EE675A - IITK, 2022-23-II

Lecture 1: Introduction

9th January 2023

Lecturer: Subrahmanya Peruru *Scribe: Harsh Kumar, Dhriti Garg*

This is the first lecture in the course *EE675A - Introduction to Reinforcement Learning* offered in the 2022-23-II semester. In this lecture, we will go though the different paradigms in machine learning. Then we will shift our focus to Reinforcement Learning and see the terminologies related to it. In the end, we shall lay a conceptual foundation of the "Bandit Problem", the most basic case of Reinforcement Learning.

1 Types of Machine Learning

1.1 Supervised Learning

In this type of Machine Learning, we deal with labeled data. This is usually seen as a curve-fitting paradigm. We are given multiple data points in an n-dimensional space, and we try to fit a curve that best predicts the labels associated with the maximum number of points. Generally, supervised learning is of two types -

- **Regression** - In a regression problem, the data points are provided with real-valued labels, i.e., a real value as an expected output for each data point. The objective of a supervised learning algorithm is to design a model and learn using the input data set; to take in a test n-dimensional input, and try to predict a real value as close to the expected value as possible. E.g., In the following table, M1 and M2 represent mid-term and end-term marks (out of 100 each), respectively, for some students, and the total score is calculated using an unknown weightage to each of the scores.

M1	M2	Score
30	53	46.1
62	71	68.3
92	83	85.7
96	84	87.6

  In this supervised regression setting, the algorithm's objective is to fit the given data points on a plane (can be any other curve too). The plane here would be $y = 0.3 * M_1 + 0.7 * M_2$

- **Classification** - In a classification problem, the data points are provided with a usually discrete label representing a distinct class of data points. The objective of a supervised learning algorithm is to design a model and learn using the input data set; to take in a test n-dimensional input and try to predict the discrete value associated with the right class.

E.g., In the following table, M1 and M2 represent mid-term and end-term marks (out of 100 each), respectively, for some students, and the result (Distinction, Pass or Fail) is calculated using an unknown weightage to each of the scores.

M1	M2	Result
30	53	Fail
62	71	Pass
92	83	Distinction
96	84	Distinction

In this supervised classification setting, the algorithm's objective is to model the given data points in such a way that the result of the students can be predicted. One "possible" model can be -

$$\text{Result} = \begin{cases} \text{Distinction} & ; \text{if } 0.3M_1 + 0.7M_2 \geq 80 \\ \text{Pass} & ; \text{if } 50 \leq 0.3M_1 + 0.7M_2 < 80 \\ \text{Fail} & ; \text{if } 0.3M_1 + 0.7M_2 < 50 \end{cases}$$

1.2 Un-supervised Learning

In this type of Machine Learning, we deal with unlabeled data. This is usually seen as a pattern-finding paradigm. We are given multiple data points in an n-dimensional space, and we try to find patterns in the data set. The objective of the unsupervised learning algorithm is to take the n-dimensional data points provided and create a model of the data such that patterns can be visualized in the data.
Examples of such unsupervised algorithms are clustering (like K-means), Hierarchical Clustering, and Principal Component Analysis.

1.3 Reinforcement Learning

In this type of Machine Learning, we deal with situations where we want our algorithm to learn and map the situation to actions, so as to maximize a numerical reward.[2] This usually seems like a learning via trial-and-error paradigm, but we will see how helpful it is in several learning situations.

2 Terminologies[2]

Below is a list of the basic terminologies required to define any Reinforcement Learning problem -

- **Agent** - An entity that can perceive and explore the environment and act upon it.
- **Environment** - An environment is a dynamic setting in which an agent is present or surrounded by. It changes state with respect to time depending upon the action of the agent.

- **State (S_t)** - State is a situation returned by the environment after each action taken by the agent.
- **Action (A_t)** - Actions are the moves taken by an agent within the environment.
- **Reward (R_t)** - A feedback returned to the agent from the environment to evaluate the action of the agent.
- **Terminal State (S_T)** - Terminal State is an optionally defined state. The game stops if this state is encountered.
- **Policy** - A policy defines the learning agents way of behaving at a given time. It is a mapping from perceived states of the environment to actions to be taken when in those states.

$$\pi(S_t) \rightarrow A_t$$

- **Return (G_t)** - Return is defined as the cumulative reward $\sum_{i \geq t}^{T-1} R_{i+1}$

3 Immediate RL setting (k-armed Bandit Problem)[1]

The first problem we shall explore is a simple version of the k-armed bandit problem.

The environment consists of K arms or levers labelled $a_1, a_2, ..., a_k$. Think of the lottery levers of a slot machine; each action selection is like a play of one of the slot machines levers, and the rewards are the payoffs for hitting the jackpot. The agents goal is to maximize its total reward over T rounds. We list some important assumptions -

- Every lever has an underlying stationary probability distribution from which the numerical reward is chosen at any time t, depending on the arm selected.
- The reward for each action is independent and identically distributed ($i.i.d$). Every time an action is chosen, the reward is sampled independently from the distribution of the corresponding lever. The reward distributions are initially unknown to the agent.
- We restrict the value of the rewards to $[0, 1]$ for simplicity.

Let us denote the mean of each arm a by $\mu(a)$. The optimal arm is the arm a^* with $\mu(a^*) = \mu^* = max_{a \in A} \mu(a)$, where A is the set of arms. The difference $\mu^* - \mu(a)$ denotes the suboptimality of any arm a as compared to μ^*.

3.1 Objectives

Let's discuss different objectives to be fulfilled by the agent -

1. **Best arm Identification** - In this setting, the agent chooses the arm with the best mean reward. Hence, all available T steps can be used to find the best arm, by playing each arm $\frac{T}{K}$ times and choosing the arm with the best sample mean reward.
2. **Regret Minimization** - This approach quantifies the 'regret' of not knowing the best arm in advance. We compare the cumulative reward with $\mu^* \cdot T$ - the expected reward if the optimal arm is played for all time instances T. Formally, we define regret as follows:

$$R(T) = \mu^* \cdot T - \sum_{t=1}^{T} R_t$$

$$\mathbb{E}[R(T)] = \mu^* \cdot T - \sum_{t=1}^{T} \mu(a_t)$$

where a_t is the arm chosen at step t and $\mu(a_t)$ is the mean reward of the corresponding arm, and R_t is the actual reward sampled from the reward distribution of that arm at step t.

3.2 Approaches

We now look at some algorithms an agent can adopt to achieve these objectives, specifically "Regret Minimization" -

- **Explore, then Commit (ETC)** - We start with a simple approach:
    1. We start with some $x\%$ of the T steps for **exploration**, i.e., we use these $x\%$ of the T steps to find the best arm. This can be done by trying every arm $\frac{x}{K}\%$ of the T times and recording the rewards obtained.
    2. These $\frac{xT}{100K}$ samples for each arm can be used to get a sample mean reward for each arm. These sample means will serve as an approximation for the actual mean reward of each arm. Straight-forwardly, the arm with the best sample mean reward can be chosen as the best arm.
    3. Then, we use the remaining $(100 - x)\%$ of the T steps to **exploit** the best arm chosen (by pulling the lever corresponding to the chosen arm continuously).
- ϵ**-greedy** - The ϵ-greedy algorithm tries to spread exploration more uniformly over time. We specify a small probability ϵ. Now it works as follows -
    1. With probability ϵ, pick a random arm
    2. With probability $1 - \epsilon$, pick the arm with the best sample average.

We shall study the analysis of these algorithms in the next lecture.

References

[1] A. Slivkins. *Introduction to Multi-Armed Bandits*, volume 7. 2022.

[2] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.