Reinforcement Learning

Multi-Armed Bandits

Stefano Albrecht, Pavlos Andreadis

17 January 2020

THE UNIVERSITY of EDINBURGH
informatics

Lecture Outline

- Multi-armed bandit problem
- Exploration-exploitation dilemma
- Action-value methods
- Gradient methods

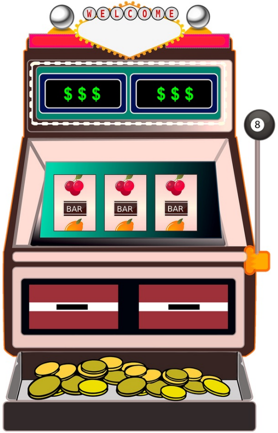
Multi-armed bandit problem:

- There are k actions ("arms") to choose from
- On each time step $t = 1, 2, 3, \ldots$, you choose an action $A_t = a$ and receive a scalar reward sampled from some *unknown* random variable R_t, where

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a]$$

R_t are iid (independently and identically distributed)

- Goal: maximise total received rewards over time



Exploration-Exploitation Dilemma

- We can form *action-value estimates*:

$$Q_t(a) \approx q_*(a)$$

- The greedy action at time t is:

$$A_t^* \doteq \arg\max_a Q_t(a)$$

- *Exploitation:* choose $A_t = A_t^*$; *Exploration:* choose $A_t \neq A_t^*$

Exploration-exploitation problem:

How to balance exploration and exploitation to maximise rewards?

$\Rightarrow$ Can't exploit or explore all the time *(why?)*

Action-value methods:

- Learn action-value estimates
- E.g. sample average:

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{\tau=1}^{t-1} R_\tau * [A_\tau = a]_1$$

  where $N_t(a)$ is number of times action a was selected until before t

- Sample average converges to true action values in the limit:

$$\lim_{N_t(a)\to\infty} Q_t(a) = q_*(a)$$

ϵ-Greedy Action Selection

- Greedy action selection:

$$A_t = A_t^* = \arg\max_a Q_t(a)$$

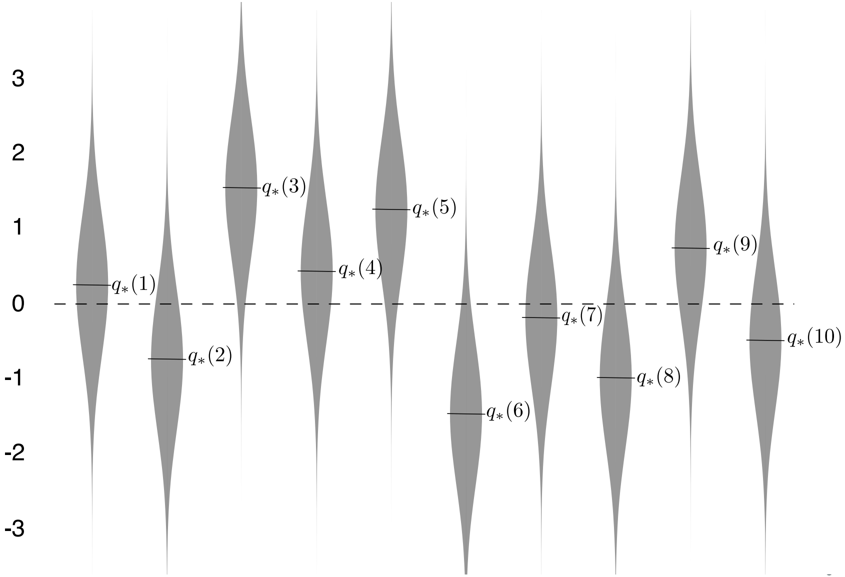
- ϵ-greedy action selection:

$$A_t = \begin{cases} A_t^* & \text{with probability } 1-\epsilon \\ \text{random action} & \text{otherwise} \end{cases}$$

