Microsoft

“Bandits”

ADITH SWAMINATHAN

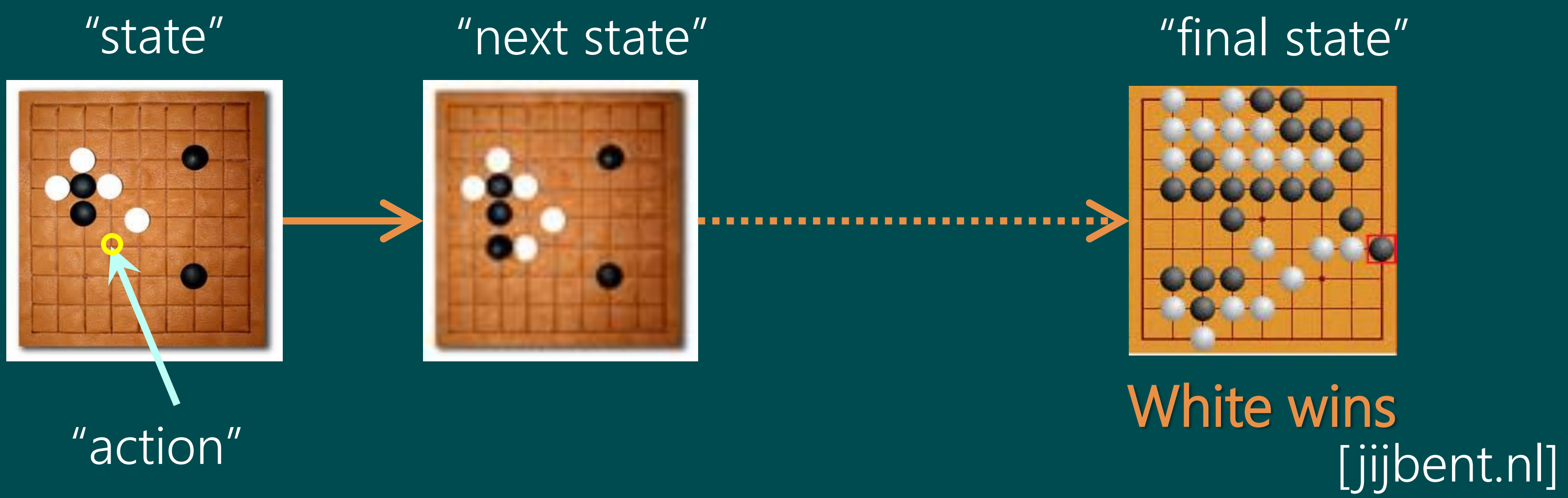
Researcher, MSR AI

Contents

To solve reinforcement learning,

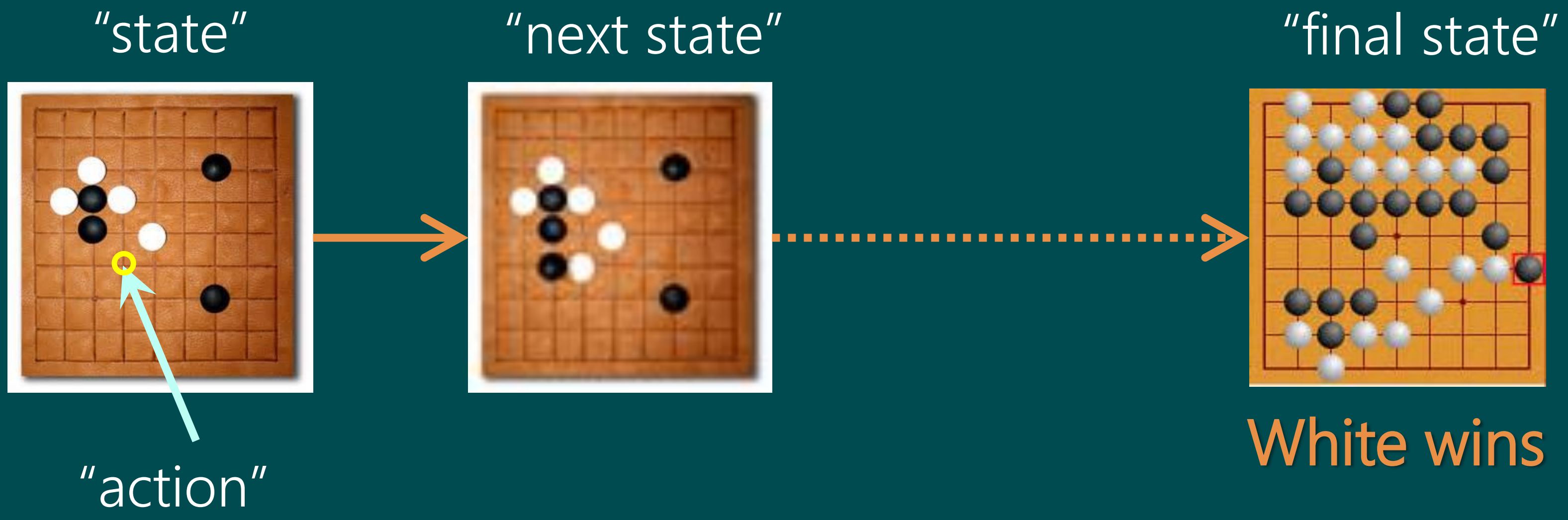
We must overcome 4 fundamental challenges:

