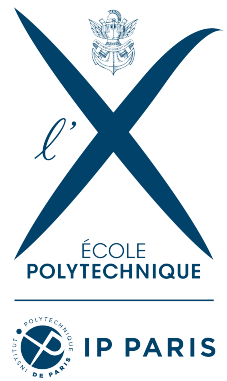
Bandit Machines

CSC_52081_EP Advanced Machine Learning and Autonomous Agents

Jesse Read



Last Updated: January 22, 2025

Agenda for Today

1. Information regarding course project (and other assessment items)
2. A quick word on Monte Carlo methods (and importance sampling) – as it relates to the Lab
3. From multi-label decisions (last week) to sequential decisions.
4. Bandits
5. Bayesian inference (and Bayesian bandits)
6. Bandits with importance sampling
7. A final word on Monte Carlo methods
   (Monte Carlo Tree Search – making use of bandits)

Monte Carlo Approximation

Objective: something like

$$\mathbb{E}_{X \sim P}[f(X)]$$

for example, for our value function

$$Q(a) = \mathbb{E}_{S_T \sim P(S_T | \boldsymbol{x})}[r(S_T, a)]$$

Suppose that this is intractable.

Vanilla Monte Carlo: Take samples, $x_m \sim P$, then estimate

$$\mathbb{E}_{X \sim P}[f(X)] \approx \frac{1}{M} \sum_{m=1}^{M} f(x_m)$$

Potential issue: $X \sim P$ (or, our example:
$S_T \sim P(S_T \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$) may be difficult to sample from.

Can we sample from some *other* P instead? For example,
$P(S_t \mid S_{t-1})$?

Importance Sampling

$$\begin{aligned}\mathbb{E}_{X\sim P}[f(X)] &= \int p(x)f(x)\,\mathrm{d}x \\ &= \int \frac{p(x)f(x)}{q(x)}q(x)\,\mathrm{d}x \quad \triangleright \text{ since } k = \frac{q}{q}k \\ &= \mathbb{E}_{X\sim Q}\left[\frac{p(x)f(x)}{q(x)}\right] \quad \triangleright \text{ definition of expectation}\end{aligned}$$

It means we can sample from[1] Q instead of P, i.e.,

$$\mathbb{E}_{X\sim Q}[f(X)] \approx \frac{1}{M}\sum_{m=1}^{M} f(x_m)\cdot\omega_m(x_m)$$

where $x_m \sim Q(x)$ and importance weight $\omega_m(x_m) = \frac{p(x_m)}{q(x_m)}$ to take into account we have sampled from the 'wrong' distribution.

[1] Note on notation: here, Q is *not* a value function here; it is a distribution

Outline

Goals of Today

Week 1: Deciding on actions
Week 2: Learning and approximate inference via search mechanisms (e.g., Monte Carlo search). We can do:

$$\boldsymbol{a}_* \approx \underset{\boldsymbol{a} \in \mathcal{A}^T}{\operatorname{argmax}} \, \mathbb{E}_{\boldsymbol{S} \sim P_\theta(\boldsymbol{S}|\boldsymbol{x})}[r(\boldsymbol{a}, \boldsymbol{S})]$$

where θ parameters learned from training examples $\{\boldsymbol{x}_i, \boldsymbol{a}_i\}_{i=1}^n$.

Today:

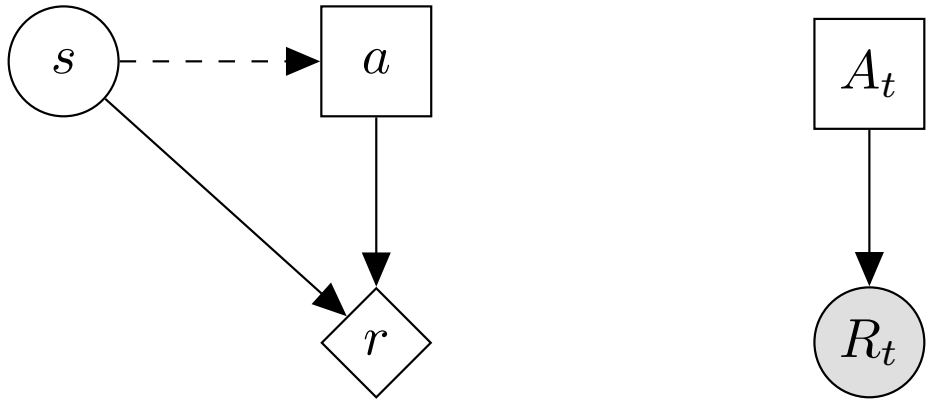
$$a_t^* = \underset{a_t \in \mathcal{A}}{\operatorname{argmax}} \, \mathbb{E}_{R \sim P_\theta^{(t)}}(R \mid a_t)[R]$$

- There's no observation! R becomes a random variable
- Action a_t chosen, and *then* r_t is observed ('training pair' r_t, a_t); i.e., more knowledge, we update $P_\theta^{(t+1)}$, which influences our choice at a_{t+1} which influences r_{t+1} which ....

Exploration vs exploitation.

Umbrella Example: Sequential Decisions

Should I take my umbrella today (no observation)?



$$a_t^* = \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}_{S \sim P_\theta(S)}[r(S, a)]$$

$$= \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}_{R \sim \nu_a}[R] \quad \triangleright \text{ Here we have marginalised out } S$$

we can use Monte Carlo approximation as needed.

But what if we start at $t = 1$ (it's our first day on earth). To get samples to update our knowledge, we need to choose some a_t. Then a_{t+1}, a_{t+2}. How to do this optimally?

Introduction to Bandits

Bandits

We consider stochastic multi-arm bandits.

Minimal game example: ‘Rock-Paper-Scissors’.

Motivating Example: Adaptive Drug Trials

A doctor, specialising in treatment of a particular disease, can prescribe one of k possible drugs (bandit arms). S/he sees a patient, assigns a drug (= takes an action/pulls a bandit arm), and observes the effectiveness (= reward), then proceeds to the next patient, and thus in sequence. The goal is to cure as many patients as possible.

For patient $t = 1, 2, \ldots, T$:

1. Give drug $A_t \in \{1, \ldots, k\}$ (action) to patient t
2. Observe outcome (reward) $R_t \in \{0, 1\}$ $\quad \triangleright\ 1 \equiv$ recovered
3. Update knowledge about the estimated outcome: value $Q(a)$

Bandit Applications

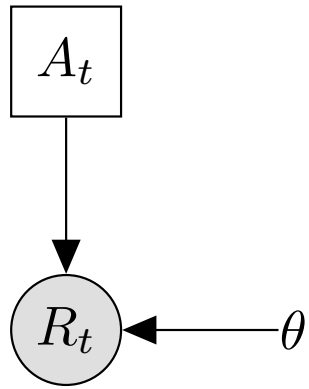
- Medical trials (prescribe drugs; maximize health outcome)
- Games (choose action; maximize chances of winning)
- Finance (invest in stock; maximize payoff)
- Influence Maximization (select influencer; maximize influence)
- Services (select provider to use; maximize service quality)
- Routing (select route; maximize data-transfer)
- Recommender systems (recommend movie; maximize rating)
- Marketing (which ad. to display; maximize revenue from ads)
- Robotics and logistics (select strategy; maximize productivity)






Bandits: Reward and Value

We take an action $A_t \in \{1, \ldots, k\}$ and receive reward $R_t \in \mathbb{R}$:



We maximize expected reward (value):

$$q_*(a) = \mathbb{E}_{R_t \sim \nu_a}[R_t \mid A_t = a]$$

where $R_t \sim \nu_a \Leftrightarrow R_t \sim \nu(R \mid A_t = a)$.