- Simplest way to balance exploration and exploitation

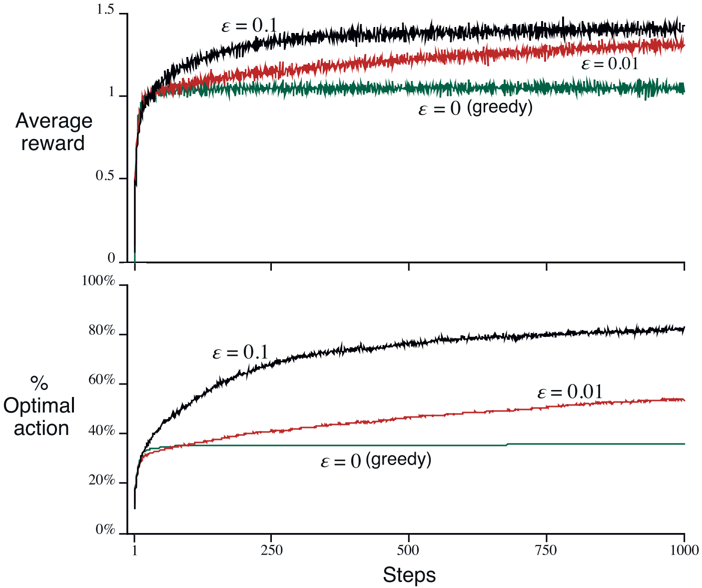
10-Armed Bandit Testbed

2000 random testbeds each with 1000 time steps



ϵ-Greedy Methods on the 10-Armed Testbed

Where is $\epsilon = 0.1$ after 10,000 time steps?



Averaging Learning Rule

- Sample average (for 1-armed bandit):

$$Q_n = \frac{R_1 + R_2 + ... + R_{n-1}}{n-1}$$

- Can compute incrementally:

$$Q_{n+1} = Q_n + \frac{1}{n}\left[R_n - Q_n\right]$$

- This is a standard form for update rules:

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize}\left[\text{Target} - \text{OldEstimate}\right]$$

Derivation of Incremental Update

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
&= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

Simple Bandit Algorithm

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$

Non-Stationary Action Values

Suppose the true action values change slowly over time

- We then say that the problem is *non-stationary*
- Sample average not appropriate (why?)
- Many RL methods have to deal with non-stationarity (e.g. due to bootstrapping)

Have to "track" action values, e.g. using step size parameter $\alpha \in (0, 1]$

$$\begin{aligned} Q_{n+1} &= Q_n + \alpha \left[R_n - Q_n \right] \\ & (1-\alpha)^n Q_1 + \textstyle\sum_{i=1}^{n} \alpha (1-\alpha)^{n-i} \, R_i \end{aligned}$$

$\Rightarrow$ *Exponential, recency-weighted average*

Standard Stochastic Approximation Convergence Conditions

Estimates $Q_t(a)$ will converge to true values $q_*(a)$ with probability 1 if:

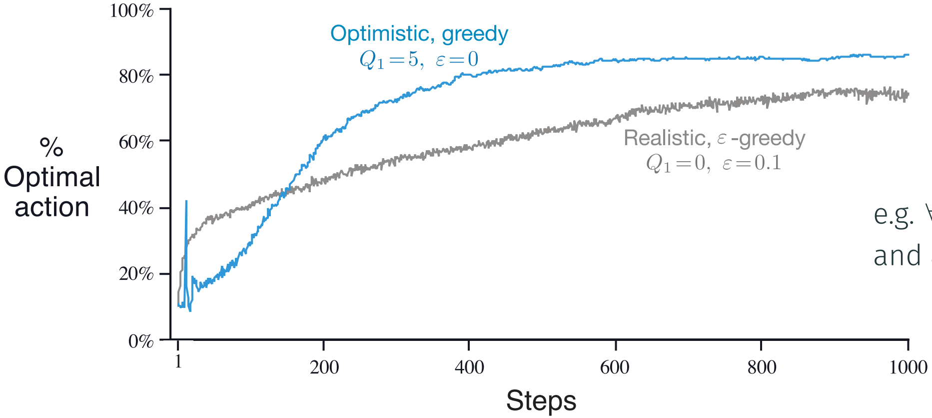
$$\sum_{n=1}^{\infty} \alpha_n(a) \to \infty \qquad \text{and} \qquad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

- e.g. $\alpha_n = \frac{1}{n}$
- not $\alpha_n = \frac{1}{n^2}$
- If $\alpha_n = n^{-p}$, $p \in (0, 1)$, then convergence is at optimal rate $O(1/\sqrt{n})$

