e-PGPathshala

Subject: Computer Science

Paper: Machine Learning

Module: Basics of Reinforcement Learning-I

Module No: CS/ML/37

Quadrant 1- e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning. In this module we will discuss the basics of reinforcement learning.

Learning Objectives

The learning objectives of the module are as follows:

- A Basic Introduction to Reinforcement Learning
- To explain the Elements of Reinforcement Learning
- To discuss the applications of ReinforcementLearning

37.1 Reinforcement learning

Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards. For example, consider teaching a monkey a new trick: you cannot instruct the monkey what is to be done, but you can reward (a banana) or punish (scold) it depending on whether it does the right/wrong thing. Here the monkey has to figure out what it did that made it get the reward or punishment, which is known in reinforcement learning context as the credit assignment problem. Learningtakesplaceasa resultofinteraction between an agentand the world. In other words percept received by an agent should be used not only for understanding, interpreting or prediction, as in the machine learning tasks we have discussed so far, butalsoforacting. Reinforcement learning is more general than supervised and unsupervised learning and learn from interaction with the environment to achieve a goal and getting an agent to act in the world so as to maximize its rewards. It allows agents to automatically determine the ideal behaviour within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behaviour; this is known as the reinforcement signal.

The motivation behind reinforcement learning is that it allows an agent to learn its behavior based on feedback from the environment. This behavior can be learnt once and for all, or keep on adapting as time goes by. If the problem is modelled with care, some reinforcement learning algorithms can converge to the global optimum; this is the ideal behavior that maximizes the reward.

It is a trial-and-error learning paradigm which learns from rewards and punishments. Reinforcement learning is not just an algorithm but a new paradigm in itself. Its objective is to learn about a system from minimal feedbacklike its behavior, control. It is inspired by behavioral psychology.

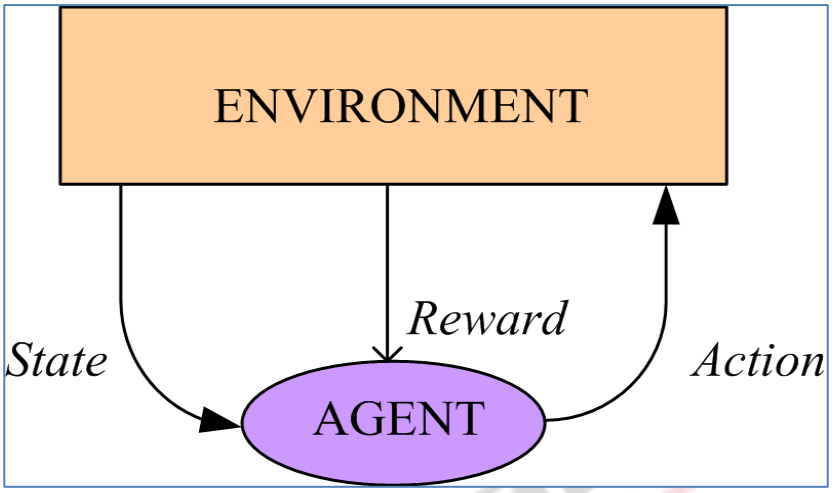


Figure: 37.1

As mentioned in figure 37.1, the reinforcement learning consists of the agent and the environment. The agent performs the action under the policy being followed and the environment is everything else other than the agent. Reinforcement learning is learning from interaction and it is a goal-oriented learning. It is learning about, from, and while interacting with an external environment. It is a learning which tells you, what to do (what action to take), how to map situations to actions so as to maximize a numerical reward signal.

The general definition of reinforcement learning is as follows "Reinforcement learning is learning what to do — how to map situations to actions — so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them."

37.2 Difference between Reinforcement, Supervised and Unsupervised learning

In supervised learning, the training information contains the desired (target) outputs. In reinforcement learning the training information contains evaluations (“rewards” / “penalties”) and the output are actions of the agent (Figure 37.2). In supervised learning we get the target output and if we subtract the actual output from the target output we will get the error of the system but that is not the case with reinforcement learning. In reinforcement learning the objective is to get as much reward as possible.

37.2.1 Supervised Learning

In supervised learning we have a situation in which sample (input, output) pairs of the function to be learned can be perceived or are given. We can think as if there is a kind teacher who makes training data: (X,Y). (features, label) available and the job of the learning system is to predict Y, minimizing some loss.

