

Reinforcement Learning

Notes

Jingye Wang

✉ wangjy5@shanghaitech.edu.cn

Spring 2020

Contents

1	Introduction	3
2	Review of Basic Probability	5
2.1	Interpretation of Probability	5
2.2	Transformations	5
2.3	Limit Theorem	5
2.4	Sampling & Monte Carlo Methods	6
2.5	Basic Inequalities	8
2.6	Concentration Inequalities	10
2.7	Conditional Expectation	12
3	Bandit Algorithms	14
3.1	Bandit Models	14
3.2	Stochastic Bandits	14
3.3	Greedy Algorithms	15
3.4	UCB Algorithms	16
3.5	Bayesian Bandits and Thompson Sampling Algorithms	17
3.6	Gradient Bandit Algorithms	17
4	Markov Chains	18
4.1	Markov Model	18
4.2	Basic Computations	18
4.3	Classification of States	19

CONTENTS	2
4.4 Stationary Distribution	19
4.5 Reversibility	20
4.6 Markov chain Monte Carlo	20
5 Markov Decision Process	22
5.1 Markov Process	23
5.2 Markov Reward Process	23
5.3 Markov Decision Process	24
5.4 Dynamic Programming	26
6 Model-free Prediction	28
6.1 Monte-Carlo Policy Evaluation	28
6.2 Temporal-Difference Learning	29
7 Model-free Control	30
7.1 On Policy Monte-Carlo Control	31
7.2 On Policy Temporal-Difference Control	33
7.3 Off-Policy Q-Learning Control	33
7.4 Off-Policy Importance Sampling Control	34
8 Value Function Approximation	35
8.1 Introduction on Function Approximation	35
8.2 Incremental Method	35
8.3 Batch Methods	38
8.4 Deep Q-Learning	39
9 Policy Optimization	39
9.1 Policy Optimization	40
9.2 Monte-Carlo Policy Gradient	41
9.3 Actor-Critic Policy Gradient	44
9.4 Extension of Policy Gradient	45

1 Introduction

Course Prerequisite:

- Linear Algebra
- Probability
- Machine Learning relevant course (data mining, pattern recognition, *etc*)
- PyTorch, Python

What is Reinforcement Learning and why we care:

a computational approach to learning whereby *an agent* tries to *maximize* the total amount of *reward* it receives while interacting with a complex and uncertain *environment*. [4]

Difference between Reinforcement Learning and Supervised Learning:

- Sequential data as input (*not i.i.d*);
- The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them;
- *Trial-and-error* exploration (balance between exploration and exploitation);
- There is *no supervisor*, only a reward signal, which is also *delayed*

Big deal: Able to Achieve Superhuman Performance

- Upper bound for Supervised Learning is human-performance.
- Upper bound for Reinforcement Learning ?

Why Reinforcement Learning works now?

- Computation power: many GPUs to do trial-and-error rollout;
- Acquire the high degree of proficiency in domains governed by simple, known rules;
- End-to-end training, features and policy are jointly optimized toward the end goal.

Sequential Decision Making:

- Agent and Environment: the agent learns to interact with the environment;
- Rewards: a scalar feedback signal that indicates how well agent is doing;
- Policy: a map function from state/observation to action models the agent's behavior;
- Value function: expected discounted sum of future rewards under a particular policy;
- Objective of the agent: selects a series of actions to maximize total future rewards;
- History: a sequence of observations, actions, rewards;
- Full observability: agent directly observes the environment state, formally as Markov decision process (MDP);

- Partial observability: agent indirectly observes the environment, formally as partially observable Markov decision process (POMDP)

All goals of the agent can be described by the maximization of expected cumulative reward.

Types of Reinforcement Learning Agents based on What the Agent Learns

- Value-based agent:
 - Explicit: Value function;
 - Implicit: Policy (can derive a policy from value function);
- Policy-based agent:
 - Explicit: policy;
 - No value function;
- Actor-Critic agent:
 - Explicit: policy and value function.

Types of Reinforcement Learning Agents on if there is model

- Model-based:
 - Explicit: model;
 - May or may not have policy and/or value function;
- Model-free:
 - Explicit: value function and/or policy function;
 - No model.