Optimistic Initial Values

- All methods so far depend on $Q_1 \rightarrow$ they are *biased* by Q_1

  $\Rightarrow$ Can incentivise exploration by using "optimistic" initial values for $Q_1(a)$



e.g. $\forall a : Q_1(a) = 5$

and $\alpha = 0.1$

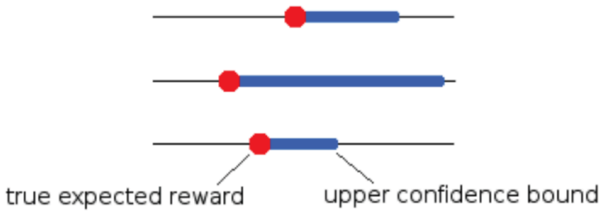
Upper Confidence Bound (UCB) Action Selection

Upper Confidence Bound (UCB): Instead of estimating action value directly, estimate upper bound on action value and choose action with highest bound:

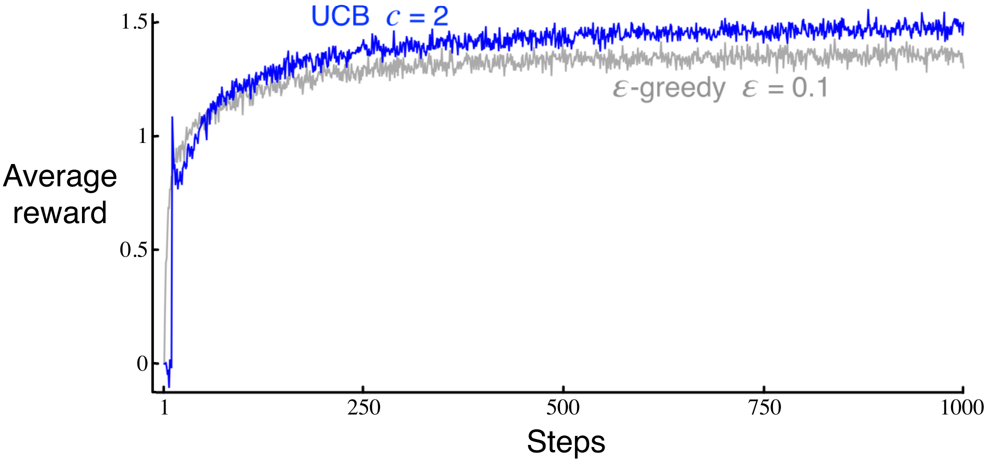
$$A_t = \begin{cases} a \ \text{ if } \ N_t(a) = 0, \ \text{ else} \\ \arg\max_a \left[Q_t(a) + c\sqrt{\log t / N_t(a)} \right] \end{cases}$$

(Standard UCB assumes rewards in $[0, 1]$ range)

Intuition: second term is size of one-sided confidence interval for average reward



Upper Confidence Bound (UCB) Action Selection



Gradient Bandit Algorithm

Greedy, ϵ-greedy, and UCB use estimates of $q_*(a)$

- Can we select actions without computing estimates of q_*?

Gradient Bandit Algorithm

Greedy, ϵ-greedy, and UCB use estimates of $q_*(a)$

- Can we select actions without computing estimates of q_*?

Action policy:

- $\pi_t(a) =$ probability of selecting action a at time t

  $\Rightarrow$ Use stochastic gradient ascent to optimise policy

Gradient Bandit Algorithm

Greedy, ϵ-greedy, and UCB use estimates of $q_*(a)$

- Can we select actions without computing estimates of q_*?

Action policy:

- $\pi_t(a) =$ probability of selecting action a at time t

  $\Rightarrow$ Use stochastic gradient ascent to optimise policy

- Need differentiable policy representation, e.g. *softmax* distribution:

$$\pi_t(a) \doteq \frac{e^{H_t(a)}}{\sum_b e^{H_t(b)}}$$

Gradient Bandit Algorithm

Greedy, ϵ-greedy, and UCB use estimates of $q_*(a)$

- Can we select actions without computing estimates of q_*?

Action policy:

