**Nuts and Bolts of Reinforcement Learning: Introduction to Temporal Difference (TD) Learning**

[**https://www.analyticsvidhya.com/blog/2019/03/reinforcement-learning-temporal-difference-learning/**](https://www.analyticsvidhya.com/blog/2019/03/reinforcement-learning-temporal-difference-learning/)

**Reinforcement Learning: Introduction to Monte Carlo Learning using the OpenAI Gym Toolkit**

[**https://www.analyticsvidhya.com/blog/2018/11/reinforcement-learning-introduction-monte-carlo-learning-openai-gym/?utm\_source=blog&utm\_medium=reinforcement-learning-temporal-difference**](https://www.analyticsvidhya.com/blog/2018/11/reinforcement-learning-introduction-monte-carlo-learning-openai-gym/?utm_source=blog&utm_medium=reinforcement-learning-temporal-difference)

## Table of Contents

* Monte Carlo Learning
* Introduction to Temporal Difference (TD) Learning
  + TD Prediction
  + TD Control
* Off-Policy vs On-Policy Learning
* Q-learning: Off-Policy Temporal Difference Learning
* SARSA: On-Policy Temporal Difference Learning
* Case Study: Taxi Scheduling using Q-learning in Python

**Monte Carlo Learning:**

1. **Monte Carlo learning is in which we do not have any information about the environment and everything is to be learned by experience.**

**Important links to understand the subject better:**

1. [*Simple Beginner’s guide to Reinforcement Learning & its implementation*](https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/)
2. [*Nuts & Bolts of Reinforcement Learning: Model Based Planning using Dynamic Programming*](https://www.analyticsvidhya.com/blog/2018/09/reinforcement-learning-model-based-planning-dynamic-programming/)
3. [*Reinforcement Learning Guide: Solving the Multi-Armed Bandit Problem from Scratch in Python*](https://www.analyticsvidhya.com/blog/2018/09/reinforcement-multi-armed-bandit-scratch-python/)

## Table of Contents in Monte Carlo Learning

1. Model-Based vs Model-Free Learning
2. Monte Carlo Methods – An Example
3. Monte Carlo Reinforcement Learning
   1. Monte Carlo Prediction
   2. Monte Carlo Control
4. Implementation in Python using OpenAI Gym

**Model-Based vs Model-Free Learning**

Dynamic programming is used to solve problems where the environment is known beforehand.

Model: The transition probabilities from one state to another (or the so called model of the environment) are not known beforehand in most of the real life situations.

**Monte Carlo Methods – An Example**

**Any method which solves a problem by generating suitable random numbers, and observing that fraction of numbers obeying some property or properties, can be classified as a Monte Carlo method.**

Monte Carlo approximation to find the Pi value

Area of a Circle = Pi \* r ^ 2

Area of a square with this Circle is inscribed will have an area of 4 \* r ^ 2

Area of Circle/Area of Square = Pi/4

So, Pi = 4\* (Area of Circle/Area of Square)

Here comes approximation to randomly ask a computer to select points in this circle.

Pi turns out to be this.

Pi = 4 \* (Total number of points that fall in the circle / Total number of points that fall in the square)

Eg; If computer selects 3000 points, Pi = 4 \*(Points in circle/Points in Square)

**Monte Carlo Reinforcement Learning**

Refer this link

<https://www.analyticsvidhya.com/blog/2018/11/reinforcement-learning-introduction-monte-carlo-learning-openai-gym/?utm_source=blog&utm_medium=reinforcement-learning-temporal-difference>

Introduction to Temporal Difference Learning

We can use Monte Carlo Learning to solve Markov Decision Process when the model dynamics of the environment are not known in advance.

But Monte Carlo Learning is only applicable for episodic tasks.

Examples of episodic tasks (which lasts a finite amount of time) are:

* 1. Playing a single game of chess.

Temporal Difference is a model-free learning algorithm and it could be applied to learn the non-episodic tasks as well.



**Off-Policy vs On-Policy learning**

* Off-policy learning algorithm: Q-learning
* On-policy learning algorithm: SARSA

What’s the difference between off-policy and on-policy learning? The answer lies in their names:

* **Off-policy learning:** The agent learns about policy π from experience sampled from another policy µ
* **On-policy learning:** The agent learns about policy π from experience sampled from the same policy π

Let me break this down in the form of an example. Let’s say you joined a new firm as a data scientist. In this scenario, you can equate on-policy learning as learning on the job. You’ll be trying different things and learning only from your own experience.

Off-policy learning would be where you have full access to the actions of another employee. All you do in this scenario is learn from that employee’s experience and not repeat something that the employee has failed at.

Simply to say, off-policy is like doing operations work and on-policy is like doing the development.