How are you today?

Reinforcement Learning

Intuition: learning to play backgammon
- Play game
- Win/Lose
- "Reinforce" decisions that lead to a win
- "Punish" decisions that lead to a loss
- Credit assignment problem
  - Which moves are responsible
  
(Tesauro, Neurogammon)
(AlphaGo)

\[ S: \text{States} \]
\[ A: \text{Actions} \quad A: S \rightarrow S \]
\[ R: \text{Reward} \quad R: S \rightarrow \{-1, 0, 1\} \]

Markov Decision Process

Actions are probabilistic
\[ P(s'|s, a) \quad s \in S \quad a \in A \]

\[ \pi: \text{Policy} \quad \pi: S \rightarrow A \]
Cumulative Reward (Utility)
\[ U_h(L_{s_0}, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} R(s_t) \]

Cumulative Discounted Reward (Utility)
\[ 0 \leq \gamma \leq 1 \]
\[ U_h(L_{s_0}, \ldots, s_t, \ldots) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \]

Expected Utility of a policy
\[ U^\pi(s_0) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \]

Where, \( s_t = q_t(s_{t-1}) \)
\[ q_t = \pi(s_{t-1}) \]

Optimal Policy \( \pi^* \)
\[ \pi^*(s) = \arg\max_{\pi} U^\pi(s) = U(s) \]
\[ = \arg\max_{\pi} \sum_{s' \in S} P(s' \mid s, a) U(s') \] \[ a \in A \]
\[ = R(s) + \gamma \max_{a \in A} \sum_{s' \in S} P(s' \mid s, a) U(s') \]

Bellman Equation

Example

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>start</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ P(\text{going up} \mid \text{up action}) = .6 \]
\[ P(\text{going L} \mid \text{up}) = .1 \]
\[ P(\text{going R} \mid \text{up}) = .1 \]

and so on
Bellman Equation

\[ u_{(1,1)} = .4 + V_{max} \left( \frac{1}{.5} \cdot 0.8 \cdot u_{(1,2)} \right) \]

\[ \cdot + 0.1 \cdot u_{(2,1)} \]

\[ + 0.1 \cdot u_{(1,1)} \]

\[ + .9 \cdot u_{(1,1)} + 0.1 \cdot u_{(2,1)} \]

\[ + .8 \cdot u_{(2,1)} + 0.1 \cdot u_{(1,2)} \]

\[ + 0.1 \cdot u_{(1,1)} \]

Value Iteration

\[ V_0(s) = 0 \] for all states

Repeat

\[ V_{k+1}(s) = R(s) + V_{max} \sum_{a \in A} P(s'|s, a) V_k(s') \]

Until \(<\text{stopping condition}>\)

(Assumes you maintain all states)

At each state \(s\)

\[ V^*(s) = \arg \max_{a \in A} \sum_{s' \in S} P(s'|s, a) V(s') \]

(Assumes you know \(P(s'|s, a)\))