We want to learn about distributions $\nu_1, \ldots, \nu_k$,

in order to learn $Q \approx q_*$.,

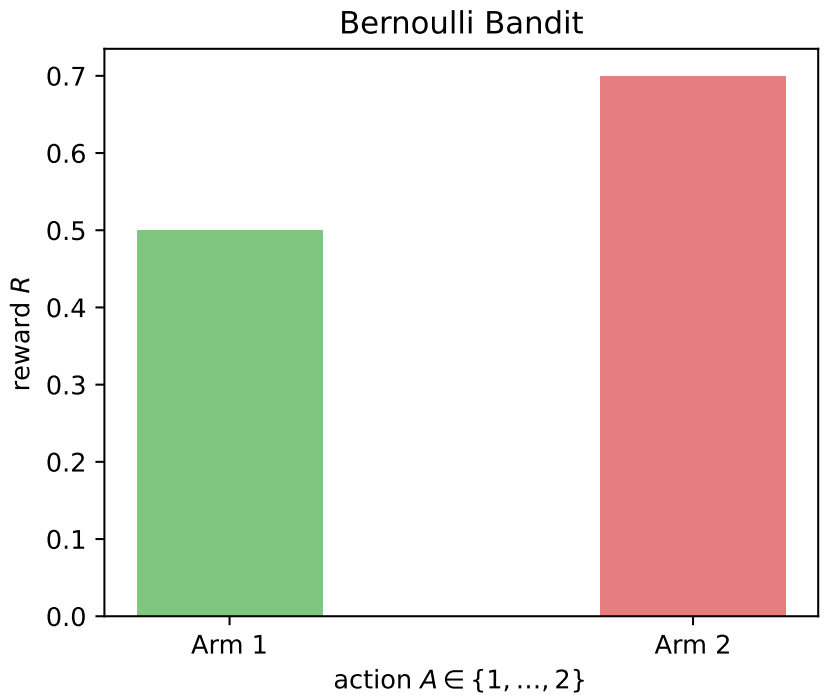
with which to maximize cumulative reward $\sum_{t=1}^{T} r_t$.

An Example: Bernoulli Bandit

Give drug A_t (of k to choose from) to patient t.

Observe reward $R_t \in \{0, 1\}$ ($r_t = 1 \Leftrightarrow$ patient recovered)

Question to answer: Which action A_t? (and in particular for $t = 1, t = T$?)

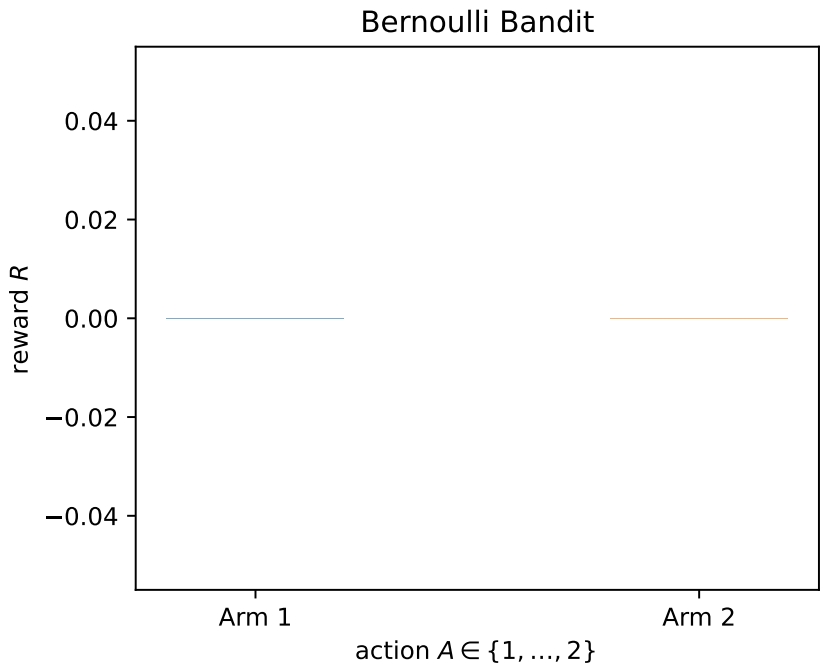


An Example: Bernoulli Bandit

Give drug A_t (of k to choose from) to patient t.

Observe reward $R_t \in \{0, 1\}$ ($r_t = 1 \Leftrightarrow$ patient recovered)

Question to answer: Which action A_t? (and in particular for $t = 1, t = T$?)



Multiple vs Sequential Decision Making

Earlier: For each test instance the 'agent'/model h:

1. Observes X
2. Takes 'action' $\hat{\boldsymbol{Y}} = h(X)$

(training is done 'offline' and prior to testing)

Generic bandits framework: For $t = 1, \ldots, T$, the agent (with policy π)

1. Takes action $A_t \sim \pi$
2. Observes reward R_t
3. Update our agent π, under new knowledge (A_t, R_t)

(Note: R_t *only* wrt A_t)

Reward and Regret

The expected reward (value) of an arm (action, decision) at time t:

$$q_*(a) := \mathbb{E}_{R_t \sim \nu_a}[R_t \mid A_t = a]$$

The highest value (best reward) is obtained with

$$a^* = \operatorname*{argmax}_{a \in \mathcal{A}} q_*(a)$$

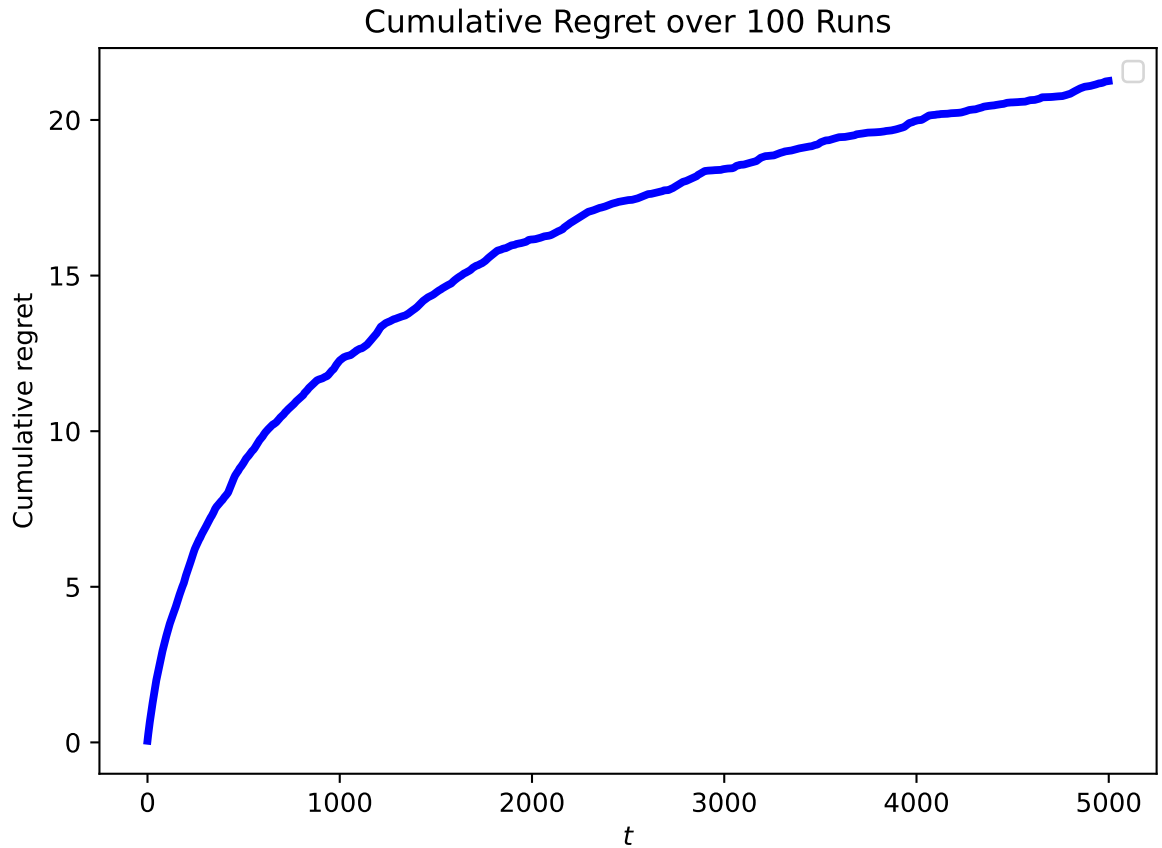
At round T, the regret for our policy (decision-making) π_t is

$$\mathcal{R}_T = q_*(a^*)T - \underbrace{\mathbb{E}_{A_t \sim \pi_t}\left[\sum_{t=1}^{T} R_t\right]}_{G}$$

Goal: **Regret should be as small as possible**! Intuition: act such that in hindsight, couldn't have done any better.

Optimal decisions lead to regret of 0, but not possible under 'cold start'; we have to learn! To learn we have to make mistakes ...

Expected behaviour ($\mathcal{R}_t$ vs t):



$\log(T)$ can be considered as a solution to the bandits problem.

Our policy π_t is updated (learning)! $\mathbb{E}_{A_t \sim \pi_t}[R_t]$ rises wrt t.