- $\pi_t(a) =$ probability of selecting action a at time t

  $\Rightarrow$ Use stochastic gradient ascent to optimise policy

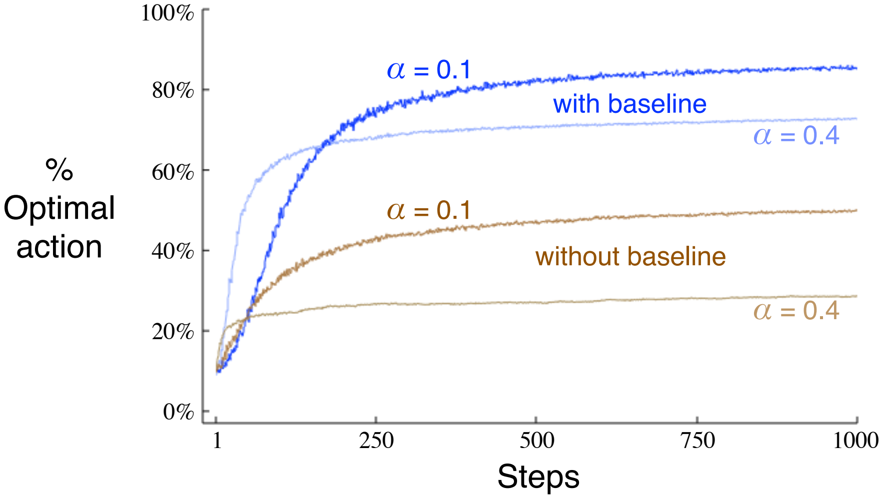
- Need differentiable policy representation, e.g. *softmax* distribution:

$$\pi_t(a) \doteq \frac{e^{H_t(a)}}{\sum_b e^{H_t(b)}}$$

- Update policy with

$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)([a = A_t]_1 - \pi_t(a)), \qquad \text{where } \bar{R}_t = \frac{1}{t}\sum_{\tau=1}^{t} R_\tau$$

Gradient Bandit Algorithm



$\bar{R}_t = \frac{1}{t} \sum_\tau R_\tau$

$\bar{R}_t = 0$

Baseline reduces variance in updates

In **exact** gradient ascent:

$$H_{t+1} \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} \quad \text{where} \quad \mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x)$$

Derivation of Gradient Bandit Algorithm

In **exact** gradient ascent:

$$H_{t+1} \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} \quad \text{where} \quad \mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x)$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi_t(x) q_*(x) \right]$$

Derivation of Gradient Bandit Algorithm

In **exact** gradient ascent:

$$H_{t+1} \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} \quad \text{where} \quad \mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x)$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi_t(x) q_*(x) \right]$$

$$= \sum_x q_*(x) \, \frac{\partial \pi_t(x)}{\partial H_t(a)} \qquad \text{(product derivative rule)}$$

Derivation of Gradient Bandit Algorithm

In **exact** gradient ascent:

$$H_{t+1} \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} \quad \text{where} \quad \mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x)$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi_t(x) q_*(x) \right]$$

$$= \sum_x q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} \qquad \text{(product derivative rule)}$$

$$= \sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)} \qquad \text{(B_t is “baseline”)}$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \pi_t(x)}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \pi_t(x)$$

Derivation of Gradient Bandit Algorithm

$$
\begin{aligned}
\frac{\partial \pi_t(x)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(x) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_y e^{H_t(y)}} \right]
\end{aligned}
$$

Derivation of Gradient Bandit Algorithm

$$\begin{aligned}
\frac{\partial \pi_t(x)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(x) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_y e^{H_t(y)}} \right] \\
&= \frac{\frac{\partial e^{H_t(x)}}{\partial H_t(a)} \sum_y e^{H_t(y)} - e^{H_t(x)} \frac{\partial \sum_y e^{H_t(y)}}{\partial H_t(a)}}{(\sum_y e^{H_t(y)})^2} \qquad \text{(quotient derivative rule)}
\end{aligned}$$

Derivation of Gradient Bandit Algorithm

$$
\begin{aligned}
\frac{\partial \pi_t(x)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(x) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_y e^{H_t(y)}} \right] \\
&= \frac{\frac{\partial e^{H_t(x)}}{\partial H_t(a)} \sum_y e^{H_t(y)} - e^{H_t(x)} \frac{\partial \sum_y e^{H_t(y)}}{\partial H_t(a)}}{\left(\sum_y e^{H_t(y)}\right)^2} && \text{(quotient derivative rule)} \\
&= \frac{[a = x]_1 \, e^{H_t(x)} \sum_y e^{H_t(y)} - e^{H_t(x)} e^{H_t(a)}}{\left(\sum_y e^{H_t(y)}\right)^2} && \left(\frac{\partial e^x}{\partial x} = e^x\right)
\end{aligned}
$$