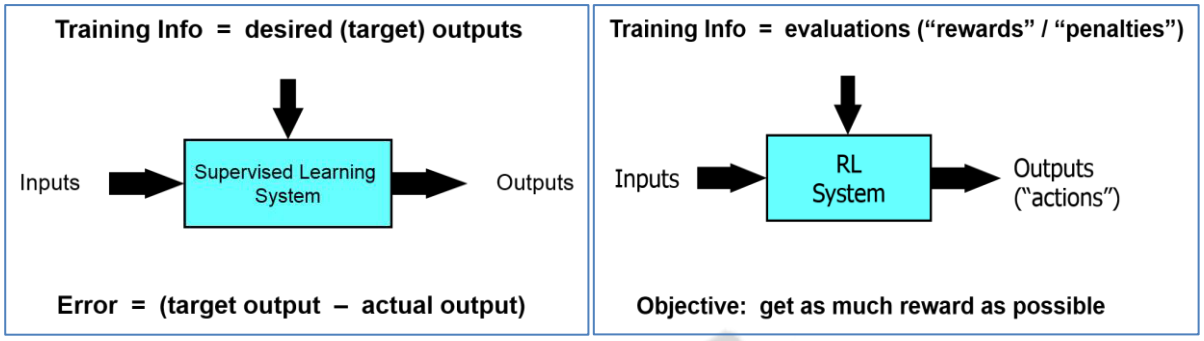


Figure 37.2 Comparison Of Supervised Learning and Reinforcement Learning

37.2.2 Unsupervised Learning

In unsupervised learning the training data contains X (the features only) and the job of the learning system is to find "similar" points in high-dimensional X-space.

37.2.3 Reinforcement Learning

In reinforcement learning the agent acts on its environment, it receives some evaluation of its action (reinforcement), but is not told of which actionisthecorrectonetoachieveitsgoal. The training data: (S, A, R). (State-Action-Reward) and the learning system needs to develop an optimal policy(sequence of decision rules) so as to maximize its long-term reward.

37.3 General Reinforcement Learning Algorithm

The first step is to initialise the learner's internal state. Then the algorithm should repeat forever the following steps

1.Observe current state *s*

2.Choose action *a* using some evaluation function

3.Execute action *a*

4. Let *r* be immediate reward, *s'* new state

5.Update internal state based on *s,a,r,s'.*

Here learning is concerned with what to do that is how to map situations to actionsso as to maximize a numerical reward signal. Therefore learning to choose an action based on an evaluation function is the learning part.

37.4 Key Features of Reinforcement Learning

The following are the key features of reinforcement learning. The learner is not told which actions to take and in that sense it is a trial-and-error search. It has the possibility of delayed reward (sacrifice short-term gains for greater long-term gains). In reinforcement learning there is a need to explore and exploit. It considers the whole problem of a goal-directed agent interacting with an uncertain environment. The reinforcement model assumes that each percept(*e*) is enough to determine the State(the state is accessible) and the agent can decompose the Reward component from a percept. Therefore the agent task: to find a optimal policy, mapping states to actions, that maximize long-run measure of the reinforcement.