- Representation
- Generalization
- Temporal Credit Assignment
- Exploration



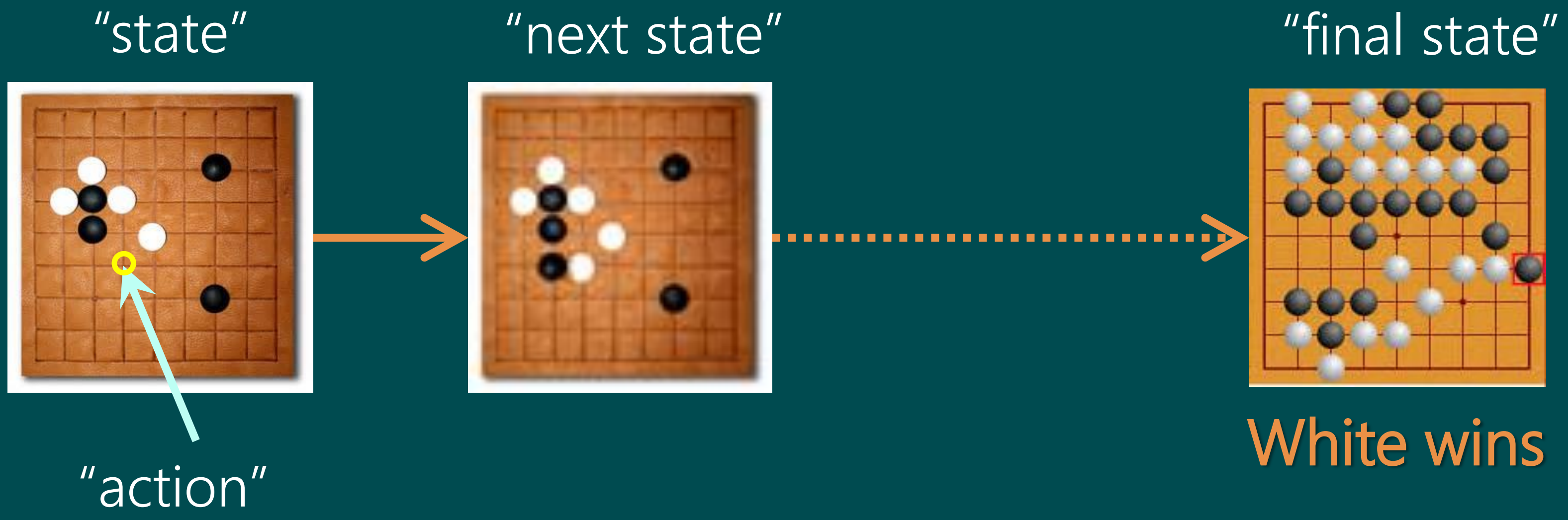
Representation

"The key to artificial intelligence has always been the representation" – Jeff Hawkins