Derivation of Gradient Bandit Algorithm

$$
\begin{aligned}
\frac{\partial \pi_t(x)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(x) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_y e^{H_t(y)}} \right] \\
&= \frac{\frac{\partial e^{H_t(x)}}{\partial H_t(a)} \sum_y e^{H_t(y)} - e^{H_t(x)} \frac{\partial \sum_y e^{H_t(y)}}{\partial H_t(a)}}{(\sum_y e^{H_t(y)})^2} && \text{(quotient derivative rule)} \\
&= \frac{[a = x]_1 \, e^{H_t(x)} \sum_y e^{H_t(y)} - e^{H_t(x)} e^{H_t(a)}}{(\sum_y e^{H_t(y)})^2} && \left(\frac{\partial e^x}{\partial x} = e^x\right) \\
&= \frac{[a = x]_1 \, e^{H_t(x)}}{\sum_y e^{H_t(y)}} - \frac{e^{H_t(x)} e^{H_t(a)}}{(\sum_y e^{H_t(y)})^2}
\end{aligned}
$$

Derivation of Gradient Bandit Algorithm

$$
\begin{aligned}
\frac{\partial \pi_t(x)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(x) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_y e^{H_t(y)}} \right] \\
&= \frac{\frac{\partial e^{H_t(x)}}{\partial H_t(a)} \sum_y e^{H_t(y)} - e^{H_t(x)} \frac{\partial \sum_y e^{H_t(y)}}{\partial H_t(a)}}{(\sum_y e^{H_t(y)})^2} && \text{(quotient derivative rule)} \\
&= \frac{[a = x]_1 \, e^{H_t(x)} \sum_y e^{H_t(y)} - e^{H_t(x)} e^{H_t(a)}}{(\sum_y e^{H_t(y)})^2} && \left(\frac{\partial e^x}{\partial x} = e^x\right) \\
&= \frac{[a = x]_1 \, e^{H_t(x)}}{\sum_y e^{H_t(y)}} - \frac{e^{H_t(x)} e^{H_t(a)}}{(\sum_y e^{H_t(y)})^2} \\
&= [a = x]_1 \, \pi_t(x) - \pi_t(x) \, \pi_t(a) \qquad = \pi_t(x) \left([a = x]_1 - \pi_t(a) \right)
\end{aligned}
$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)} / \pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x)\text{)}$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)}/\pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x)\text{)}$$

$$= \mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)}/\pi_t(A_t)\right] \qquad \text{(write as expectation over } x\text{)}$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)} / \pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x)\text{)}$$

$$= \mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad \text{(write as expectation over } x\text{)}$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad (\mathbb{E}[R_t|A_t] = q_*(A_t) \text{ and } B_t \doteq \bar{R}_t)$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)} / \pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x)\text{)}$$

$$= \mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad \text{(write as expectation over } x\text{)}$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad (\mathbb{E}[R_t|A_t] = q_*(A_t) \text{ and } B_t \doteq \bar{R}_t)$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t)\,\pi_t(A_t)([a = A_t]_1 - \pi_t(a))/\pi_t(A_t)\right]$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)} / \pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x))$$

$$= \mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad \text{(write as expectation over } x)$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \qquad (\mathbb{E}[R_t|A_t] = q_*(A_t) \text{ and } B_t \doteq \bar{R}_t)$$

$$= \mathbb{E}\big[(R_t - \bar{R}_t)\, \pi_t(A_t)([a = A_t]_1 - \pi_t(a)) / \pi_t(A_t)\big]$$

$$= \mathbb{E}\big[(R_t - \bar{R}_t)([a = A_t]_1 - \pi_t(a))\big]$$

Derivation of Gradient Bandit Algorithm

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)}/\pi_t(x) \qquad \text{(multiply by } \pi_t(x)/\pi_t(x))$$

$$= \mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)}/\pi_t(A_t)\right] \qquad \text{(write as expectation over } x)$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)}/\pi_t(A_t)\right] \qquad (\mathbb{E}[R_t|A_t] = q_*(A_t) \text{ and } B_t \doteq \bar{R}_t)$$

$$= \mathbb{E}\big[(R_t - \bar{R}_t)\, \pi_t(A_t)([a = A_t]_1 - \pi_t(a))/\pi_t(A_t)\big]$$

$$= \mathbb{E}\big[(R_t - \bar{R}_t)([a = A_t]_1 - \pi_t(a))\big]$$

Thus:

$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)([a = A_t]_1 - \pi_t(a)) \quad \square$$

Derivation of Gradient Bandit Algorithm

$$\sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)} = \sum_x (q_*(x) - B_t)\, \pi_t(x) \left([a = x]_1 - \pi_t(a)\right)$$