Exploitation vs Exploration

- Exploit: Take greedy actions (best under current knowledge).
- Explore: Improve knowledge for the future (i.e., learning).

The dilemma: We cannot do both always. It's a trade-off.

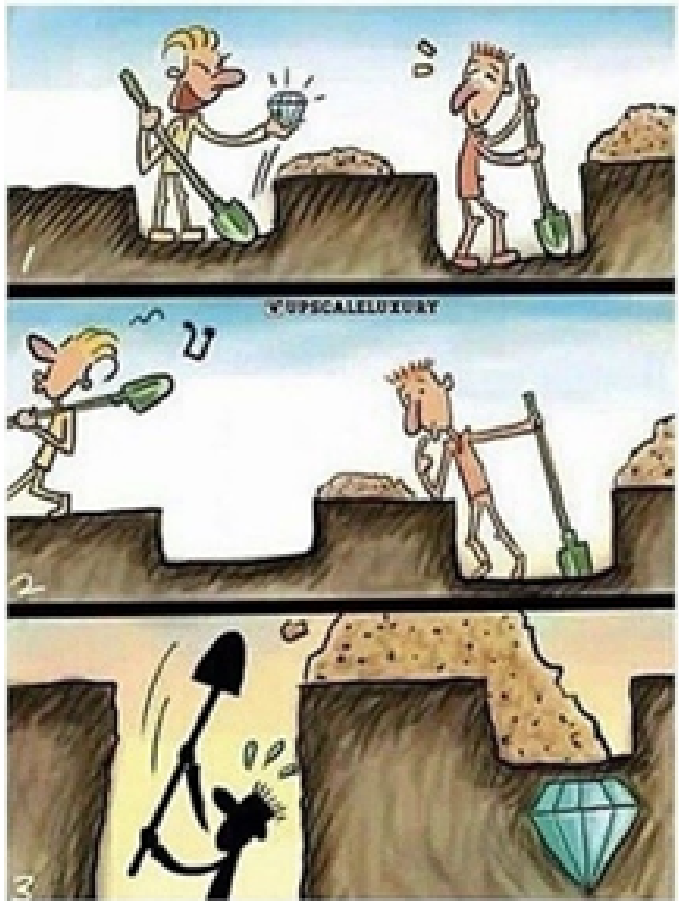


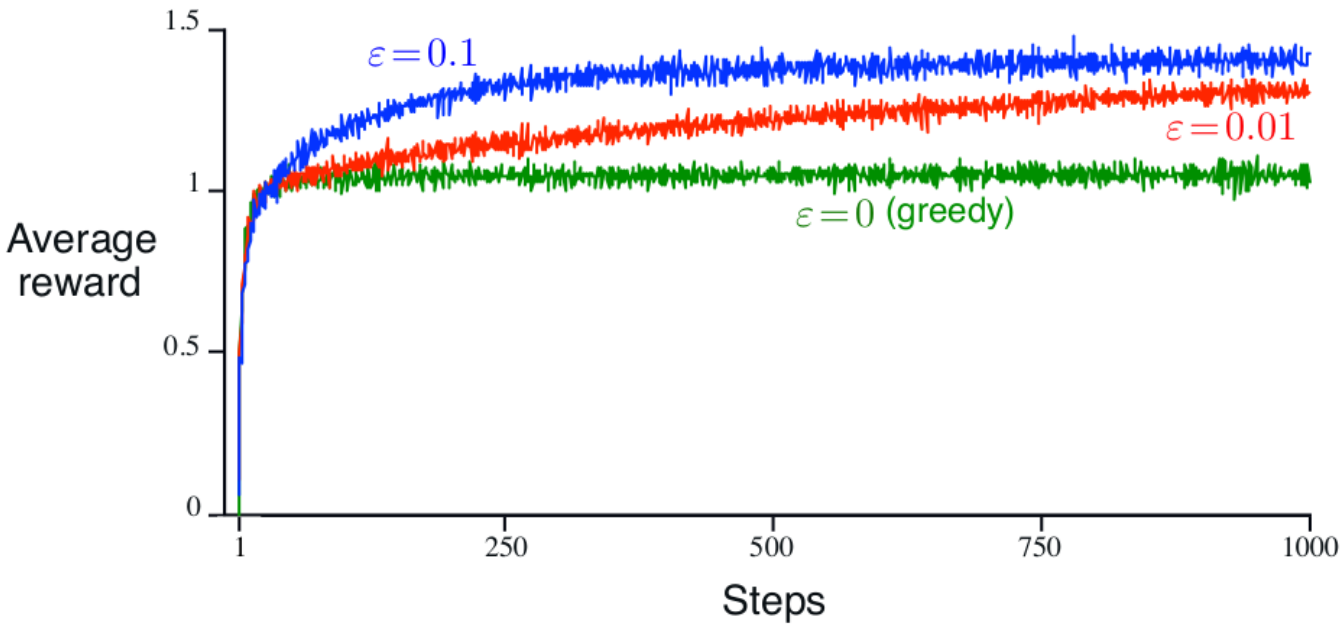
Image Source: [1]

The ϵ-Greedy Algorithm

ϵ-Greedy[2]: play the best arm with probability $1 - \epsilon$ (greedy action):

$$\hat{a}^* = \underset{a \in \mathcal{A}}{\text{argmax}}\, Q_\theta(a)$$

and (else) play a random arm $a \sim \mathcal{A}$ with probability ϵ (explore). Then, we can update value function $Q_\theta(a)$ for the arm played.

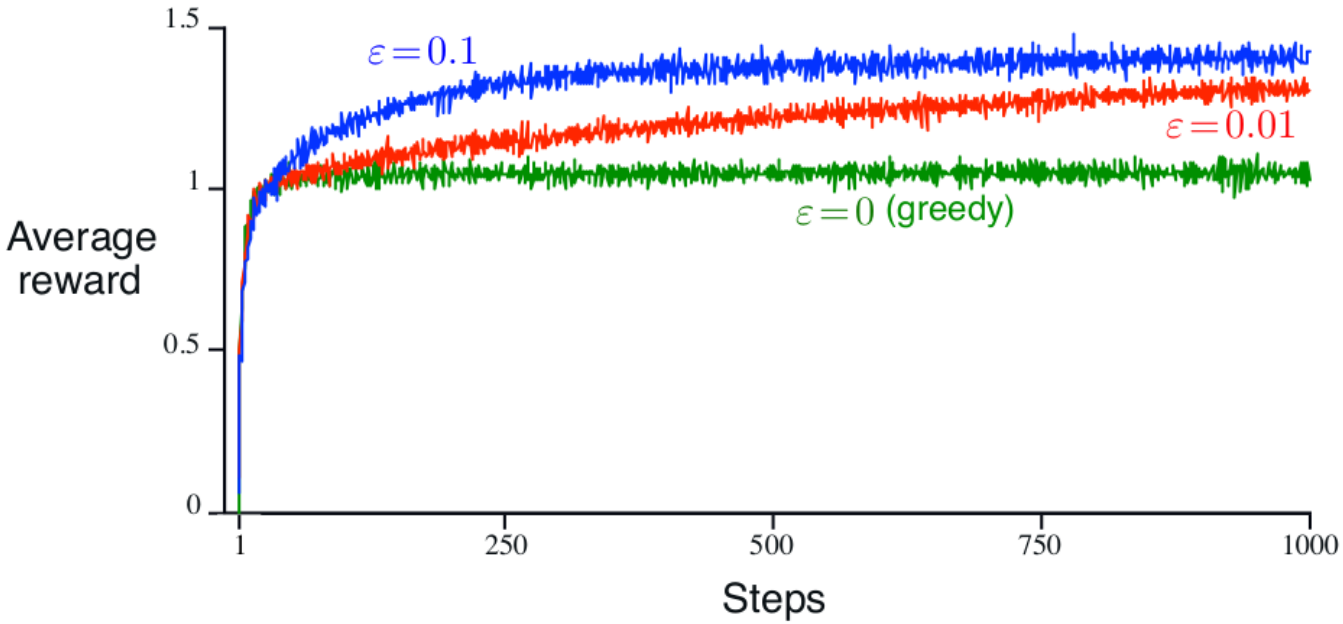


ϵ-greedy vs greedy. Ref: Sutton and Barto. *Reinforcement Learning: An Introduction*, 2020.

[2] N.B. Related to ϵ-approximate tree search, but it's not the same!

