IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol-6, Special Issue-4, May 2018 E-ISSN: 2347-2693

# Optimally Facing the uncertainty : A brief survey on Reinforcement Learning

**R. Raja Rajeswari[1]*, A. Pethalakshmi[2]**

[1*,2] Department of Computer Science, Mother Teresa Women's University, Kodaikanal

*Corresponding Author: raj.saravanan2002@gmail.com

**Available online at: www.ijcseonline.org**

***Abstract*—** Reinforcement Learning is a combination of supervised learning and unsupervised learning, the two main streams of Machine Learning .It has many applications in Artificial Intelligence arena. Multi Armed Bandits problem, a classical Reinforcement Learning task employs exploration and exploitation tradeoff. Efficient Bandit Algorithms for solving Bandit problem proides solutions for various problems from Dynamic pricing to online multi class prediction. This research article analyses the elements of Reinforcement learning, mathematical formulation of multi armed Bandits problem and attempts to present a naive RL algorithm for N-Queens problem for an instance of N=4 and concludes with applications of Reinforcement Learning .

***Keywords*—** Reinforcement Learning, multiarmed Bandit problem, ϵ- Greedy algorithm

## I. INTRODUCTION

### A. An Overview

Reinforcement Learning, a subdomain of Machine Learning, deals with approaching uncertain environments with a balance between exploitation and exploration. It finds applications in various significant domains like game playing and Robotics. This research paper discusses basics of Reinforcement Learning and takes a peep into algorithms for multi Arm Bandits problem, a classical Reinforcement Learning task. A naïve attempt on solving N queens problem for an instance of N=4, using Reinforcement Learning approach also has been proposed. The article concludes with application area of Reinforcement Learning and its future directions.

### B. History

Reinforcement Learning [7] existence dates back to 1950s in the name of dynamic programming. Through the years, it has evolved into current scenario and the nomenclature may have derived from the term "Secondary reinforcers" in animal learning Psycology. A secondary reinforcer is a stimulus that has been paired with a primary reinforcer such as food or pain. Apart from temporal difference learning, learning automata, and Q learning and actor critic architecture, the distinct components of Reinforcement Learning, this domain has seen a leaping growth in recent years. The following section peeps into basics of Reinforcement Learning

## 2. REINFORCEMENT LEARNING: AN INTRODUCTION

### A. Elements of reinforcement learning

Reinforcement Learning [7, 1, 8] differs from supervised Learning and unsupervised Learning, the two significant tasks of Machine Learning, in the way of interactions with an environment. In Reinforcement Learning, a learning agent interacts with its environment to achieve its goal. In the process rewards or penalties await. The agent has to decide whether to exploit the previous knowledge on environment or explore new paths, at each state of transition. The balancing act leads in attaining the goal.

Apart from the agent and the environment there are four main sub elements of a reinforcement learning system:

(i) A policy

A policy defines the learning agent's way of behavior at a given time

(ii) A reward function

A reward function defines the goal in a reinforcement learning problem.

(iii) A value function

A value function specifies the cumulative reward after n iterations and

(iv) A model of the environment (optional)

 

A model of the environment simulates the behavior of the environment.

The most important feature distinguishing reinforcement learning from other types of learning is that it uses training information that evaluates the actions taken rather than instructs by giving correct actions.

*B. The agent – environment interface*

The Agent-Environment interface forms the base of Reinforcement Learning. Consider an agent who interacts with a dynamic environment.

An agent usually has only partial knowledge of its environment and therefore will use some form of learning scheme based on the observed signals. Based on the state and action chosen an immediate reward is seen.

More specifically [7], the agent and environment interact at each of a sequence of discrete time steps = 0,1,2,3 ……. At each time step t, the agent receives some representation of the environment's state, st $\epsilon$ S, where S in the set of possible states, and on that basis selects an action at at $\epsilon$ A (st), where A(st) is the set of actions available in state st. One time step later, in part as a sequence of its action, the agent receives of numerical reward, rtt1 $\epsilon$ R and finds itself in a new state st+1.

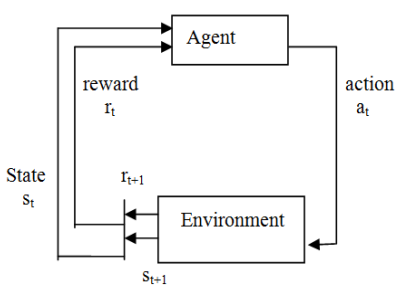


Fig.1 The Agent – Environment Interface

At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted $\pi_t$, where $\pi_t$ (s, a) is the probability that $a_t$ =a if $s_t$=s. Reinforcement Learning methods explore how the agent changes its policy as a result of its experience. The agent's goal is to maximize the total amount of reward it receives over the long run.

C. Markov Decision Processes

Reinforcement learning task can be mathematically formalized as a Markov Decision process. Markov Decision Processes [10] include.

1. A finite set of states
2. A set of actions available in each state
3. Transitions between states
4. Rewards associated with each transition.
5. A discount factor $\gamma$ between 0 and 1, a quantity that formulates the importance of immediate and future rewards. and
6. Future is independent of the past, given the present. The goal is to maximize sum of rewards in the long term

$$\sum_{t=0}^{\alpha} \gamma^t r_t (s_t, a_t).$$

Having seen the basics, the next section unveils a classical reinforcement Learning Task, Multi Armed Bandits problem.

