Machine Learning for Economists and Business Analysts

Reinforcement Learning

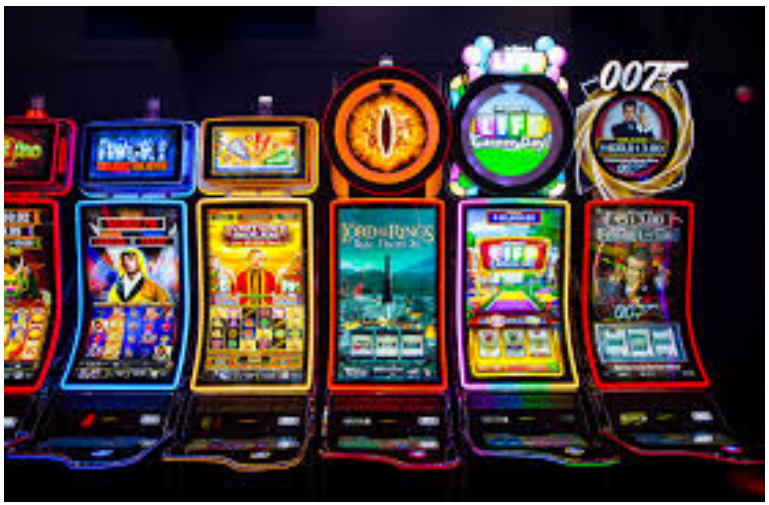
Anthony Strittmatter

Literature

- Sutton and Barto (2015): "Reinforcement Learning: An Introduction", 2nd ed., MIT Press, download, **Chapter 1-2**.
- Bubeck and Cesa-Bianchi (2012): "Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems", Foundations and Trends in Machine Learning, 5(1), download.

Multi-Armed Bandit Problem

- We stand in front of n different slot machines and can play each of them sequentially.
- Question: Playing which slot machine has the highest reward?



- **Action:** Playing slot machine $a \in \{a_1, a_2, ..., a_n\}$ at time t
- **Reward:** We observe the reward $r_{t+1}(a)$ in time period $t + 1$

Evaluative Feedback

- Find the action a that has the **maximum expected reward** over some time period (e.g. 1000 action selection steps)

$$\max_a E[r_t(a)]$$

- $E[r_t(a)]$ is not known in advance (uncertainty), but we may have an estimate/prior.

Action Value Function

- Let's say until time t action a has been selected $N_t(a)$ times and has yielded rewards $r_1(a), r_2(a), ..., r_{N_t(a)}(a)$.
- Then we can estimate the **action value function**

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{s=1}^{N_t(a)} r_s(a),$$

  which changes after every time period.
- For $N_t(a) = 0$ we have to initialize $Q_t(a)$ with some starting value, such as $Q_t(a) = 0$.
- When $N_t(a) \to \infty$, $Q_t(a)$ converges to $E[r_t(a)]$

Greedy Action Selection

- The reward is maximized by choosing the action a that maximizes $\max_a Q_t(a)$ from the set of possible actions $a \in \{a_1, a_2, ..., a_n\}$.
- Let's say we know that a_1 is a good action (salami pizza).
- We could be greedy and just **exploit** action a_1.
- But then we will not learn whether there is an even better action or not, because we do not observe the rewards of the other actions.

$\Rightarrow$ We need to **explore** if we want to learn!

Exploitation

Exploration

Exploration-Exploitation Trade-Off

- We need to explore to learn the action that maximizes the action value function.
- But during exploration we will make mistakes and select actions that do not maximize the expected rewards (trial and error approach).
- The idea is that decrease the amount of exploration over time and exploit the optimal action that we learned.

$\Rightarrow$ Finding the optimal balance between exploration and exploitation is challenging!

Possible Applications of Bandit Algorithms

- Selecting the design of an online advertisement platform.
- Dynamic pricing of flight tickets.
- Prescription of pills to patients.
- Assignment of training programs to unemployed persons.
- Movie recommendation on Netflix.

Full Randomization

- There are n different actions and $\pi(a)$ is the probability that we select action a.
- We call $\pi(a)$ a **policy**.
- We could select action a from an uniform distribution with probability

$$\pi(a) = \frac{1}{n}.$$

$\rightarrow$ Then we would only explore and not exploit!

