REINFORCEMENT LEARNING:
LEARNING TO LEARN

Sandjai Bhulai
Vrije Universiteit Amsterdam
s.bhulai@vu.nl
AN ERA OF CHANGE

Technologies
Industrial Robots

$500,000 (2008) to $22,000 (2013)
23x in 5 years
AN ERA OF CHANGE

Demonstration of the task via kinesthetic teaching
A MOTIVATING GAME

Rules of the game

- There is a heap with 21 sticks
- Each player is allowed to pick 4 sticks at maximum
- A player is not allowed to repeat the previous move
- A player wins if the opponent cannot play
The optimal strategy is simple:

**zones with multiples of 5 are safe!**

Suppose we are allowed to pick 5 sticks at maximum. What is the optimal strategy now?

No simple solution!
A MOTIVATING GAME

<table>
<thead>
<tr>
<th>$s$</th>
<th>$z = 1$</th>
<th>$z = 2$</th>
<th>$z = 3$</th>
<th>$z = 4$</th>
<th>$z = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
A MOTIVATING GAME

- Recall:
  \( s = \text{number of sticks in the heap}, \)
  \( z = \text{previous move}. \)

- Look at the rewards:
  \( V(0, z) = -1, \)
  \( V(s, z) = 1 \) for all \( s < 0 \) and \( s + z \geq 0. \)

- Dynamic programming:
  \[
  V(s, z) = \max \left( \min_{a \neq z, b \neq a} [V(s - a - b, b)] \right)
  \]
$V(x) = \max_a \left[ r(x, a) + \sum_y p(x, a, y)V(y) \right]$
THE IOWA GAMBLING TASK

\[
V(x) = \max_a \left[ r(x, a) + \sum_y p(x, a, y)V(y) \right]
\]
THE IOWA GAMBLING TASK

Demo
THE IOWA GAMBLING TASK

Actions taken over time by four participants

Actions participant 1

Actions participant 2

Actions participant 3

Actions participant 4
THE IOWA GAMBLING TASK
Stochastic Bandit setting

Environment: distributions \((\nu_1, \ldots, \nu_K)\) of arm rewards.

Protocol: For \(t = 1, 2, \ldots\)

- Learner picks arm \(I_t\)
- Learner observes and receives reward \(X_{I_t,t} \sim \nu_{I_t}\)

Objective: Minimize pseudo-Regret w.r.t. best expert after \(T\) rounds:

\[
\bar{R}_T = T\mu^* - \mathbb{E}_{I_1, \ldots, I_T} \left\{ \sum_{t=1}^{T} X_{I_t,t} \right\}.
\]
THE IOWA GAMBLING TASK

UCB algorithm

Protocol:
For \( t = 1, \ldots, K \):
Initialize: \( T_i(K) = 1, i = 1, 2, \ldots, K \).
For \( t = K + 1, K + 2, \ldots, T \):

- Do:

\[
I_t = \arg \max_{1 \leq i \leq K} \left[ \hat{\mu}_{i,t} + (\psi^*)^{-1} \left( \alpha \log t \frac{1}{T_i(t - 1)} \right) \right]
\]

- Observe reward \( X_{I_t,t} \)
THE IOWA GAMBLING TASK

UCB algorithm for the IGT

Protocol:
For $t = 1, \ldots, 4$:
Initialize: $T_i(4) = 1, i = 1, 2, 3, 4$.
For $t = 5, 6, \ldots, 95$:

- Do:

$$I_t = \arg \max_{1 \leq i \leq K} \left[ \hat{\mu}_{i,T_i(t-1)} + \sqrt{\frac{\alpha \log t}{2T_i(t-1)}} \right]$$

- Observe reward $X_{I_t,t}$
THE IOWA GAMBLING TASK

Q-learning for the IGT

Protocol:
Choose $\epsilon < 1$
Initialize for $a = 1, 2, 3, 4$:
$Q(a) \leftarrow 0$ and $N(a) \leftarrow 0$

- Loop until $t = T = 95$:

$$A = \begin{cases} \arg\max_a Q(a) & \text{w.p. } 1 - \epsilon \\ \sim \mathcal{U}\{1, 4\} & \text{w.p. } \epsilon \end{cases}$$

- Observe reward $R$
- $N(A) \leftarrow N(A) + 1$
- $Q(A) \leftarrow Q(A) + \frac{1}{N(A)}(R - Q(A))$
THE IOWA GAMBLING TASK

success probability $p$

success probability ???
THE IOWA GAMBLING TASK

- Tension between control vs. learning

Adaptive control
THE IOWA GAMBLING TASK

Thompson sampling for the IGT

Protocol:
For each $\theta_i$ with $i \in \{1, 2, 3, 4\}$ set

- $\theta_i \sim \text{Dir}(\alpha_i)$

as prior. Where $\alpha_1 = (1, 1, 1, 1, 1)$, $\alpha_2 = (1, 1)$, $\alpha_3 = (1, 1, 1, 1)$ and $\alpha_4 = (1, 1)$

For each $t = 1, \ldots, 95$ do:

- Draw a sample $\theta_i \sim \text{Dir}(\alpha_i)$ for each $i \in \{1, 2, 3, 4\}$
- Compute $\mathbb{E}_{\theta_i}[X_i]$ for each $i \in \{1, 2, 3, 4\}$
- $a_t = \arg \max_i \mathbb{E}_{\theta_i}[X_i]$
- Observe outcome in deck $i$
- Update parameter $\alpha_i$
THE IOWA GAMBLING TASK

The average proportion of optimal action over 100 simulations

% optimal action vs # Simulation

- TS
- UCB
- Q-learning
Q-learning

\[ Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left( r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \]

- Challenges:
  - The number of states might be very large
  - The state space might be very complex
A Markov decision process is a framework for modeling decision making in which outcomes are partly random and partly under the control of a decision maker. It is a formal mathematical framework for modeling decision making in which outcomes are partly random and partly under the control of a decision maker.

Let $x$ be the current state, $a$ the action, $y$ the next state, $r$ the reward, and $p$ the transition probability. The value function $V(x)$ represents the expected utility of being in state $x$ and choosing action $a$. The Bellman equation for a Markov decision process is given by:

$$V(x) = \max_a \left[ r(x, a) + \sum_y p(x, a, y)V(y) \right]$$
Solution is function approximation

Model: \( Q_\theta(s_t, a_t) \)

Training data: \( \langle s_t, a_t, r_t, s_{t+1} \rangle \)

Loss function: \( \mathcal{L}(\theta) = \| y_t - Q_\theta(s_t, a_t) \|^2_2 \)

where \( y_t = r_t + \gamma Q(s_{t+1}, \pi(s_{t+1})) \)
Pole-balancing
  > move car left/right to keep the pole balanced

State representation
  > position and velocity of car
  > angle and angular velocity of pole

Solution
  > coarse discretization of 4 state variables
  > left, center, right
  > totally non-Markov, but still works
AMAZON DEEP RACER
QUESTIONS