37.5 Agent-Environment Interface

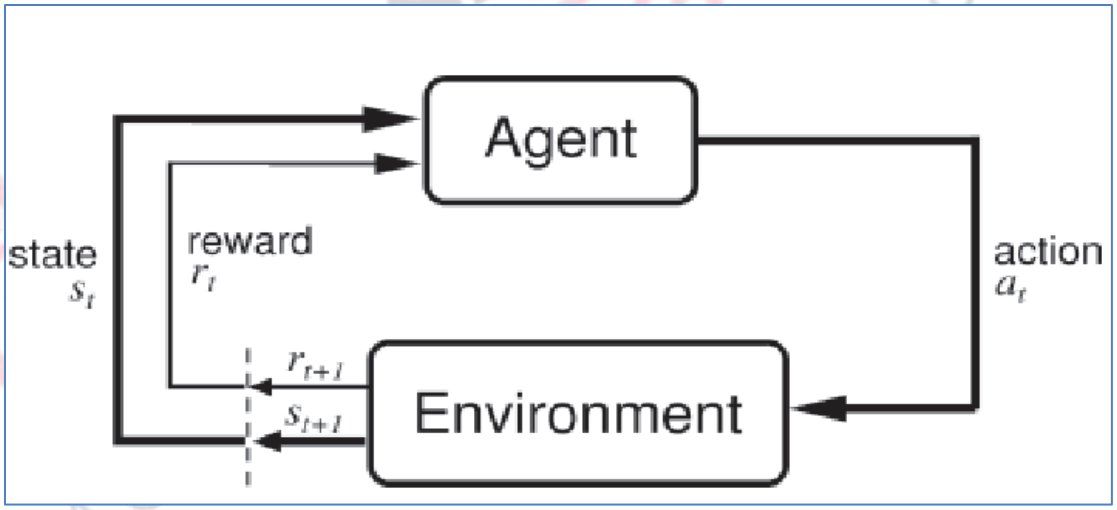


Figure 37.3 Agent-Environment Interface

We have the agentenvironment interface as shown in Figure 37.3. Associated with this interface is $\mathbf{s_t}$ which is the state at time t, $\mathbf{r_t}$which is the reward by the environment to the agent at time t and $\mathbf{a_t}$ which is the action taken by the agent at time t. After taking the action at time t, the environment goes to the next state $\mathbf{s_{t+1}}$ at time t+1, and gives the reward $\mathbf{r_{t+1}}$at time t+1.

The task of reinforcement learning is to learn how to behave successfully so as to achieve a goal while interacting with an external environment and learn through experience from trial and error. Some examples of reinforcement learning are:

- Game playing: The agent knows it has won or lost, but it doesn't know the appropriate action in each state
- Control: a traffic system can measure the delay of cars

37.6 Elements of Reinforcement Learning

The elements of reinforcement learning are shown in Figure 37.4. They are:

1. Policy: what to do
2. Reward: what is good- defines the goal in a reinforcement learning problem and gives the agent a sense of what is good in an immediate sense
3. Value: what is good because it predicts reward -The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state in other words gives the agent a sense of what is good in the long run.
4. Model: what follows what -used to predict the states the environment will be in after the agent performs its actions- the agent often uses the model to compute series of potential state–action sequences

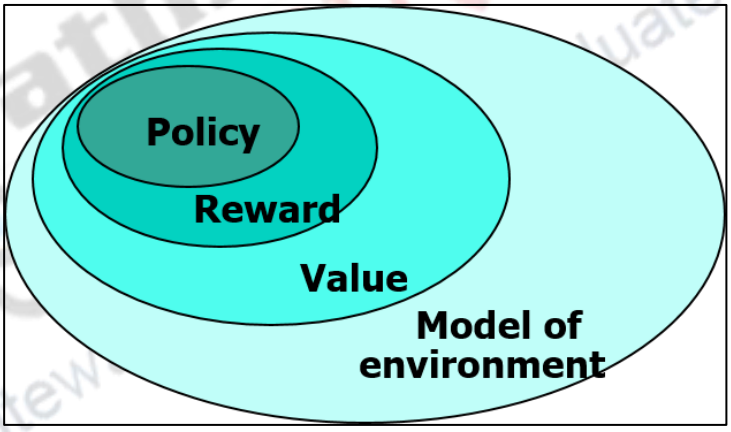


Figure 37.4 Elements of RL

Figure 37.5

We have the environment and a state given to the agent. The agent does an action based on the policy. The action effects the environment. The environment moves to another state and gives you a reward. Figure 37.5shows this transition model, (i.e.) how action influence states. Given the state and the action, we have the reward R, the immediate value of state-action transition and we have the policyπ which maps states to actions.

Let us take an example shown in Figure 37.6, which has the immediate reward values and we explain about the value function for this example. In the figure, the immediate reward values are the numbers like 90, 100. They are the reward given to the action chosen by the agent. The actions of the agent are indicated by different arrows namely upward, downward, left, right. The Figure shows us the environment, the agent is acting upon. Initially we assume that all states have values of 0 except move up from state f to c to goal and stateb to c which have value 100 where we assume c is the goal. Now in the next step assuming that discount factor γ= 0.9 we can calculate value function of state from and to states a and c as 90 and to and from state d as 81 using equation given.