Epsilon Greedy Algorithm

- We could select action a with probability

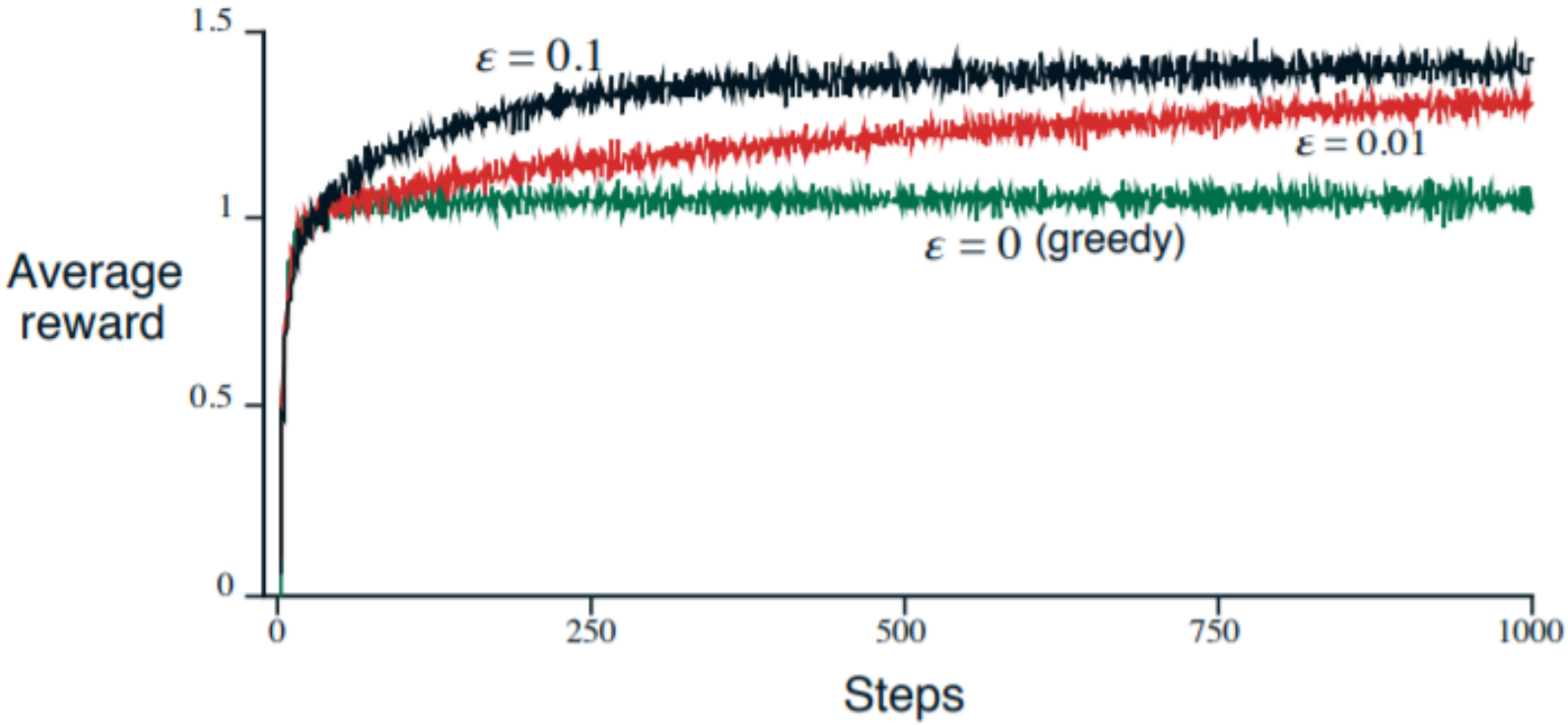
$$\pi(a) = \begin{cases} 1 - \varepsilon & \text{if } a = \max_a Q_t(a) \\ \varepsilon/(n-1) & \text{else,} \end{cases}$$

  where $0 \leq \varepsilon \leq 1$.

- By selecting ε we can balance exploration and exploitation:
  - $\rightarrow$ $\varepsilon = 1$ implies only exploration.
  - $\rightarrow$ $\varepsilon = 0$ implies only exploitation.

- The epsilon greedy algorithm gives preference to the action with the highest estimated reward.

