An Introduction to Reinforcement Learning

Adapted from David Silver's Lecture 1 Notes

Jane Lee

University of Pennsylvania

janehlee@seas.upenn.edu

Overview

Introduction

Characteristics of RL

What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a *reward* signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agents actions affect the subsequent data it receives

Examples of RL

- Fly stunt manoeuvres in a helicopter
- Defeat the world champion at Backgammon
- Manage an investment portfolio
- Control a power station
- Make a humanoid robot walk
- Play many different Atari games better than humans

Examples of RL

The RL Problem

Reward

- A reward R_t is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximize cumulative reward

Reinforcement learning is based on the **reward hypothesis**.

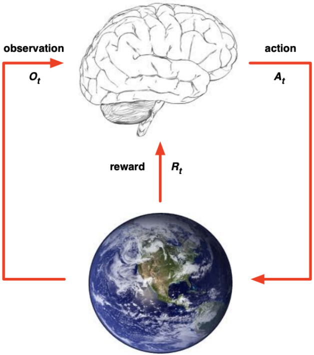
Definition (Reward Hypothesis)

All goals can be described by the maximisation of expected cumulative reward

Sequential Decision Making

- Goal: select actions to maximise total future reward
- Actions may have long term consequences
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward

Agent and Environment



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

Definitions

State

Definition (History)

The **history** is the sequence of observations, actions, rewards

$$H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t.$$

In other words, all observable variables up to time t.

Definition (State)

State is the information used to determine what happens next. Formally, it is a function of the history

$$S_t = f(H_t).$$

State

Definition (Environment State)

The environment state S_t^e is the environment's private representation

- i.e. whatever data the environment uses to pick the next observation/reward.
- The environment state is not usually visible to the agent. Even if S_t^e is visible, it may contain irrelevant information.

Definition (Agent State)

The agent state S_t^a is the agent's internal representation.

- i.e. whatever information the agent uses to pick the next action
- i.e. it is the information used by reinforcement learning algorithms
- It can be any function of history: $S_t^a = f(H_t)$

State

Definition (Information State)

An information state (a.k.a. Markov state) contains all useful information from the history.

Definition (Markov State)

A state S_t is Markov if and only if

$$P[S_{t+1} \mid S_t] = P[S_{t+1}|S_1, ..., S_t].$$

In other words, "The future is independent of the past given the present," or once the state is known, the history may be thrown away.

(The environment state and the whole history are both Markov)

Environment

Definition (Full Observability)

Agent directly observes environment state

$$O_t = S_t^a = S_t^e.$$

- Agent state = environment state = information state
- Formally, this is a **Markov decision process (MDP)**. (Next)

Environment

Definition (Partial Observability)

Agent indirectly observes environment. (ex: a robot with camera vision isn't told its absolute location, a poker playing agent only observes public cards)

- Now, agent state $\neq$ environment state.
- Formally, this is a **partially observable Markov decision process (POMDP)**.
- Agent must construct its own state representation S_t^a:
  - Complete history: $S_t^a = H_t$
  - **Beliefs** of environment state: $S_t^a = (P[S_t^e = s^1], \ldots, P[S_t^e = s^n])$
  - Recurrent neural network: $S_t^a = \sigma\left(S_{t-1}^a W_s + O_t W_o\right)$

RL Agent

An RL agent may include one or more of these components:

- Policy: agent's behaviour function
- Value function: how good is each state and/or action
- Model: agent's representation of the environment

Policy

A policy is the agent's behaviour. It is a map from state to action, e.g.