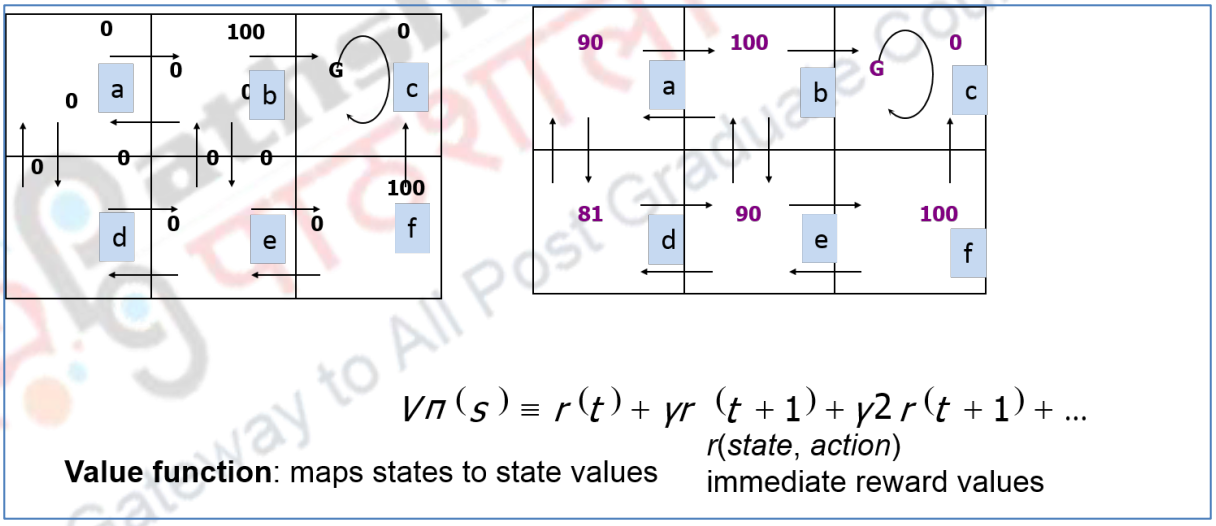


Figure 37.6

The equation for the value function is given below. It maps the states to state values.

$$V\pi(s) \equiv r(t) + \gamma r(t+1) + \gamma 2\, r(t+1) + \ldots$$

For each action, we go to a different state and the environment gives reward. This gets added up for each action, until we reach the final position.

37.7 Reinforcement model

In the reinforcement model, each percept (e) is enough to determine the State (the state is accessible). The agent can decompose the Reward component from a percept. The agent task is to find an optimal policy, mapping states to actions that maximize long-run measure of the reinforcement. We can think of reinforcement in terms of reward that the agent gets. It can be modeled as MDP **(Markov Decision Process)** model.

37.7.1 Review of MDP model

In Markov Decision Process the transitions are probabilistic and the observation = state. The assumption is that reward and next state are (probabilistic) functions of current observation and action only. The goal is to learn a good strategy for collecting reward, rather than necessarily to make a model.Markov Decision Process model consists of four components namely S, T, A and R. Here S is the set of states; A is the set of actions; T is the probability of transition from s to s' given action a and is written as T(s,a,s') = P(s'|s,a).

S – set of states of the environment

$A(s)$ – set of actions possible in state $s \varepsilon S$

$P(s,s',a)$ – probability of transition from s to s' given a

$R(s,s',a)$ – expected reward on transition s to s' given a

γ– discount rate for delayed reward

discrete time, $t = 0, 1, 2, \ldots$ (Figure 37.6)

R(s,s',a) is the expected reward for taking action a in state s and it is given by the following formula.

$$R(s,s',a) = \sum_{s'} P(s'|\, s,a) r(s,a,s')$$

$$R(s,s',a) = \sum_{s'} T(s,a,s') r(s,a,s')$$

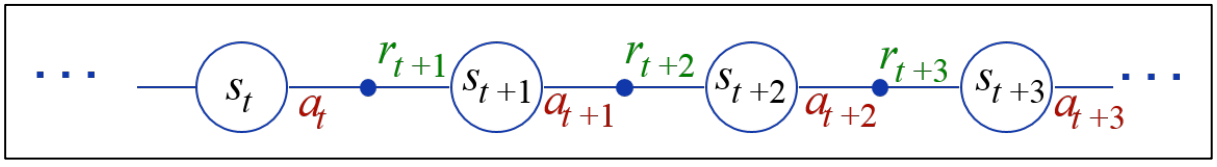


Figure 37.6 MDP model of Reinforcement Learning

The reinforcement learning framework tells about how to learn from close interaction in a stochastic environment. It gives anoisy delayed scalar evaluation and itmaximizes the long term performance of the system.