## II. MULTI ARMED BANDIT PROBLEM

*A. The Problem*

Multi armed Bandit problems [6] have been an active area of research since the 1950s. The multi armed bandit problem can be described as the problem that gambler faces at an array of slot machines, when deciding which machine to play at each time instant. Each machine provides a reward from distribution specific to that machine. The goal of the gambler is to maximize his total expected reward. Mathematically a simple multi –armed bandit problem can be stated as follows: Let N denote the set of n bandit arms N = {1,2, …..n}

Each bandit arm i $\epsilon$ N gives a reward by a random variable Xi. This random variable Xi has mean $\mu_i$. The distribution followed by any arm i as well as the mean $\mu_i$ is unknown. Let i* be such that

$$\mu_i^* = \text{Max } \mu_i \quad i \epsilon N$$

The objective is to maximize the total expected reward in a finite Horizon T.

*B. Algorithm*

Generally, Bandit Algorithms [2], have two objectives (i) exploit the optimal arm to gain more pay off and (b) explore the non optimal arms to increase the knowledge. These two objectives together contribute to the final goal : maximize the cumulative reward in a long term. Most widely used algorithm include $\epsilon$- greedy, Softmax and UCB1. This research paper discusses briefly $\epsilon$-greedy algorithm [6,9]. $\epsilon$-greedy is one of the simplest algorithms for the MABP and a basic version follows.

Initialization :

t =1 ;

for t ≤ T do

begin

Pick $\epsilon_t$ Σ (0,1)

 

(a constant) (eg. 0.1)

j = arm with max reward at time t

Play $j^{th}$ arm with probability $1 - \epsilon_t$

and with $\epsilon_t$ play a randon arm;

t= t+1;

end

For a simple, 3 armed bandit problem in which odd numbered arm gives a reward of 10 for an odd iteration else -10 and an even numbered arm gives a reward of 10 for an even iteration else – 10. Let $\epsilon$= 0.1 and $\epsilon$ = 0.4 be two initial instances. A finite experiment for 10 and 20 iterations makes an inference that as $\epsilon$ increases, rewards get maximized.

The following section brings an attempt on solving N - Queens problem with reinforcement learning.

## IV. N. QUEENS PROBLEM

### A. *The Problem*

N Queens Problem [3] is a classic combinatorial problem which tries to place N queens on a N X N chess Board so that no two " attack", that is so that no two of them are on the same row, column, or diagonal. Let us number the rows and column of the chess board, 1 through N. The queen may also be numbered 1 through N. Since each queen must be on a different row, we can without loss of generality assume queen i is to be placed on row i. All solutions to the N Queens problem can therefore be represented as N tuples (X1, X2, …. XN) where Xi is the column on which i is placed.

### B. *Proposed Algorithm*

The following algorithm attempts to solve N Queen problem for the instance N =4, armed with reinforcement learning.
For i = 1 to 100

begin

reward (i) = 0;

for j ← 1 to 4

X [j] = 0;

for j ← 1 to 4

begin

do

X [ j ] = (rnd (i) ) * 100 mod 5

while (X [j] ! = 0)

end

for j ← 1 to 3

begin

if X (j + 1) = X (j) + 1 or

X (j) = X (j + 1) + 1

or X (j) = X (j +1)

penalty (i) + = j

else

reward (i) + = j

if penalty(i) > 0 break;

else

If reward (i) == 10

for K ← 1 to 4

print X (k)

end

Fig. 2 RL Algorithm

If a generated individual satisfies the fitness condition for a solution, cumulative reward is calculated else penalty is computed. Once penalty occurs, break is recommended else iteration continues. The execution of this algorithm will result in two solutions for 4 – Queens problem with existing or increased number of iterations. It may be seen that the algorithm pays a trade off between exploration and exploitation, advocating exploration for a maximum. Expected solutions are as follows.

X = 2413 X = 3142

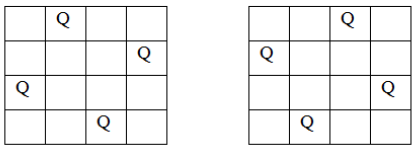


Fig.3 Solutions

The same algorithm may be extended for enhanced values of N, with limited modifications.

## V. CONCLUSION

Within the scope of this research paper, only basics of Reinforcement Learning has been dealt .Reinforcemnnt Learning approaches may empower, classical algorithms to NP – complete problem like N Queen problem.

Recently, Reinforcement Learning with the advent of Deep Learning, has seen frontiers like Deep Reinforcement learning and Deep Q learning emerge, which finds applications in Game playing. Deep Learning tools like

 

Tensor flow advocate further growth of this domain. Banditron, a perceptron based Bandit algorithm has a role in online multiclass prediction [4] and applications are with Internet advertising, Network server selection and Robotics domains. It may be concluded that Reinforcement Learning has a definite role in artificial Intelligence applications and ample opportunities await.

## REFERENCES


[1] Alpaydm, Introduction to Machine Learning, MIT Press, 2010.

[2] Dai Shi, Exploring Bandit algorithms for Automatic content selection, Barcelona, 2014.

[3] Ellis Horrowitz, Fundamentals of Computer Algorithms, Galgottia Publictions,1996

[4] Kakade et al, Efficient Bandit Algorithms for online multiclass prediction, Conference Proceedings, University of Pennysylvnia,2008.

[5] Nahum Shimkin, Reinforcement Learning –Basic algorithms, Learning in Complex Systems Lecture notes, Spring 2011.

[6] Prathamesh, Multi armed bandit approach for Dynamic pricing, M.Tech Dissertation, Iit, Mumbai, 2015

[7] Sutton and Barto, Reinforcement Learning – an introduction, The MIT Press, England, 2017.

[8] Szepesvari, Algorithms for Reinforcement Learning, Synthesis lectures on Artificial Intelligence and machine learning, 2009.

[9] http://cs.mcgill.ca/~dprecup/courses /AI/Lectures/ai-lecture13.pdf

[10] www.medium.com/machinelearning for humans .html


 