Baseline B_t does not change expectation because:

Derivation of Gradient Bandit Algorithm

$$\sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)} = \sum_x (q_*(x) - B_t)\, \pi_t(x) \left([a = x]_1 - \pi_t(a) \right)$$

Baseline B_t does not change expectation because:

$$\begin{aligned}
&\sum_x \left(q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} - B_t \frac{\partial \pi_t(x)}{\partial H_t(a)} \right) \\
&= \sum_x q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} - \sum_x B_t \frac{\partial \pi_t(x)}{\partial H_t(a)} \\
&= \ldots - B_t \underbrace{\sum_x \frac{\partial \pi_t(x)}{\partial H_t(a)}}_{=0 \quad \text{because...}}
\end{aligned}$$

Derivation of Gradient Bandit Algorithm

$$\sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)} = \sum_x (q_*(x) - B_t)\, \pi_t(x) \left([a = x]_1 - \pi_t(a)\right)$$

Baseline B_t does not change expectation because:

$$\begin{aligned}
&\sum_x \left(q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} - B_t \frac{\partial \pi_t(x)}{\partial H_t(a)} \right) \\
&= \sum_x q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} - \sum_x B_t \frac{\partial \pi_t(x)}{\partial H_t(a)} \\
&= \ldots - B_t \underbrace{\sum_x \frac{\partial \pi_t(x)}{\partial H_t(a)}}_{=0 \quad \text{because...}}
\end{aligned}$$

$$\begin{aligned}
&\sum_x \pi_t(x) \left([a = x]_1 - \pi_t(a)\right) \\
&= \sum_x \pi_t(x) [a = x]_1 - \sum_x \pi_t(x)\, \pi_t(a) \\
&= \pi_t(a) - \sum_x \pi_t(x)\, \pi_t(a) \\
&= \pi_t(a) - \pi_t(a) \underbrace{\sum_x \pi_t(x)}_{=1} = 0
\end{aligned}$$

Deterministic Policies

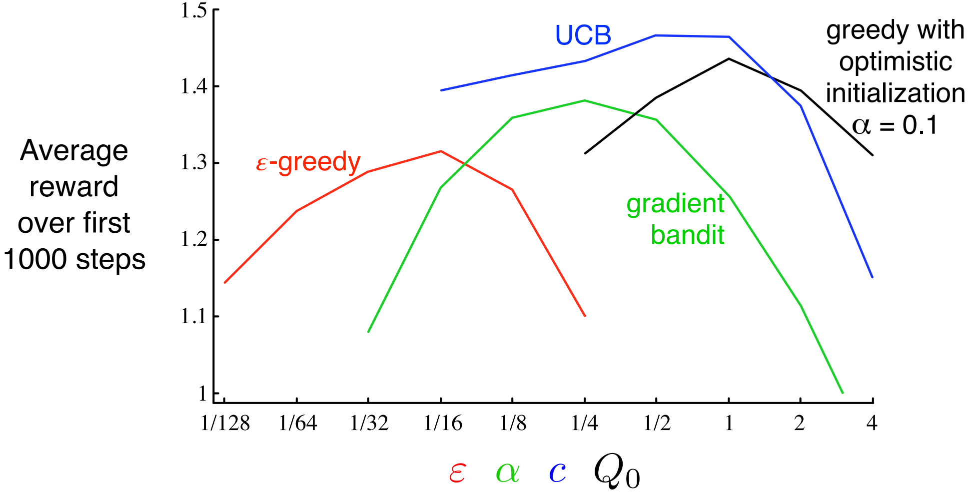
$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)([a = A_t]_1 - \pi_t(a))$$

Bonus questions:

- What if some actions have zero probability?
- E.g. what if initial policy is *deterministic*?

$$\pi_1(a) = 1 \text{ for some } a$$

Summary Comparison of Bandit Algorithms



Conclusion

Multi-armed bandit problem is simplest type of RL problem

- Bandit algorithms seek to maximise total reward over extended time
- Must balance exploration and exploitation – a key problem in RL
- First building block for more complex RL algorithms

Reading

Required:

- RL book, chapter 2

Optional:

- UCB paper:
  P. Auer, N. Cesa-Bianchi, P. Fischer (2002). Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47(2-3), 235-256.
- *Bandit Algorithms*
  by Tor Lattimore and Csaba Szepesvári
  Free download: `http://downloads.tor-lattimore.com/banditbook/book.pdf`