The Upper Confidence Bounds (UCB) Algorithm

Problems with ϵ-Greedy



ϵ-greedy vs greedy. Ref: Sutton and Barto. *Reinforcement Learning: An Introduction*, 2020.

- ϵ may be too large (over-exploring) or too small (over-greedy)
- algorithm may be biased by initial estimates of $Q_\theta(a)$

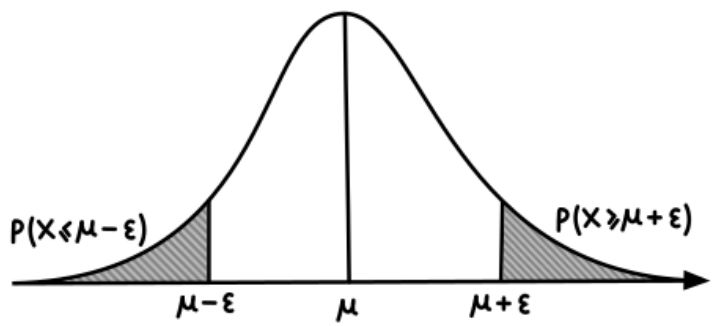
Tail Probabilities

Suppose iid samples from a-th arm: $R_1, R_2, \ldots, R_T$ ($R_t \sim \nu_a$).

Our unbiased estimate $\hat{\mu}$ (empirical mean over T draws)[3]:

$$\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} R_t$$

How confident can we be that is $\hat{\mu}$ is not 'too far off' (within ϵ of) the true mean μ?



We're interested in the tail probabilities. See, e.g., Lattimore and Szepesvári, Bandit Algorithms, 2020. Chap. 5. for derivations.

[3] Recall, $Q(a) = \mu(a)$, mean of distribution ν_a

Upper-Confidence-Bounds (UCB)

Recall: Only one arm can be pulled per round!

Let

- $N_t(a)$ the number of samples from arm a seen by time-step t
- $Q_t(a)$ empirical mean (estimate of $q_*(a)$ at time t).

$$\text{UCB}_{t-1}(a) = Q_{t-1}(a) + \sqrt{\frac{\alpha \ln t}{N_{t-1}(a)}}$$

where $\text{UCB}_{t-1} = \infty$ when $N_t(a) = 0$; exploration-factor α.

- Optimism in the face of uncertainty: $\text{UCB}_t(a)$ is the highest plausible value for arm a
- Essentially two terms: exploitation + exploration
- Optimism will decay with experience.
- The true $q_*(a)$ expected in confidence interval around $Q(a)$.

The UCB Algorithm

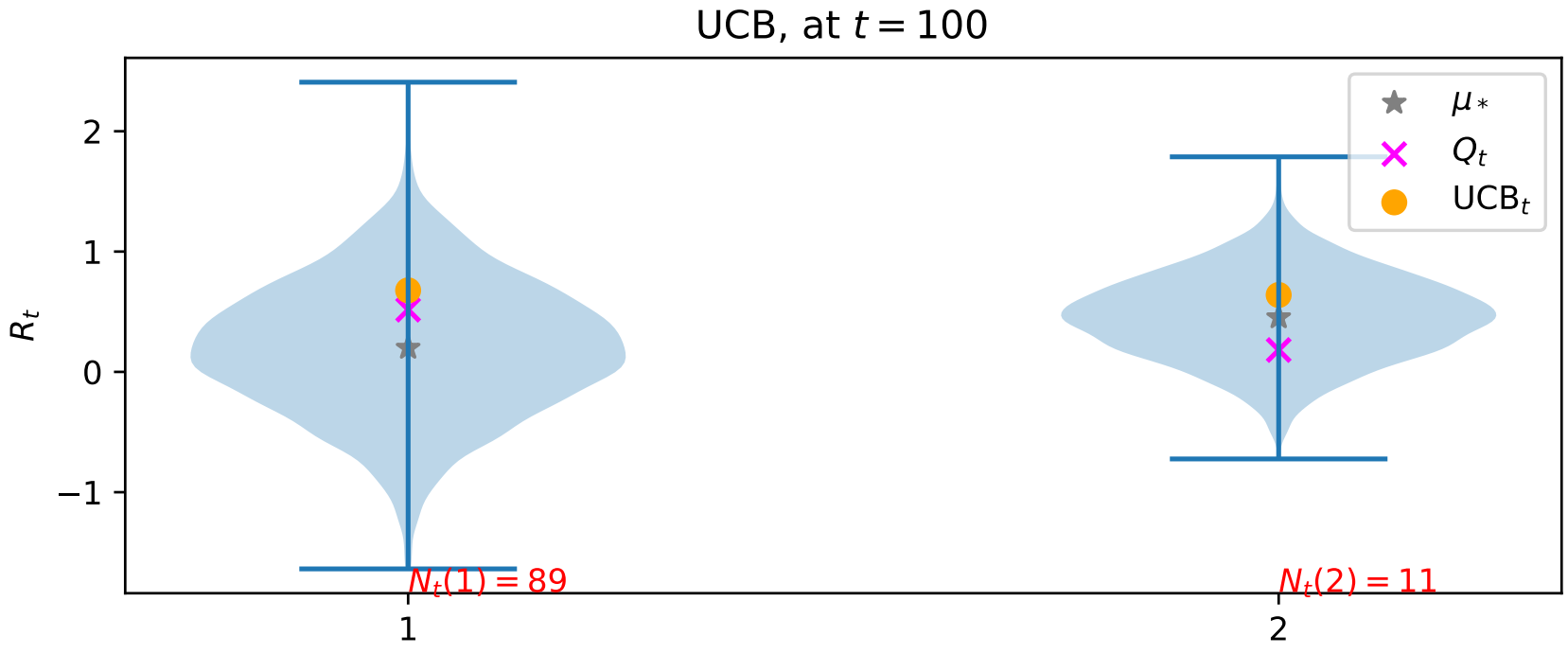
1: **procedure** UCB ALGORITHM(α)
2: Init. $\{N_0(a)\}_{a\in\mathcal{A}} = 0$ and $\{\text{UCB}_0(a)\}_{a\in\mathcal{A}} = \infty$
3: **for** $t = 1, \ldots, T$ **do**
4: Choose action

$$A_t = \operatorname*{argmax}_{a\in\mathcal{A}} \text{UCB}_{t-1}(a)$$

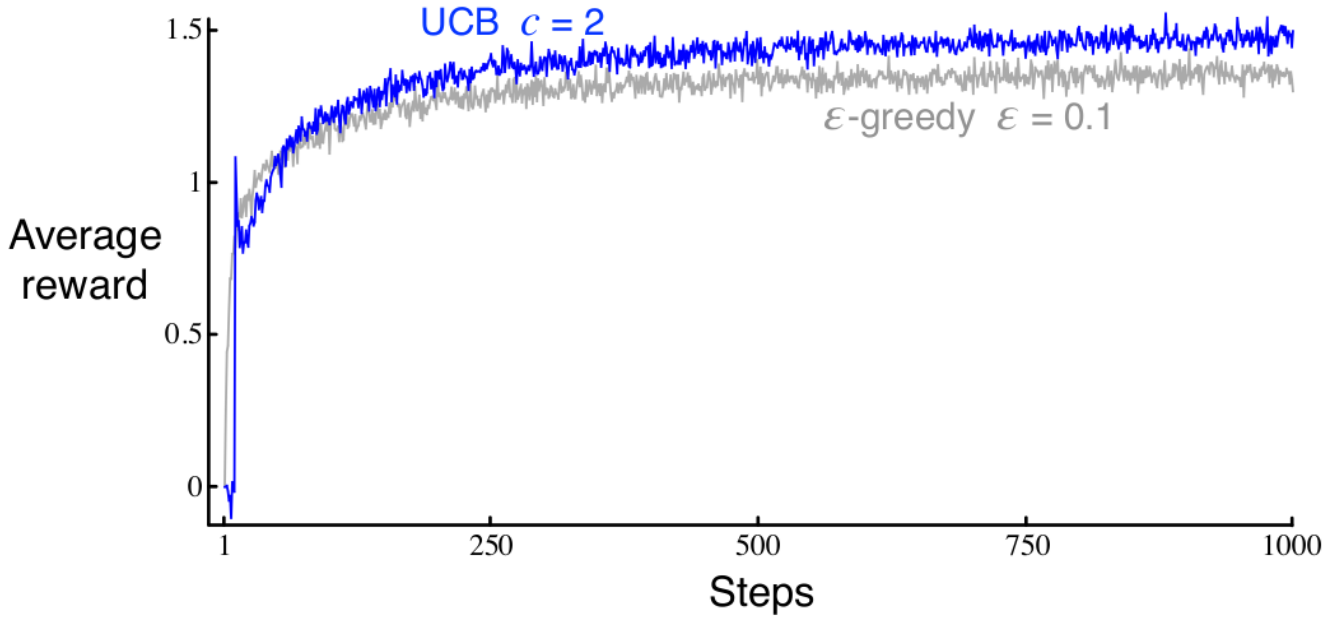
5: Pull the arm A_t and observe reward R_t
6: Update UCB_a: $N_t(A_t)$ and $Q_t(a)$

for choice of α: should be small enough to ensure reasonable optimism, but not so large as to encourage too much exploration of suboptimal arms.