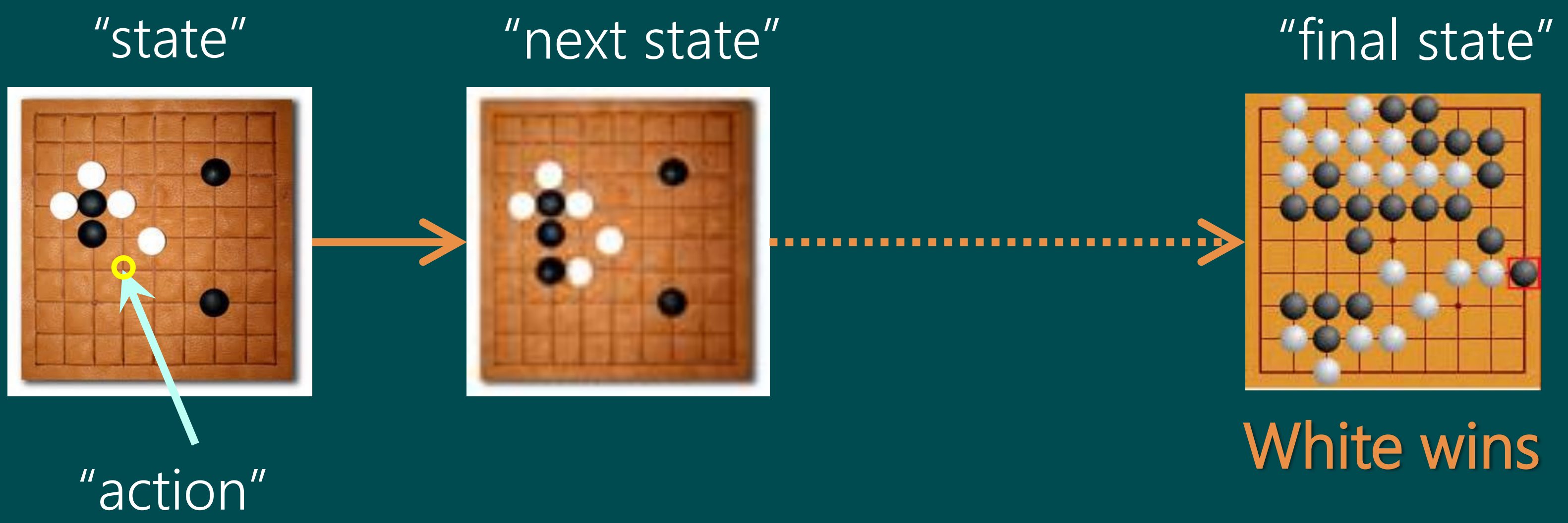
Generalization

The ability to behave well in hitherto unseen states.



Temporal Credit Assignment

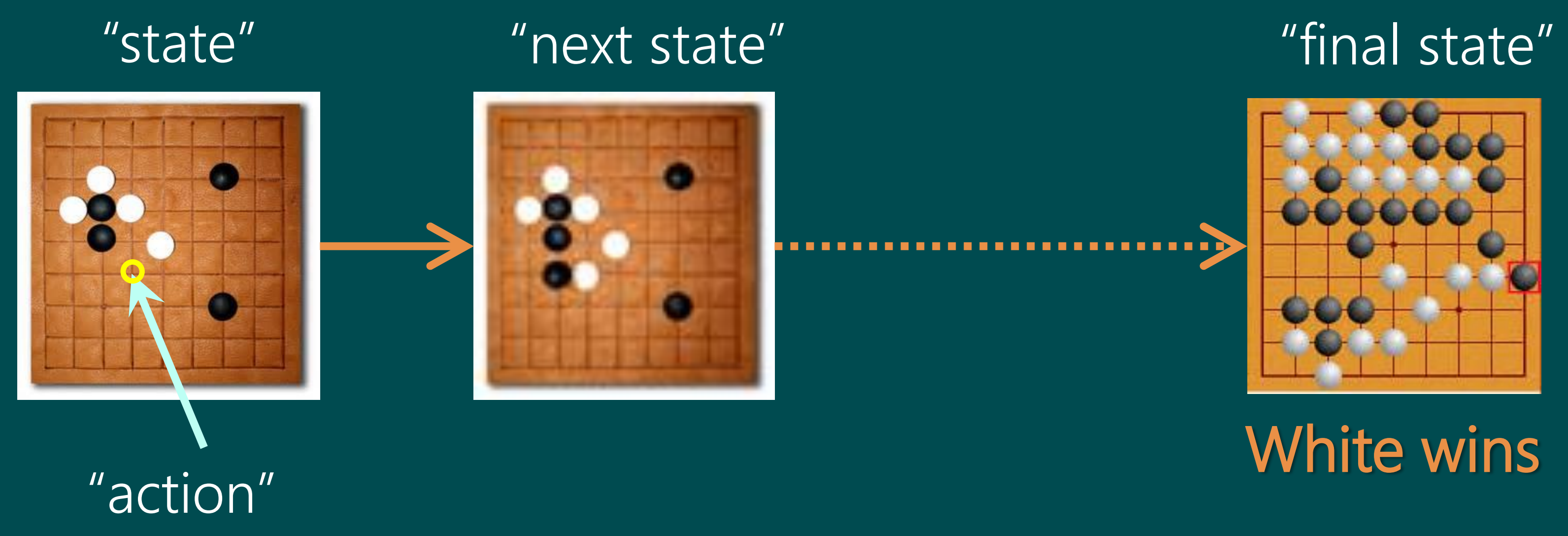
Which of the actions was salient for the eventual observed outcome?



