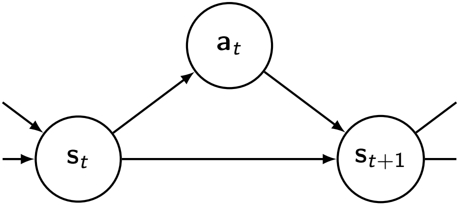
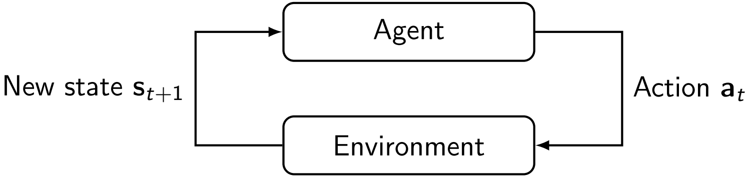
Machine Learning

Reinforcement learning

Maxime Gasse



Agent / environment interaction loop



Action space $\mathbf{a} \in \mathcal{A}$.
State space $\mathbf{s} \in \mathcal{S}$.
Reward $r : \mathcal{S} \to \mathbb{R}$.

Unknown environment.

Agent objective: take actions that maximize long-term reward

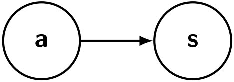
$$\sum_{t=0}^{\infty} r(\mathbf{s}_t).$$

Multi-armed bandit

Problem setup

N slot machines

- $a \in \{1, \ldots, N\}$
- $\mathbf{s}_{t+1} \perp\!\!\!\perp \mathbf{s}_t$ (static system)



Agent action $a \sim p(a)$, environment response $\mathbf{s} \sim p(\mathbf{s}|a)$ (unknown).

Action value $v(a) = \mathbb{E}_{\mathbf{s}|a}[r(\mathbf{s})] = \int_{\mathbf{s}} r(\mathbf{s}) \times p(\mathbf{s}|a) d\mathbf{s}$.

Optimal action $a^\star = \arg\max_a v(a)$.

Optimal action distribution $p^\star(a) = 0 \iff v(a) \neq \max_{a'} v(a')$.

Stochastic policy learning

Consider $p(a|\theta)$ a parametric model.

Model value $v(\theta) = \mathbb{E}_{a|\theta}[v(a)] = \sum_a p(a|\theta) \int_{\mathbf{s}} r(\mathbf{s}) \times p(\mathbf{s}|a) d\mathbf{s}$.

Empirical maximization: $\theta^\star = \arg\max_\theta \sum_a p(a|\theta) \sum_{\mathbf{s} \sim p(\mathbf{s}|a)} r(\mathbf{s})$.

Stochastic optimization: start from arbitrary θ_0 and iterate

- collect samples $a \sim p(a|\theta_i)$ (agent), $\mathbf{s} \sim p(\mathbf{s}|a)$ (environment)
- update θ_{i+1} s.t. $v(\theta)$ increases

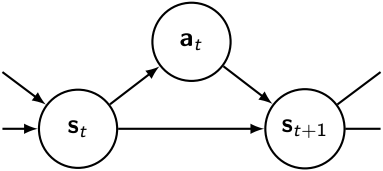
Exploration / exploitation dilemma !
$\Longrightarrow$ Upper Confidence Bound algorithm (UCB)

P. Auer, N. Cesa-Bianchi, and P. Fischer (2002). Finite-time Analysis of the Multiarmed Bandit Problem.

Markov decision process (MDP)

Problem setup

- $\mathbf{s}_{t+1} \not\perp\!\!\!\perp \mathbf{s}_t, \mathbf{a}_t$ (dynamic system)
- $\mathbf{s}_{t+1} \perp\!\!\!\perp \mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{s}_{t-2}, \mathbf{a}_{t-2}, \cdots \mid \mathbf{s}_t$ (Markov property)
- $t \in \{0, \ldots, N\}$ (finite process)



Agent action $\mathbf{a}_t \sim p(\mathbf{a}_t|\mathbf{s}_t)$, environment response $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.