General idea: Choosing the arm with the largest UCB implies only playing only arms a where the true $q_*(a)$ could reasonably be larger than the arms which have been played often.



UCB in action on Gaussian bandit machine (2 arms).



Sutton and Barto. *Reinforcement Learning: An Introduction*, 2020.

Theorem (Auer et al., 2002)

If the UCB1 policy is run over k arms, of arbitrary reward distributions $\nu_1, \ldots, \nu_k$ with support $\in [0, 1]$, expected regret is

$$\mathcal{R}_T \leq \left[4\alpha \sum_{a:\mu_a<\mu^*} \left(\frac{\log T}{\Delta_a} \right) \right] + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_{a=1}^{k} \Delta_a \right)$$

where $\Delta_a := \mu^* - \mu_a$.

$$\mathcal{O}(\sqrt{kT \log T})$$

where $q_* = \mu_*$, and $Q(a) = \hat{\mu}_a$.

Bayesian Inference (A Quick Introduction)

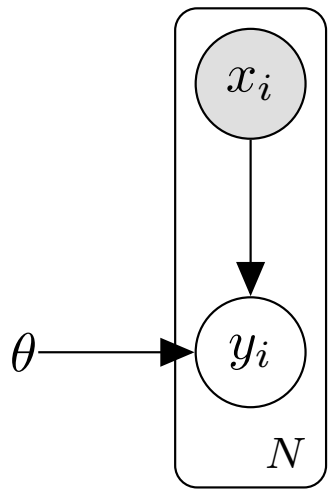
Classical Inference, Maximum Likelihood Estimation (MLE)

Given dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, write out the [log] likelihood,

$$L(\theta) = \log \prod_{i=1}^{N} p_\theta(x_i, y_i)$$

then maximise wrt θ (normally, by taking the gradient, $\nabla_\theta L(\theta)$),

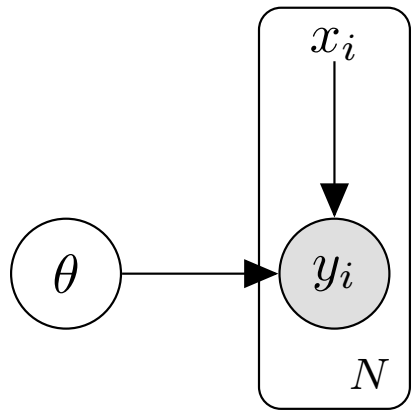
$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, L(\theta)$$



At inference time (e.g., linear regression, mean-squared-error),

$$\widehat{y} = f_{\hat{\theta}}(x) = \mathbb{E}_{Y \sim p_\theta(Y|x)}[Y] = \theta^\top x$$

Bayesian Inference

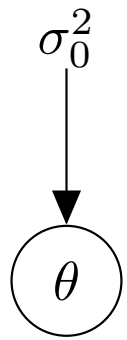


What's interesting here? What's different?

Note that θ is a variable, it has a distribution!

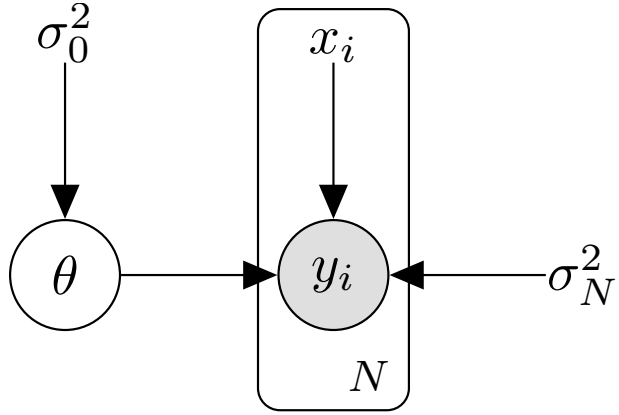
$$P(\theta, \boldsymbol{y}) = P(\boldsymbol{y} \mid \theta) P(\theta)$$

Bayesian Linear Regression: The Prior Distribution



$$P(\theta) = \mathcal{N}(\theta \mid 0, \sigma_0^2)$$

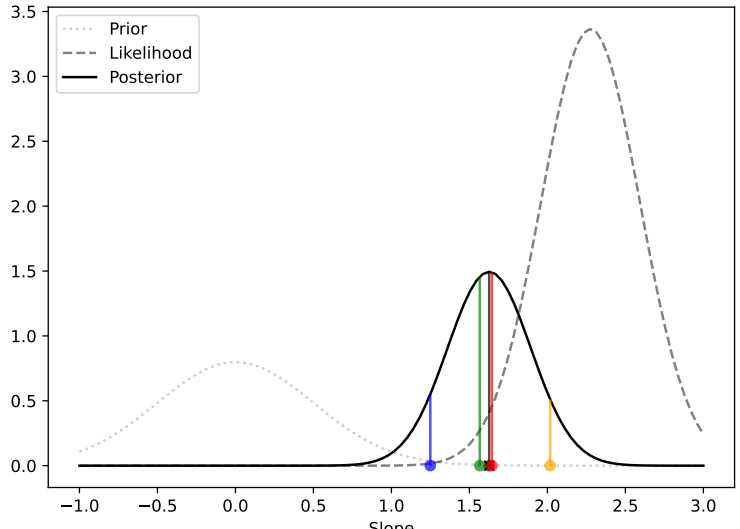
Bayesian Linear Regression: The Posterior Distribution



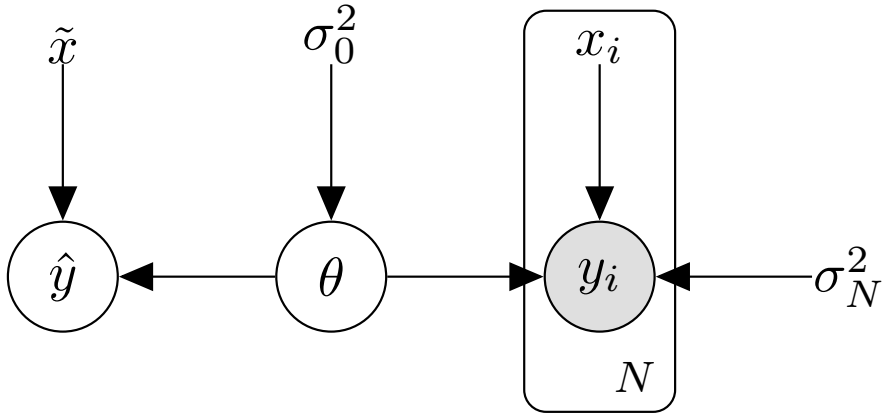
We can query

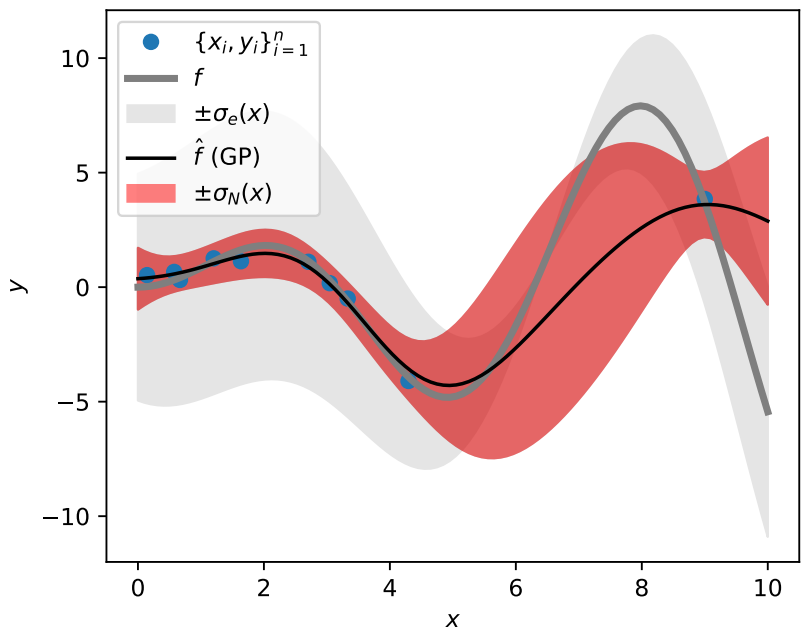
$$P(\theta \mid \boldsymbol{y}) = \frac{P(\theta, \boldsymbol{y})}{P(\boldsymbol{y})} \propto P(\boldsymbol{y} \mid \theta) P(\theta)$$