- Deterministic policy: $\pi(s) = a$
- Stochastic policy: $\pi(a \mid s) = P[A_t = a | S_t = s]$

Value

Value function is a prediction of future reward. Used to evaluate the goodness/badness of states and therefore to select between actions, e.g

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+2} \cdots \mid S_t = s]$$

Model

A model predicts what the environment will do next

- Transitions: $\mathcal{P}$ predicts the next state
- Rewards: $\mathcal{R}$ predicts the next (immediate) reward

$$\mathcal{P}_{ss'}^{a} = P[S_{t+1} = s' \mid S_t = s, A_t = a]$$

$$\mathcal{R}_{s}^{a} = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

Categorizing RL Agents

- Value Based
  - No Policy (Implicit)
  - Value Function
- Policy Based
  - Policy
  - No Value Function
- Actor Critic
  - Policy
  - Value Function

- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model

Relation to Contextual Bandits

Review Contextual Bandits

Re-introducing the contextual bandit problem using notation borrowed from RL.

for $t = 1$ to T:

Learner sees context $S_t \in \mathcal{S}$
Learner selects action $A_t \in \mathcal{A}$ (with consideration to S_t)
Learner receives reward $R_t = R_t^{A_t}$

end for

The optimal policy is one which maximizes the value function for every context $s \in \mathcal{S} : \pi^*(s) = \arg\max_a \mathbb{E}[R_t^a \mid S_t = s]$.

Multi-armed bandit problems are thought of as reinforcement learning problems with single state.

Reinforcement Learning

Now, there are T episodes as before, but each episode can have a series of state-action pairs.

for $t = 1$ to T:

Learner sees $S_{t,0} \in \mathcal{S}$

for $k = 0$ to $K - 1$:

Learner selects action $A_{t,k} \in \mathcal{A}$

Learner sees state $S_{t,k+1}^{A_{t,k}} \in \mathcal{S}$

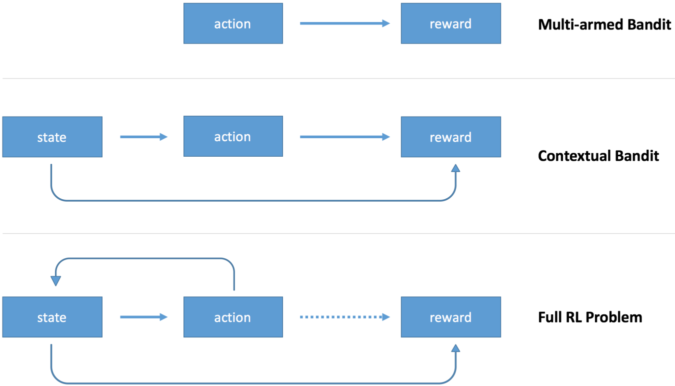
Learner receives reward $R_{t,k} = R(S_{t,k}^{A_{t,k-1}}, A_{t,k}, S_{t,k+1}^{A_{t,k}})$

end for

end for

Policy is now a vector $\pi = (\pi_0, \ldots, \pi_{K-1})$ so that $A_{t,0} = \pi_0(S_{t,0}), A_{t,1}\pi_1(S_{t,1}^{A_{t,0}}), \ldots, A_{t,K-1} = \pi_{K-1}(S_{t,K-1}^{A_{t,K-2}})$.

- Bandit problems are thought of being a special case of reinforcement learning.
- It is harder to get regret bound results for general RL problems.
- Both problems involve maximizing some cumulative reward.
- Both involve aspects of balancing *exploration* and *exploitation*.


action
reward
Multi-armed Bandit
state
action
reward
Contextual Bandit
state
action
reward
Full RL Problem

Markov Decision Process

MDP

Markov decision processes formally describe an environment for reinforcement learning where the environment is fully observable.

- i.e. The current state completely characterizes the process
- Almost all RL problems can be formalised as MDPs
  - Partially observable problems can be converted into MDPs
  - Bandits are MDPs with one state

MDP

Remember the **Markov property**: a state S_t is Markov if and only if $P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t]$.

For a Markov state s and successor state s', the *state transition probability* is defined by

$$\mathcal{P}^a_{ss'} = P[S_{t+1} = s' \mid S_t = s].$$

A state transition matrix defines transition probabilities from all states s to successor state s'. (Let $|\mathcal{S}| = n$.)

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix}$$

Markov Chain Review

Definition (Markov Chain)

A Markov chain (or Markov process) is a tuple $< \mathcal{S}, \mathcal{P} >$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a transition probability matrix.