Action value $v(\mathbf{a}_t|\mathbf{s}_t) = \mathbb{E}_{\mathbf{s}_{t+1},\ldots,\mathbf{s}_N|\mathbf{s}_t,\mathbf{a}_t}\left[\sum_{t'=t+1}^{N} r(\mathbf{s}_{t'})\right]$

$$\int_{\mathbf{s}_{t+1},\mathbf{a}_{t+1},\cdots} \sum_{t'=t+1}^{N} r(\mathbf{s}_{t'}) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \prod_{t'=t+1}^{N-1} p(\mathbf{a}_{t'}|\mathbf{s}_{t'}) p(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) d\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \ldots$$

Combinatorial problem!

Imitation learning

Imitate an expert

- collect $\mathcal{D} = \{(\mathbf{s}_t, \mathbf{a}_t)^{(i)}\}$ from expert agents
- estimate $p(\mathbf{a}_t|\mathbf{s}_t)$ from $\mathcal{D}$

Pros:

+ standard supervised learning
+ combinatorial issue vanishes
+ no exploration / exploitation trade-off

Cons:

– expert data can be expensive
– will never perform better than expert...

AlphaGo-expert: imitation learning from professional human games
AlphaGo: imitation learning within MCTS (Monte-Carlo Tree Search)

Value function learning (Q-learning)

Let $Q : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ be our value function $v(\mathbf{a}_t|\mathbf{s}_t)$.

Recursive update rule:

$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(\mathbf{s}_t, \mathbf{a}_t)}_{old\ \ value} + \underbrace{\alpha}_{learning\ \ rate} \cdot \left(\overbrace{\underbrace{r(\mathbf{s}_{t+1})}_{reward} + \underbrace{\gamma}_{discount\ \ factor} \cdot \underbrace{\max_{\mathbf{a}_{t+1}} Q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}_{estimate\ \ of\ \ optimal\ \ future\ \ value}}^{learned\ \ value} \right)$$

Learning rate $\alpha \in]0, 1[$
Discount factor $\gamma \in [0, 1]$

- $\gamma \to 0$: short-term rewards only, tractable
- $\gamma \to 1$: long-term rewards, intractable

Exploration / exploitation dilemma:

- sample (complete) sequences $\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{a}_{N-1}, \mathbf{s}_N$;
- update $Q(\mathbf{s}_t, \mathbf{a}_t)$ from $t = N-1$ to $t = 0$.

Value function learning (Q-learning)

Pros:

+ model-free
+ no expert required
+ long / short-term reward balance

Cons:

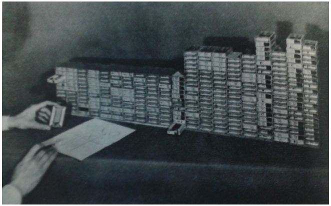
– requires intermediate rewards
– exploration / exploitation trade-off

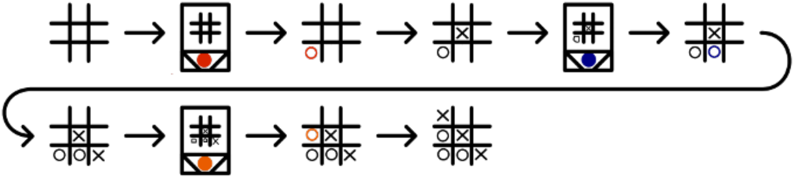
Atari games: `https://youtu.be/V1eYniJ0Rnk?t=20s` (Google Deepmind)
Physics engine: `https://goo.gl/LU8P5F` (OpenAI)
DotA2: `https://youtu.be/wpa5wyutpGc` (OpenAI)

Reinforcement learning: a quite old idea...

Matchbox Educable Noughts And Crosses Engine (MENACE)

Tic-Tac-Toe: 304 states (first player + symmetries).





http://mscroggs.co.uk/menace/

D. Michie (1961). Trial and Error.