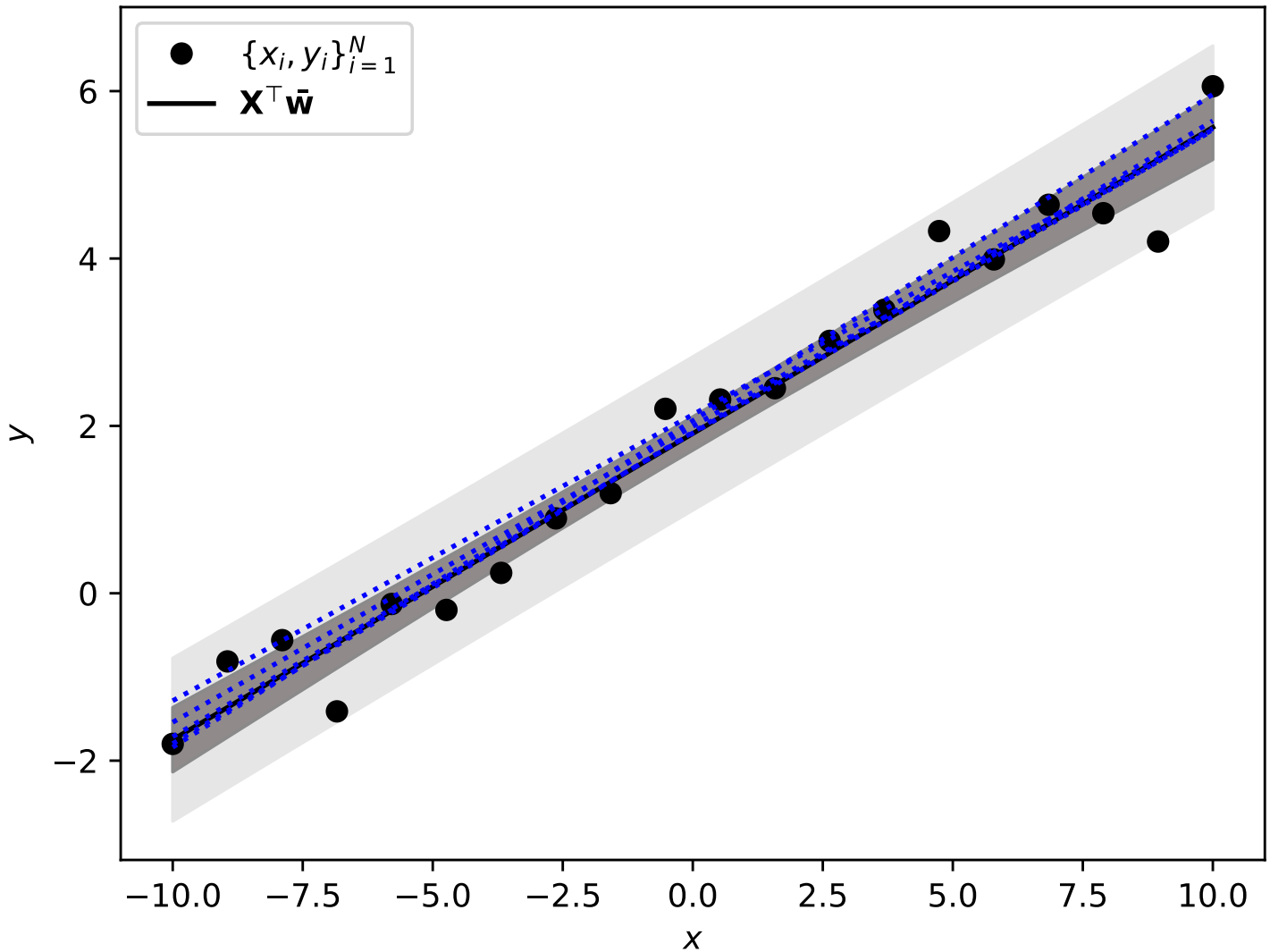
to describe our epistemic uncertainty over parameters θ.



Bayesian Linear Regression: The Predictive Distribution







Bayesian Linear Regression. The dashed lines are samples of $\boldsymbol{w}$ from the posterior.

Thompson's Sampling (Bayesian Bandits)

Bayesian Bandits

Recall: distribution $R_t \sim \nu_{\boldsymbol{\theta}}(\cdot \mid a_t)$.

Bayesians approach: consider prior distribution over parameters,

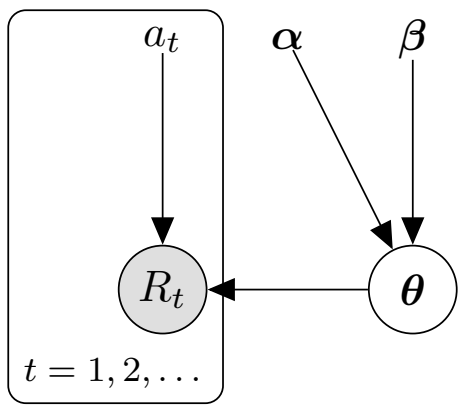
$$\theta_a \sim P(\theta_a)$$

And then work towards a posterior distribution:

$$P(\theta_a|\mathcal{D}) \propto P(\mathcal{D}|\theta_a)P(\theta_a)$$

where data (observations)

$$\mathcal{D} = \{(a_i, r_i)\}_{i=1}^{t}$$

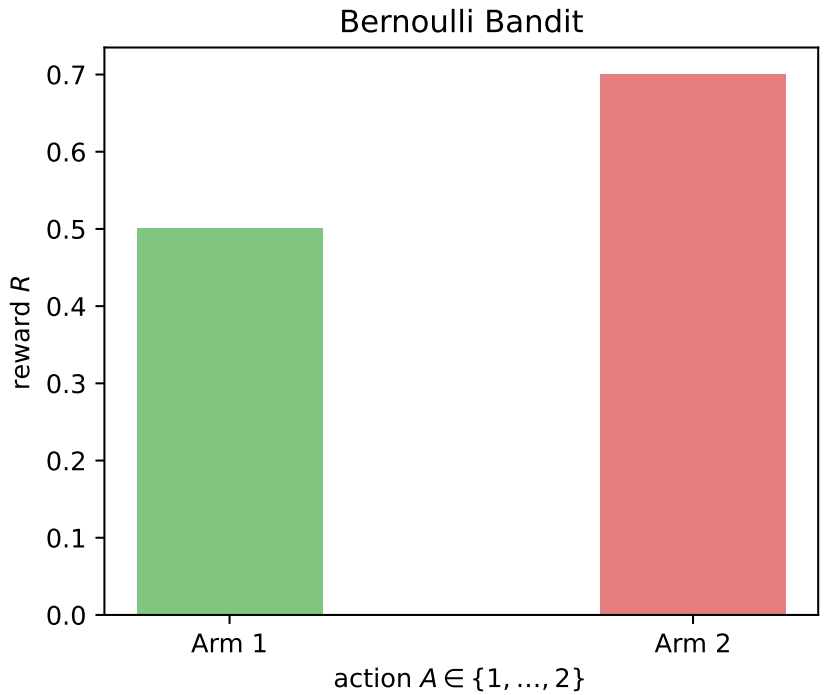


Example: Two Coins

Suppose two unfair coins, such that for coin a:

$$r_t = \begin{cases} 1 & \text{(Heads) with probability } \theta_a \\ 0 & \text{(Tails) with prob. } 1 - \theta_a \end{cases}$$

i.e., a Bernoulli distribution $P(R_t \mid a_t)$.