Selecting Epsilon



Source: Sutton and Barto (2015)

Softmax Algorithm

- We could select action a with probability

$$\pi(a) = \frac{e^{Q_t(a)/\tau}}{\sum_{k=1}^{p} e^{Q_t(a_k)/\tau}},$$

  where $\tau > 0$.

- τ is called the **temperature**:
  - If $\tau > 1$ we put more weight on exploration.
  - If $0 < \tau < 1$ we put more weight on exploitation.

- The softmax algorithm takes account of the size of all estimated rewards and treats actions with almost similar estimated rewards almost similarly.

Accounting for Estimation Uncertainty

- Softmax takes account how close the estimated action value is to the optimal.
- The epsilon greedy and softmax algorithms do not account for uncertainty in the estimation of the action value function $Q_t(a)$.
- Both algorithms do not adapt the amount of exploration over time.
- But it would be better to select the actions according to their potential for actually being optimal, taking into account both how close their estimates are to being maximal and the uncertainties in those estimates.

Upper-Confident-Bound Algorithm

- The upper-confident-bound (UCB) algorithm selects actions according to

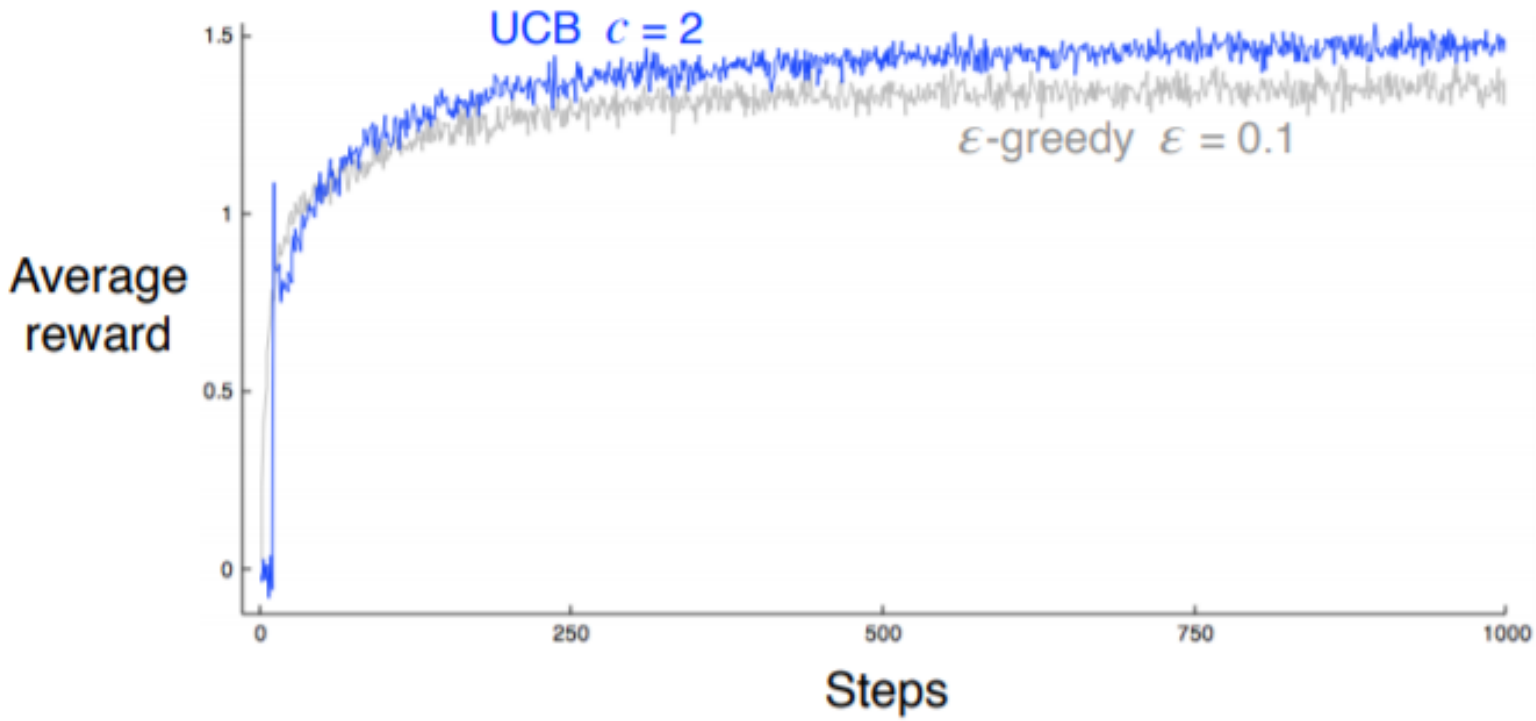
$$a = \max_{\{a_1, a_2, \ldots, a_n\}} \left[Q_t(a) + c \sqrt{\frac{log(t)}{N_t(a)}} \right]$$

