Introduction to RL

Reference: FML Chapter 14
Sutton and Barto "Reinforcement learning"

Outline:

1. Intro
2. Model-based and Model-free RL
3. Temporal difference (TD) methods
4. Function approximation for Value function
5. Actor-critic methods
6. (TBD)
Different learning frameworks

Supervised: learning from a training set of labelled examples

Unsupervised: find hidden structure in data, estimate density function

Reinforcement: learns from interaction, not from examples

goal is to max reward, not to find hidden structure

Learning from interaction

1. learn what to do
   learn actions to max a numerical reward

2. The agent is not told what to do, but it must discover the best behavior

3. The actions that it takes affect future outcome

Exploration and exploitation dilemma

In RL a goal-seeking agent must simultaneously
   * exploit current knowledge
   * explore new actions

Abstraction:

RL offers an abstraction to the problem of goal-directed learning from interaction.
Learning from interaction

- Reinforcement learning involve learning what to do
- It maps solutions to actions as to maximize a numerical reward
- The agent is not told what to do but it must discover the best behaviour
- The actions that it takes affect future outcomes
Learning from interaction in practise

- Reinforcement learning in practise gives only an approximation to a true solution
- Real problem might be continuous and high dimensional
Exploration and exploitation dilemma

In reinforcement learning we have a goal-seeking agent that must simultaneously:

- **exploit** current knowledge
- **explore** new actions

The agent must try a variety of actions and progressively favour those that appear to be best.
It proposes that the sensory, memory and control apparatus and the objective can be reduced to states, actions and rewards passing back and forth between the agent and the environment.

The agent–environment interface

\[
\begin{align*}
\text{Environment} \rightarrow \text{Agent} \\
\text{state } s_t \rightarrow \text{reward } r_t \rightarrow \text{action } a_t
\end{align*}
\]

Reward hypothesis:
maximise the expected value of the cumulative reward
RL: abstraction

- **State space** \( S = \{ s_1, \ldots, s_k \} \)
- **Action space** \( A = \{ a_1, \ldots, a_l \} \)
- **Reward space** \( R \)

- **History** \( h_t = \{ s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{t-1}, a_{t-1}, r_t, s_t, a_t \} \)

State contains all information about the environment to the agent.

- **Transition model**:
  \[ P_r(S_{t+1}=s', r_{t+1}=r \mid h_t) = \Pr(S_{t+1}=s', r_{t+1}=r \mid s_t, a_t) \]

- **Policy**:
  \[ \Pr(a_{t+1} \mid h_t, S_{t+1}) \]

Markov property: \( S_{t+1} \) only depends on \( S_t \) and \( a_t \)

\[
p(s', r \mid s, a) = \Pr(S_{t+1}=s', r_{t+1}=r \mid S_t, a_t) = \mathbb{E}[r_{t+1} \mid S_t=s, a_t=a] = \sum_r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}
\]

- **State transition probability** \( p(s' \mid s, a) = \sum_r p(s', r \mid s, a) \)
- **Expected reward** \( r(s, a, s') = \mathbb{E}[r_{t+1} \mid S_t=s, a_t=a, S_{t+1}=s'] \)

\[
r(s, a, s') = \sum_r r p(r \mid s, a, s') = \sum_r \frac{r p(s', r \mid s, a)}{p(s' \mid s, a)}
\]

- **Policy**:
  \[ \Pr(a_{t+1} \mid S_{t+1}) = \Pr(a \mid S_t) = \pi(a \mid s) \]

- **State-value function** for policy \( \pi \)
  \[ V^\pi(s) = \mathbb{E}_\pi \left[ R_t \mid S_t=s \right] \]

- **Action-value function** for policy \( \pi \)
  \[ Q^\pi(s, a) = \mathbb{E}_\pi \left[ R_t \mid S_t=s, a_t=a \right] \]

Return (accumulated future reward)

\[ R_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k+1} \]
Bellman Equation (under Markov Property)

\[ V^\pi(s) = \mathbb{E}_\pi[ R_t \mid S_t = s ] = \mathbb{E}_\pi[ R_{t+1} + r_Rt+1 \mid S_t = s ] \]

\[ = \mathbb{E}_\pi[ R_{t+1} \mid S_t = s ] + r \mathbb{E}_\pi[ R_{t+1} \mid S_t = s ] \]