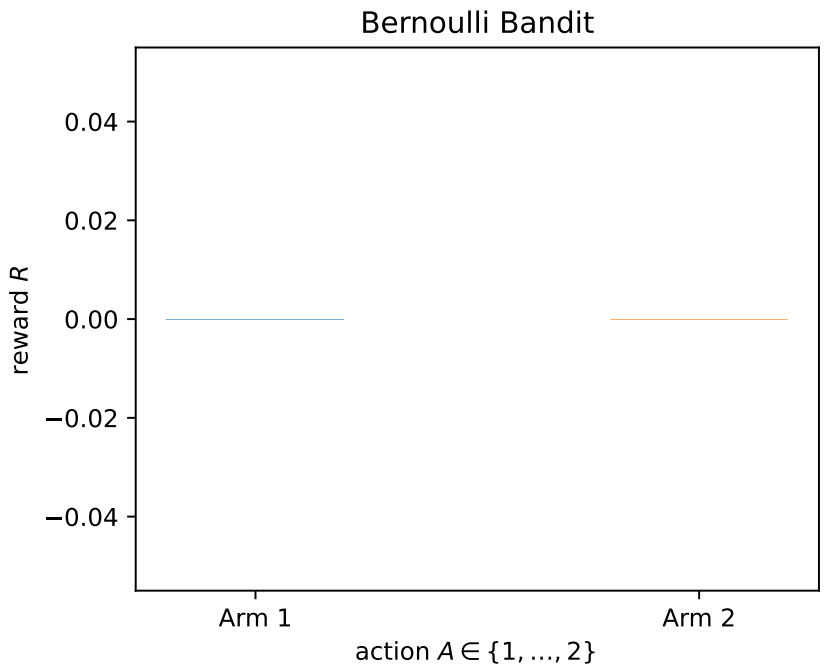
Which coin a_t to flip at time t? $a \in \{1, 2\}$.

Example: Two Coins

Suppose two unfair coins, such that for coin a:

$$r_t = \begin{cases} 1 & \text{(Heads) with probability } \theta_a \\ 0 & \text{(Tails) with prob. } 1 - \theta_a \end{cases}$$

i.e., a Bernoulli distribution $P(R_t \mid a_t)$.



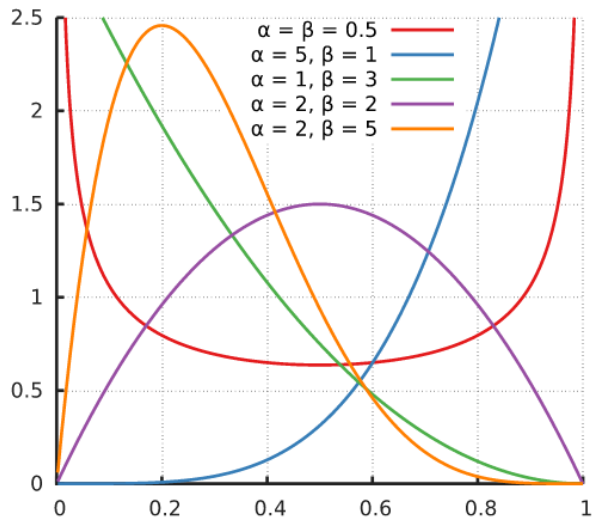
Which coin a_t to flip at time t? $a \in \{1, 2\}$.

Bernoulli and Beta Distributions

What about $P(\theta_a)$? We can use the Beta distribution:

$$\theta_a \sim \text{Beta}(\theta_a|\alpha,\beta) \propto \theta_a^{\alpha-1}(1-\theta_a)^{\beta-1}$$

which provides some $\theta_a \in [0,1]$.



$P(\theta) \equiv \text{Beta}(\theta|\alpha,\beta)$ (Image source: Wikipedia)

Thus, we need $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2]$. But what values for these?

Recall, we want posterior

$$P(\theta_a|\mathcal{D}) \propto P(\mathcal{D}|\theta_a)P(\theta_a)$$

(for any arm a): the data likelihood multiplied by the prior.

The Beta distribution is a conjugate prior for (same parametric representation) as the Bernoulli.

Main idea here: we can update the posterior in a neat way.

We take action a, observe $r_t \sim P_{\boldsymbol{\theta}}(R \mid a)$, then

$$\begin{aligned} P(\theta \mid R_t = 0) &\propto P(\theta)P(R_t = 0|\theta) && \triangleright \text{ We observed Tails} \\ &= \text{Beta}(\theta \mid \alpha + 1, \beta) \\ P(\theta \mid R_t = 1) &\propto P(\theta)P(R_t = 1|\theta) && \triangleright \text{ We observed Heads} \\ &= \text{Beta}(\theta \mid \alpha, \beta + 1) \end{aligned}$$

Thompson Sampling

We may think of environment (bandit machine) parametrised by θ.

1. Sample a bandit instance (create an 'environment')

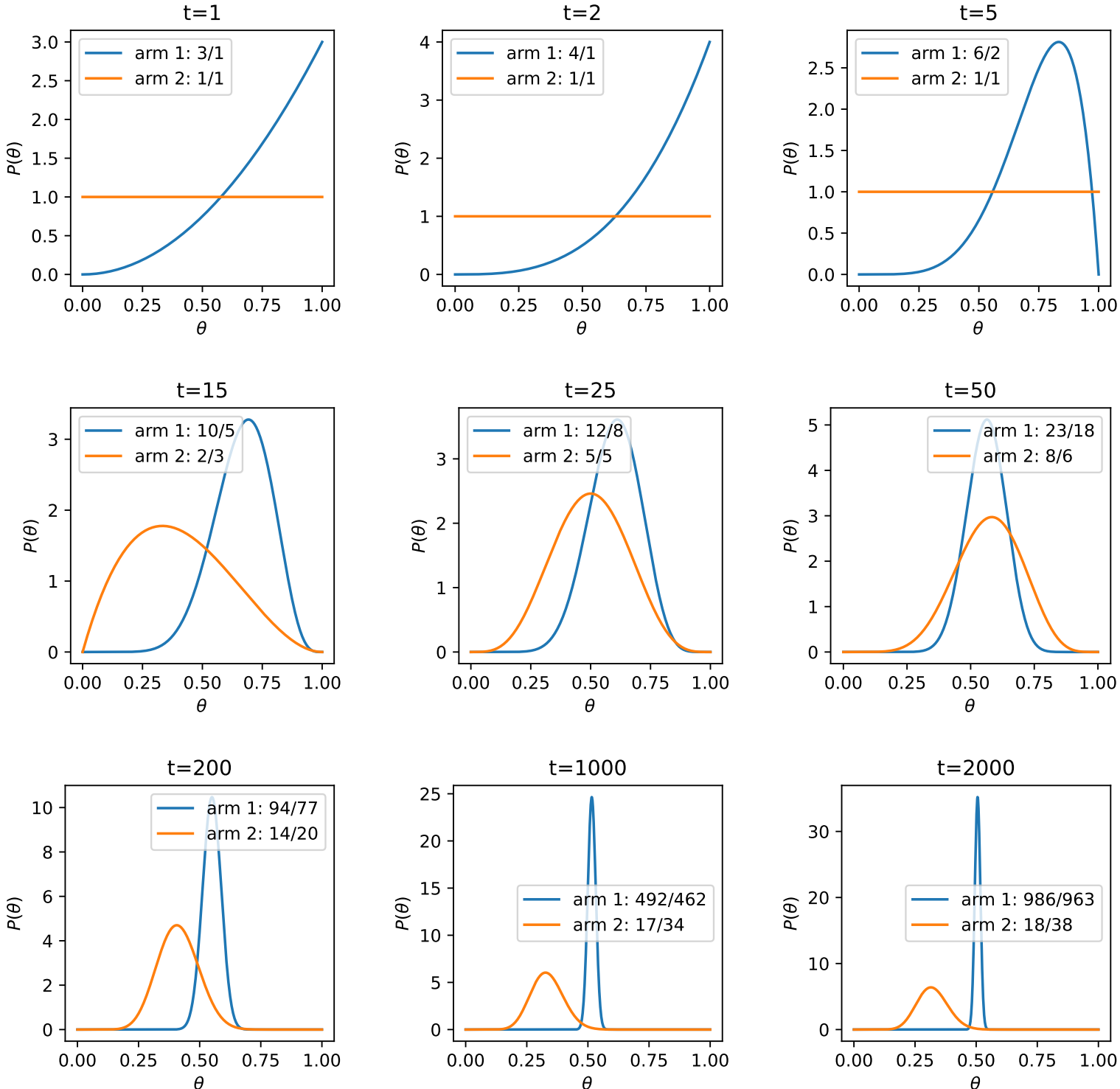
$$\theta_a \sim P(\theta_a|\mathcal{D})$$