Definition (State Probability Vector)

A vector $q_t = (q_1^t, \ldots, q_{|\mathcal{S}|}^t)$, where q_s^t means that the Markov chain is in state s at time t. Note $q^{t+1} = q^t \mathcal{P}$.

State probability vector such that $q\mathcal{P} = q$ is called **stationary distribution**.

(Others: irreducible, aperiodic, ergodic, FTMC)

MRP and MDP

Definition (Markov Reward Process)

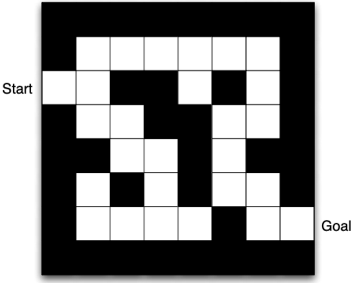
A Markov reward process is a Markov chain with values. It is formally specified by a 4-tuple $< \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma >$.

Definition (Markov Decision Process)

A Markov decision process is a Markov reward process with decisions (actions). It is formally specified by a 5-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$.
(Probabilities and rewards can be action-dependent.)

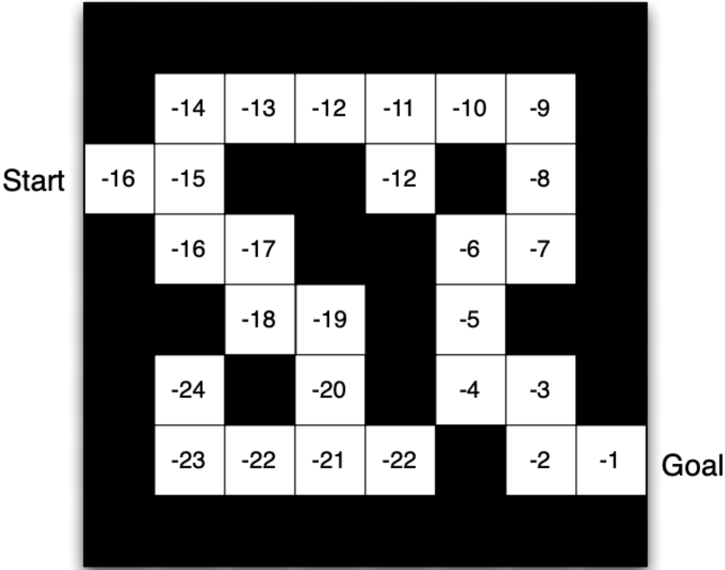
Appendix

Maze Example

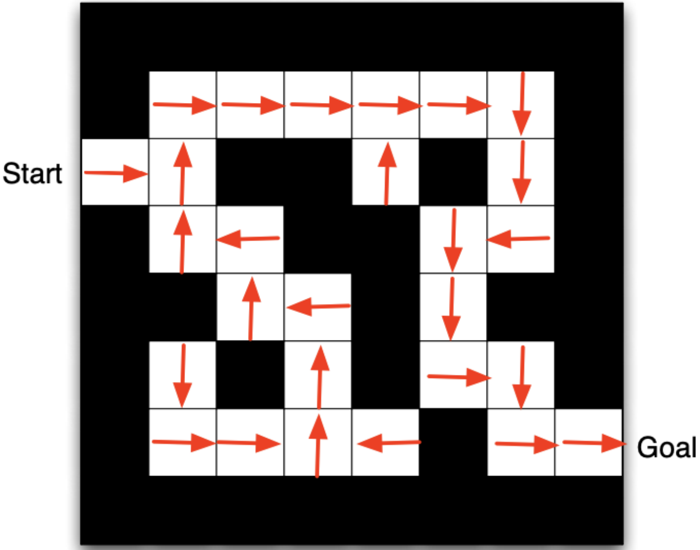


- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agents location

Value Based


Start
-14
-13
-12
-11
-10
-9
-16
-15
-12
-8
-16
-17
-6
-7
-18
-19
-5
-24
-20
-4
-3
-23
-22
-21
-22
-2
-1
Goal

Policy Based



Model Based