(A) \hspace{1cm} (B)

\[ \begin{align*}
(A) &= \sum_r \ell p(r \mid s) = \sum_r \sum_{s',a} p(r, a, s' \mid s) \\
&= \sum_r \sum_{s',a} p(s', r \mid s, a) \pi(a \mid s) \\
&= \sum_r \pi(a \mid s) \sum_{s'} p(s', r \mid s, a)
\end{align*} \]

\[ \begin{align*}
(B) &= \mathbb{E}_\pi[ R_{t+1} \mid S_t = s ] \\
&= \sum_{s'} \mathbb{E}_\pi[ R_{t+1} \mid S_{t+1} = s' ] \mathbb{P}_r(S_{t+1} = s' \mid S_t = s) \\
&= \sum_{s'} \mathbb{E}_\pi[ s' ] \mathbb{P}_r(S_{t+1} = s', R_{t+1} = r \mid S_t = s) \\
&= \sum_{s'} \mathbb{E}_\pi[ s' ] \mathbb{P}_r(S_{t+1} = s', R_{t+1} = r \mid S_t = s, a_t = a) \pi(a \mid s)
\end{align*} \]

\[ \Rightarrow V^\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s' \mid r \mid s, a) [r + \gamma V^\pi(s')] \]
Optimal Value function:

\[ V^*(s) = \max_{a} V^*(s) \quad Q^*(s,a) = \max_{a} Q^*(s,a) \]

\[ Q^*(s,a) = \max_{a} \mathbb{E}[R_t \mid s_t = s, a_t = a] \]

\[ = \max_{a} \mathbb{E}[R_t \mid s_t = s, a_t = a] \]

\[ = \max_{a} \mathbb{E}[R_{t+1} \mid s_t = s, a_t = a] + r \mathbb{E}[R_{t+1} \mid s_t = s, a_t = a] \]

Bellman Optimality Equation

\[ V^*(s) = \max_{a} \sum_{s',r} P(s',r \mid s,a) \left( r + V^*(s') \right) \]

\[ Q^*(s,a) = \max_{a} \sum_{s',r} P(s',r \mid s,a) \left( r + \max_{a'} Q^*(s',a') \right) \]

If we know \( P(s',r \mid s,a) \)

A system of \( |S| \cdot |A| \) equations with \( |S| \cdot |A| \) unknowns.

\[ V^*(s') = \max_{a} Q^*(s',a) \]

\[ V^*(s') = \sum_{s} P(s, a \mid s') Q^*(s,a) \]
Optimal value function

The optimal value function for each state gives highest the expected return that can be obtained from that state.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
The optimal Q-function for each state and action gives the highest expected return that can be obtained from that state when that action is taken.
Optimal policy

The optimal policy is the policy associated with the optimal value function or the optimal Q-function.
RL diagram:
1. Know \( P(\text{rt+1} = r', \text{srt+1} = s' | \text{str} = s, \text{act} = a) \)

Dynamic Programming:

i. Policy iteration

\[
\begin{align*}
\text{policy evaluation} & : V_{k+1}(s) = \sum_a \pi(a|s) \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s')) \\
\text{policy improvement} & : \pi(a|s) = \arg \max_a \mathbb{E}(s, a) = \arg \max_a \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s'))
\end{align*}
\]

ii. Value iteration

\[
V_{k+1}(s) = \max_a \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s'))
\]

Dynamic Programming:

i. Policy iteration

\[
\begin{align*}
\text{policy evaluation} & : V_{k+1}(s) = \sum_a \pi(a|s) \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s')) \\
\text{policy improvement} & : \pi(a|s) = \arg \max_a \mathbb{E}(s, a) = \arg \max_a \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s'))
\end{align*}
\]

ii. Value iteration

\[
V_{k+1}(s) = \max_a \sum_{s', r} P(s', r | s, a) (r + \gamma V_k(s'))
\]

Unknown \( P(\text{rt+1} = r', \text{srt+1} = s' | \text{str} = s, \text{act} = a) \)

i. Monte Carlo prediction

simulate \( V_\pi(s) \), \( \forall s \)

Let's say \( n \) trajectories for each \( s \)

\[
\Pr \left[ \frac{1}{n} \sum_{i=1}^{n} V_\pi(s) - V_\pi(s) > \epsilon \right] \leq e^{-\frac{n \epsilon^2}{2}}
\]

Now, if \( V_\pi(s) \in [c_0, T] \) using Hoeffding's Inequality

ii. Now to do planning, we need to estimate \( Q(s, a) \)