2. Act optimally according to *this* bandit instance

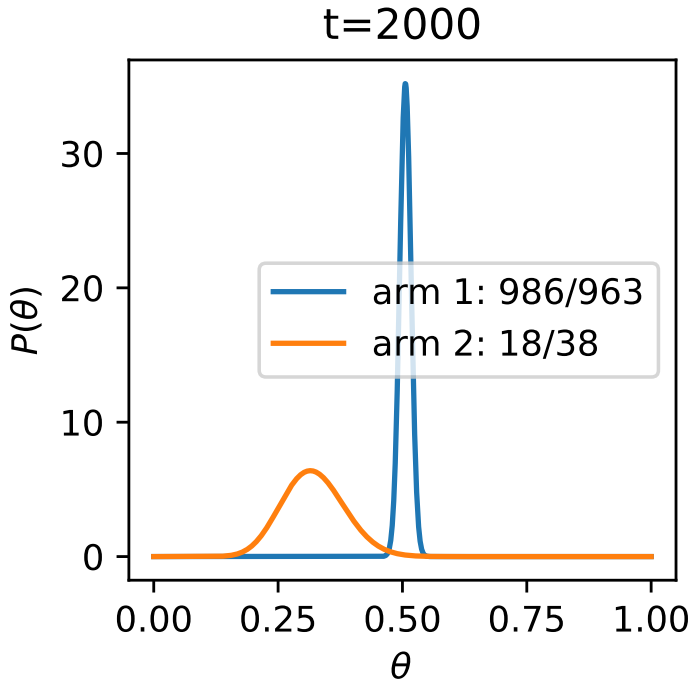
$$a_t = \underset{a}{\text{argmax}}\, Q_t(a)$$

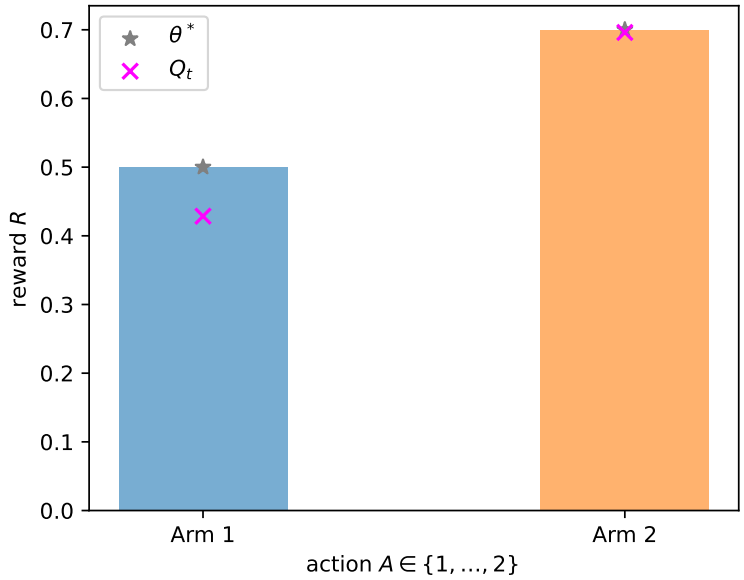
where $Q_t(a) = \mathbb{E}[R_t|A_t = a] = \theta_a$.

3. Update beliefs (with new knowledge (a_t, r_t)).


t=1
arm 1: 3/1
arm 2: 1/1
t=2
arm 1: 4/1
arm 2: 1/1
t=5
arm 1: 6/2
arm 2: 1/1
t=15
arm 1: 10/5
arm 2: 2/3
t=25
arm 1: 12/8
arm 2: 5/5
t=50
arm 1: 23/18
arm 2: 8/6
t=200
arm 1: 94/77
arm 2: 14/20
t=1000
arm 1: 492/462
arm 2: 17/34
t=2000
arm 1: 986/963
arm 2: 18/38
P(θ)
θ

Thompson Sampling,





- Simple to implement (just requires sampling)
- Theoretically near optimal
- Works better than UCB in practice
- But, may over-explore

EXP3

EXP3 (Exponential-weight Exploration and Exploitation)

EXP3 is is designed specifically for adversarial settings (rewards controlled by an adversary – more on this Week 9!) and when the reward distributions are unknown or *changing*.

Initialize cumulative reward estimates $\widehat{S}_{0,a} = 0$ for all arms a; then, for $t = 1, \ldots, T$:

1. sample $A_t \sim P_t(a)$ where

$$P_t(a) = \frac{\exp(\eta \widehat{S}_{t-1,a})}{\sum_a \exp(\eta \widehat{S}_{t-1,a})}$$

2. update (where, we have received reward R_t)

$$\widehat{S}_{t,a} = \widehat{S}_{t-1,a} + \left(1 - \frac{[\![A_t = a]\!] \cdot (1 - R_t)}{P_t(a)}\right)$$

for learning rate η (assume does not depend on t).

Notice that the loss-based importance-weighted estimator.
Note: We have assumed that $R_t \in [0, 1]$.

Summary

Summary (So Far)

Bandits . . .

. . . imply sequential decision making: a decision we make now indirectly affects our loss (regret) in the future (if we try to learn from our mistakes – which we should).

. . . could be also be called 'reinforcement learning from the same state' or '. . . with no influence over the state'

. . . have many real-world applications.

. . . involve the exploration vs exploitation tradeoff.

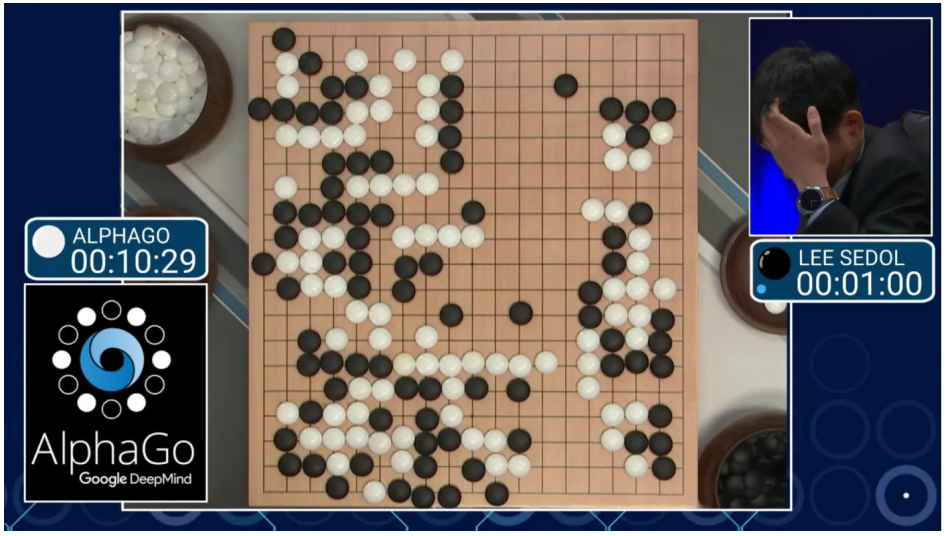
There are many applications of bandits. Many theoretical results.

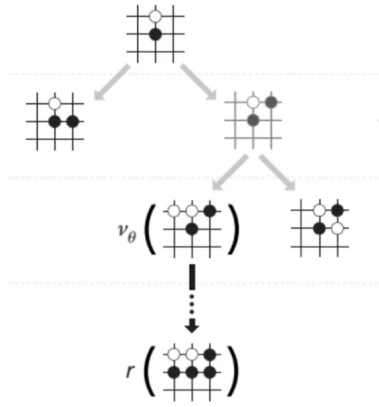
But what's missing?

An autonomous agent should be able to observe *and change* its environment, i.e., $\Rightarrow$ reinforcement learning (coming weeks).

Monte Carlo Tree Search (UCT)

Case study: Go. **What's your next move?** (action a)





A search tree!

Previously (last week):

1. Monte Carlo (or, some heuristic) search to leaves
2. identify one with the biggest path (final) reward

But (among other problems): stochastic (because of adversary), and tree is just too deep. We need $\approx Q(a) = \mathbb{E}[\sum_t r_t]$.

Solution: **Use a bandit algorithm to perform selection**.

Guides search towards promising areas of the search space, exploration of the search tree.

Bandit Machines

CSC_52081_EP Advanced Machine Learning and Autonomous Agents

Jesse Read