- The constant $c \geq 0$ controls for the degree of exploration ($c = 0$ means no exploration).
- The *square-root-term* is a measure for uncertainty.
- We maximize over the upper bound of the possible true action values $Q_t(a)$.

Upper-Confident-Bound Algorithm

- Each time action a is selected the uncertainty is presumably reduced, because
  - $\rightarrow$ $N_t(a)$ is incremented and
  - $\rightarrow$ the uncertainty term is reduced (as $N_t(a)$ appears in the denominator)
- Each time an alternative action is selected the uncertainty is presumably increased, because
  - $\rightarrow$ t increases no matter if action a or an alternative action is selected
  - $\rightarrow$ the uncertainty term is increases (as t appears in the numerator)
  - $\rightarrow$ the logarithm implies that increases get smaller over time

Comparison with Epsilon Greedy Algorithm



Source: Sutton and Barto (2015)

Thompson Sampling for Gaussian Bandits

For each time period $t = 1, 2, ...$

1. For each action $a_1, a_2, ..., a_n$, draw a random value $\mu(a)$ from the normal distribution $\mu(a) \sim N(Q_t(a), Var(Q_t(a)))$.
2. Select the action a that maximizes

$$a = \max_{\{a_1, a_2, ..., a_n\}} \mu(a)$$

3. Observe the reward $r_{t+1}(a)$
4. Update $Q_{t+1}(a)$ and $Var(Q_{t+1}(a))$, with

$$Q_{t+1}(a) = \frac{1}{N_{t+1}(a)} \sum_{s=1}^{N_{t+1}(a)} r_s(a) \text{ and}$$

$$Var(Q_{t+1}(a)) = \frac{1}{N_{t+1}(a)} \sum_{s=1}^{N_{t+1}(a)} (r_s(a) - Q_s(a))^2$$

Thompson Sampling

- Thompson sampling is an alternative to the UVB algorithm.
- The estimated action values and their uncertainty are modelled explicitly.
- Actions with higher action values $Q_t(a)$ have a higher probability to be selected.
- Actions with large uncertainty about the true action value (e.g; with large variance $Var(Q_t(a))$) have a higher probability to be selected.

Incremental Updating of Action Value Function

$$
\begin{aligned}
Q_t(a) &= \frac{1}{N_t(a)} \sum_{s=1}^{N_t(a)} r_s(a) \\
&= \frac{1}{N_t(a)} \left(r_{N_t(a)}(a) + \sum_{s=1}^{N_t(a)-1} r_s(a) \right) \\
&= \frac{1}{N_t(a)} \left(r_{N_t(a)}(a) + (N_t(a)-1) \frac{1}{N_t(a)-1} \sum_{s=1}^{N_t(a)-1} r_s(a) \right) \\
&= \frac{1}{N_t(a)} \left(r_{N_t(a)}(a) + (N_t(a)-1) Q_{t-1}(a) \right) \\
&= Q_{t-1}(a) + \underbrace{\frac{1}{N_t(a)} \left(r_{N_t(a)}(a) - Q_{t-1}(a) \right)}_{\text{incremental updating}}
\end{aligned}
$$

$\Rightarrow$ Efficient storage of rewards!

Contextual Bandits

- The agent observes the context x (characteritics, covariates, features) before the decision.
- Now the optimal decision can be a function of the context.
- Example: Assume we observe the gender. A possible rule could be:
  - Select action a_1 if *male*.
  - Select action a_3 if *female*.
- In practice, the reward $r_t(a, x)$ and the value action function $Q_t(a, x)$ become functions of the action *and* the context.

$\Rightarrow$ We call contextual bandits *associative search algorithms*.