37.7 Elements of Reinforcement Learning (Markov Decision Process)

From MDP model view point, the elements of reinforcement learning can be stated as follows,

- s_t : State of agent at time t
- a_t : Action taken at time t
- In s_t ,action$a_t$is taken and the clock ticks. Then the reward r_{t+1} is received and state changes to s_{t+1}.
- The Next state probability is given by P (s_{t+1}| s_t , a_t) given current state and current action
- Reward probability is given by p (r_{t+1} | s_t , a_t)
- The Initial state(s) and the goal state(s) will be known and we have an episode which is the trial of actions from initial state to goal state.

As we have already discussed reinforcement learning is not a supervised learning. We have a very sparse "supervision" as the target output is not provided. We do not have the error gradient information available as with the supervised learning. Here the action chooses the next state and we explore to estimate the gradient through trial and error learning.Pattern detection is not the primary goal of reinforcement learning.

37.8 The Gambling example

In this example, we toss 3 different biased coins. The coin to be tossed is selected randomly from the three options and we always see which coin we are going to play next. We make bets on head or tail and the wage is always $1. If we win we get $1, otherwise we lose our bet. The reinforcement model for this example is

Input: X – a coin chosen for the next toss,

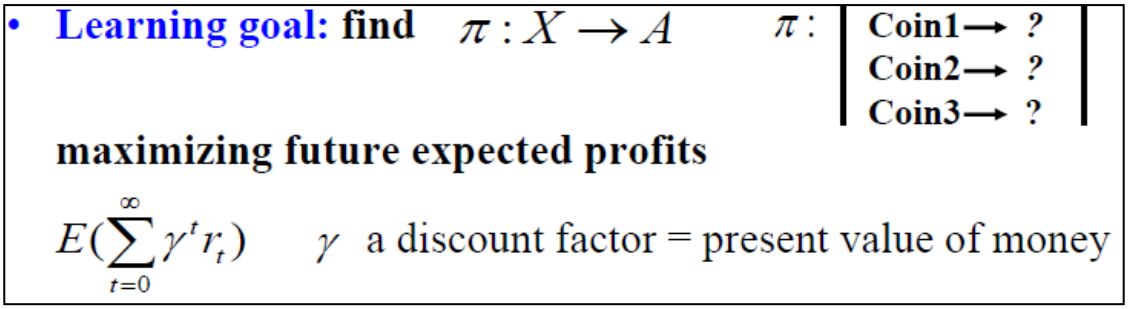
Action: A – choice of head or tail,

Reinforcements: {1, -1}

- **A policy** $\pi : X \rightarrow A$

  Example: π : | Coin1 → *head*
Coin2 → *tail*
Coin3 → *head* |

The learning goal for the example is given below.



The example has been taken from Artificial Intelligence: A Modern Approach Russell and Norvig

37.9Model based and model free approach of reinforcement learning

In the model based approach, we learn the model and use it to derive the optimal policy. Example is the Adaptive dynamic learning (ADP) approach. In the model free approach we derive the optimal policy without learning the model. Examples include LMS and Temporal difference approach

37.10Applications of Reinforcement Learning

- Robot navigation
- Adaptive control

  e.g. Helicopter pilot!
- Combinatorial optimization

  e.g. VLSI placement and routing , elevator dispatching
- Game playing

  e.g. Backgammon – world's best player!
- Computational Neuroscience

  e.g. Modeling of reward processes

Some other Notable Reinforcement Learning Applications are

- **TD-Gammon**: Tesauro
  - world's best backgammon program
- **Elevator Control**: Crites &Barto
  - high performance down-peak elevator controller

- **Dynamic Channel Assignment**: Singh &Bertsekas, Nie&Haykin
  - high performance assignment of radio channels to mobile telephone calls

37.13 Robot in a Room Example

We take an example of robot in a room that has starting position as given in Figure 37.7. The actions that can be taken by the agent are UP, DOWN, LEFT, RIGHT. In this example we bring in the probability of each action. The agent moves UP 80% of time,moves LEFT 10% of time, moves RIGHT 10% of time. We are given the information that the reward is**+1 at [4,3], -1 at [4,2].**The reward for each step is -0.04. We have to find the strategy to achieve the max reward.

