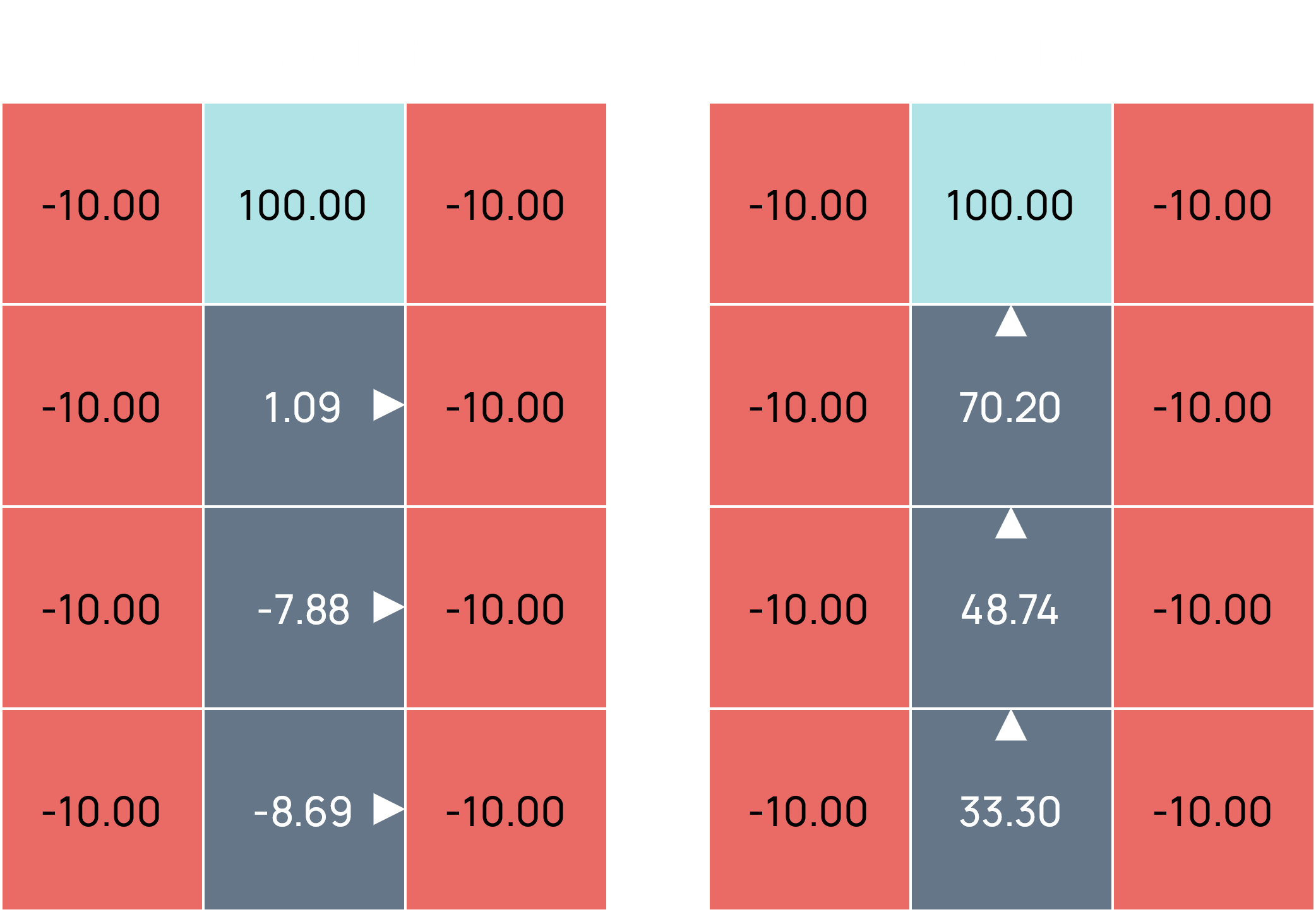
## Problems with Value Iteration

The main issue with value iteration is that it is very slow. The time complexity is . After a few iterations however, the maximum value for each state barely changes, whereas the action does not change at all. This means that the **policy** converges far sooner than the values do.

Instead of performing value iteration, we can shift to other methods which disregard the actual values and only concentrate on the policies.

## Policy Evaluation

The first mechanism is **Policy Evaluation**. Instead of finding a policy ourselves, we will take a policy that is provided to us, , and simply follow what the policy says to find the utility of each state, .



To find the values for each state, we follow the same process as value iteration, i.e., we run the policy repeatedly and recalculate the values until they converge. Each iteration has a time complexity of .

Without the calculations related to finding the maximum values, this system is just a linear system.

## Policy Extraction

Instead of finding the values of different states given a specific policy, **policy extraction** concentrates on finding the appropriate actions given the values of different states. To do this, we use a process called **mini expectimax**, where we only go one step deep, i.e., we only check the values of the immediate neighbours of a state to determine which action to take at that state.

Instead of using the values of the neighbouring states, we can also use the Q-values of the current state.

This is considerably easier to do.

## Policy Iteration

**Policy Iteration** involves both policy evaluation and policy extraction. We start with some random policy, evaluate it to get the values, use those values to extract a new policy and repeat the process until the policy converges. This converges far faster than value iteration while still being optimal.

Both value iteration and policy iteration are dynamic programming approaches to solving MDPs.

## Double Bandits

Suppose we have two slot machines, a blue one and a red one. The blue slot machine is guaranteed to give us $1 every time we play it, while the red slot machine will give us $2 75% of the time and $0 25% of the time.

Given this information, we can perform some computations to decide whether it is more beneficial to play the red slot machine or the blue one in the long run. This is essentially what we are doing when we solve an MDP. This process is called **offline planning**, since we do not actually need to play the game to perform the computations.

On the other hand, we can have a situation in which we do not know the probability with which we will win or lose on either machine. In this case, we have to actually play the game a few times and then decide on a strategy. This is called **online planning**. We are making a decision based on the behaviour of the system. This is **reinforcement learning**.

There are a few key ideas of reinforcement learning that came up in this situation:

* **Exploration** – The actions for which information was unknown had to be explored to get information about them.
* **Exploitation** – Based on the information we gather when exploring, we create a strategy that is beneficial to us.
* **Regret** – Even if we learn intelligently, we might make mistakes. That is how we learn.
* **Sampling** – Since the results of the actions are based on chance, actions had to be tried repeatedly.
* **Difficulty** – The learning process is significantly more difficult than just solving an MDP.