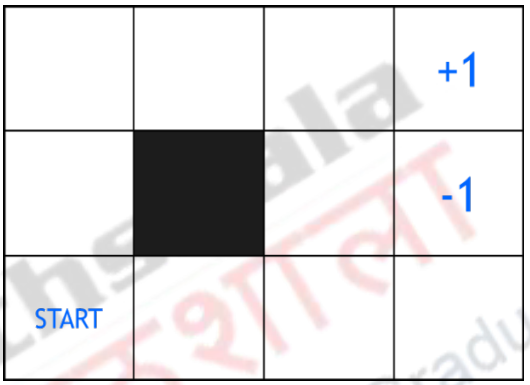


Figure 37.7 Starting Board Position

Optimal Policy Solution

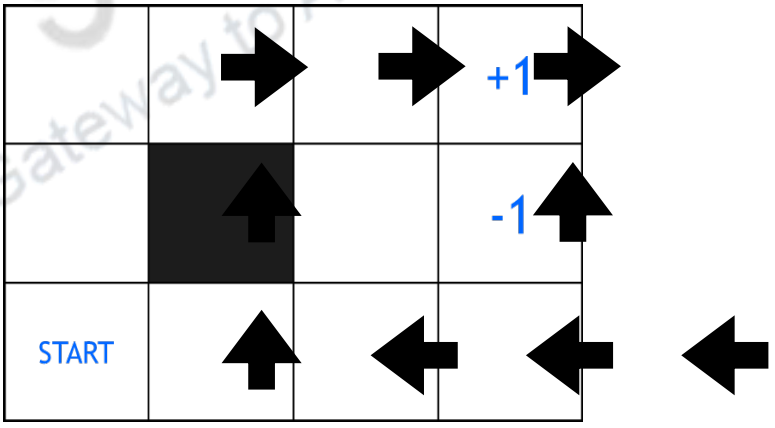


Figure 37.8

Optimal policy solution for the given robot example is shown in figure 37.8. Solution when the reward for each step is -2, -0.1 and -0.04 is given in Figure 37.9. The solutions when the reward for each step is -0.01 and +0.01 is given in Figure 37.10.

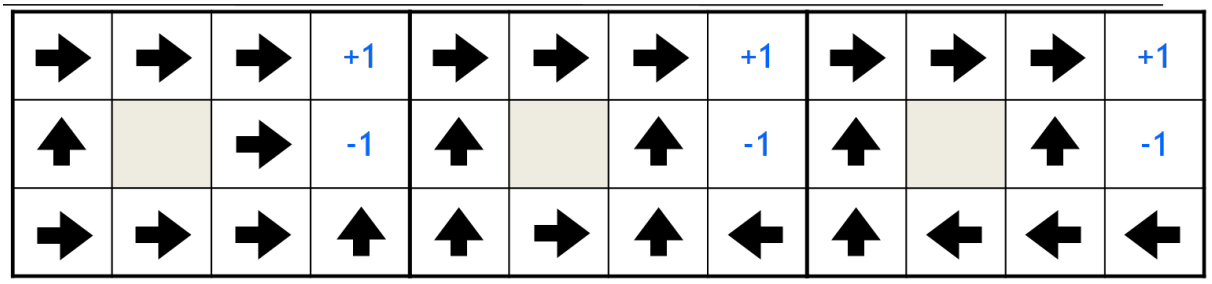


Figure 37.9 Solutions for Reward for each step being -2, -0.1, and -0.04

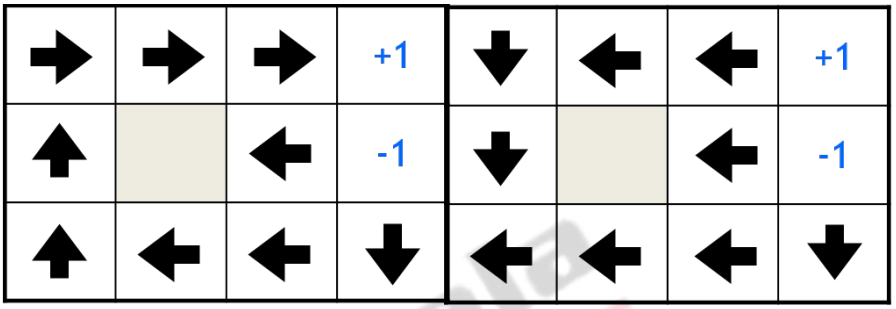


Figure 37.10 Solutions for Reward for each step being -0.01 and +0.01

As we can see from the example, when we change the policy in the environment, the optimal solution changes. It also effects the way the agent takes the action in each state.

Summary

- Outlined the key features of Reinforcement Learning
- Explained the elements & models of Reinforcement Learning.
- Reviewed the MDP model
- Discussed some applications of RL