Exploration

Is there an action we have not yet tried that could lead to an overall better outcome?

<u>This module</u>: "Bandits" – study exploration in isolation



Contents

The Multi-Armed Bandit Framework

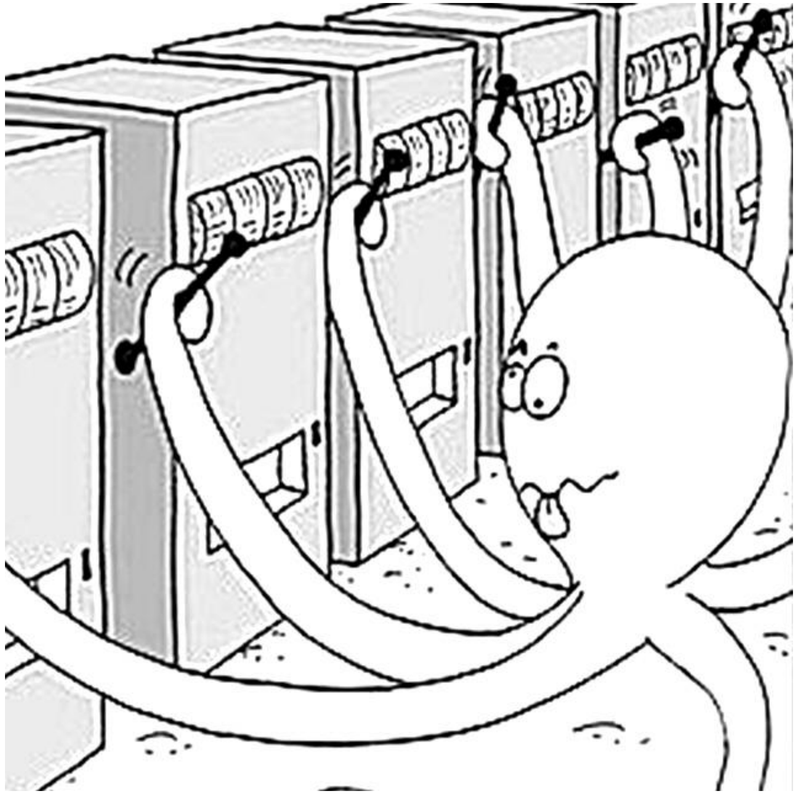
Notation

- A set of k actions ("arms") $A = \{a_1, a_2, \dots, a_k\}$
- Reward $R_a = \Pr(r|a)$ is unknown

- At each step $t = 1, 2, \dots, T$:

1. Choose $a_t \in A$
2. Receive $r_t \sim \Pr(r|a_t)$

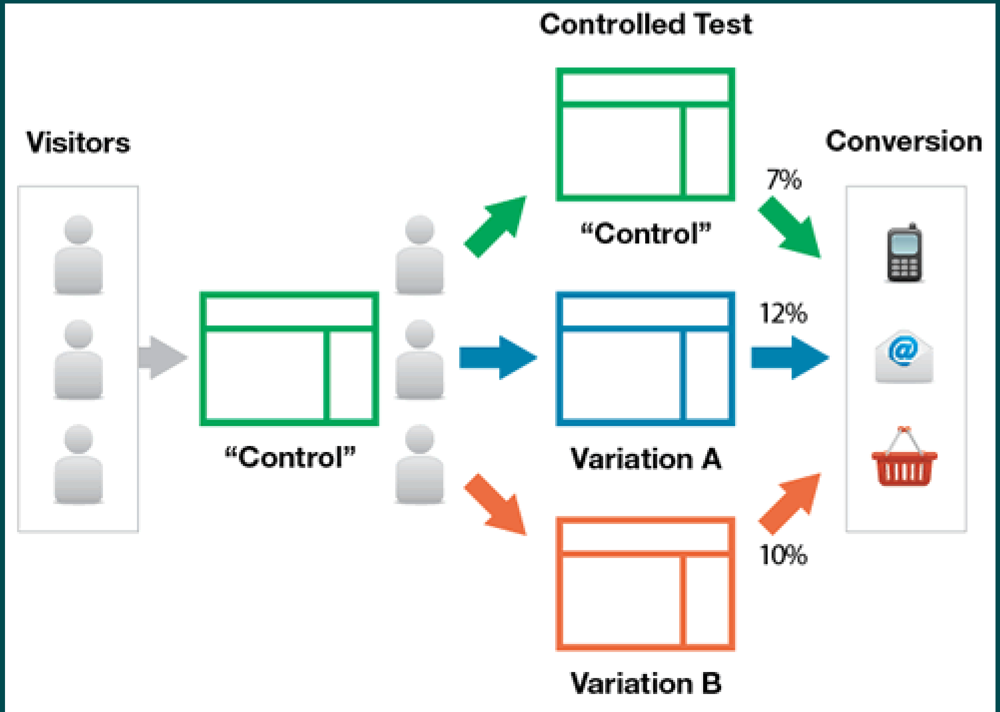
- Can you build an agent to maximize $\sum_t r_t$?



[research.microsoft.com]

Motivating Applications

Evaluating user-facing systems



[widerfunnel.com]

A/B/n controlled testing

- Many candidate variations to try
- Quickly find best candidate

- Option 1: A/B/n test
  (Randomized Controlled Trial)

- Option 2: Bandit algorithm
  http://aka.ms/mwt/

Motivating Applications

Demo: Drug Discovery


[istockphoto.com]

http://iosband.github.io/2015/07/28/Beat-the-bandit.html

- A: Set of experimental drugs
- $Reward = \{0: \text{die}, 1: \text{live}\}$
- T patients

- Save as many as you can!

- Could we have saved more?
- Can we write an ***optimal*** algorithm?

Warm-up: The Naïve Algorithm

A/B/n Testing

- Assign $\frac{T}{k}$ patients to each action
- Implement: e.g. round-robin through available actions

Exercise 0: Implement the round-robin algorithm

Can we do better to maximize $\sum_t r_t$?

Sequential Decision w/ Incomplete Info

Exploration-Exploitation Dilemma

- Exploration: Gather information
- Exploitation: Optimal decision using current information

Fundamental trade-off between exploration and exploitation!

Algorithm 1: The Greedy Algorithm

Consider algorithms that estimate $\hat{r}_a \approx \mathbb{E}\left[r|a\right]$

$$n_a = \sum_{\{t:a_t=a\}} 1 \;\; ; \;\; \hat{r}_a = \sum_{\{t:a_t=a\}} r_t/n_a$$

- Pick the action with the highest estimate

$$a_t = \operatorname*{argmax}_{a\in A} \hat{r}_a$$

<u>Problem:</u> Greedy can lock-on to sub-optimal action forever

Algorithm 1.5: Greedy Variants

Consider algorithms that estimate $\hat{r}_a \approx \mathbb{E}\,[r|a]$

- Optimistic-Greedy: Initialize $\hat{r}_a$ to a large initial value, R
- ➢ Then play Greedy algorithm

- ϵ-Greedy:
- ➢ With probability ϵ, pick a uniformly random action
- ➢ With probability $1-\epsilon$, play Greedy algorithm

<u>Question:</u> How should we set R or ϵ?

<u>Exercise 1:</u> Empirically try different ways to set hyper-parameters

Further Reading

- Chapter 2; Reinforcement Learning: An Introduction, Sutton and Barto

http://ufal.mff.cuni.cz/~straka/courses/npfl114/2016/sutton-bookdraft2016sep.pdf

- Preliminary book, http://slivkins.com/work/MAB-book.pdf

- Platform: http://aka.ms/mwt/
- Demo: http://iosband.github.io/2015/07/28/Beat-the-bandit.html

Contents

Regret

Thought experiment to quantify "price of information"

- Suppose we know all reward distributions $\mathrm{Pr}(r|a)$
- Optimal policy is to always play $a^* = \max_{a \in A} \boldsymbol{E}[r|a]$

Regret: $L_T = T\boldsymbol{E}[r|a^*] - \sum_t \boldsymbol{E}[r|a_t]$

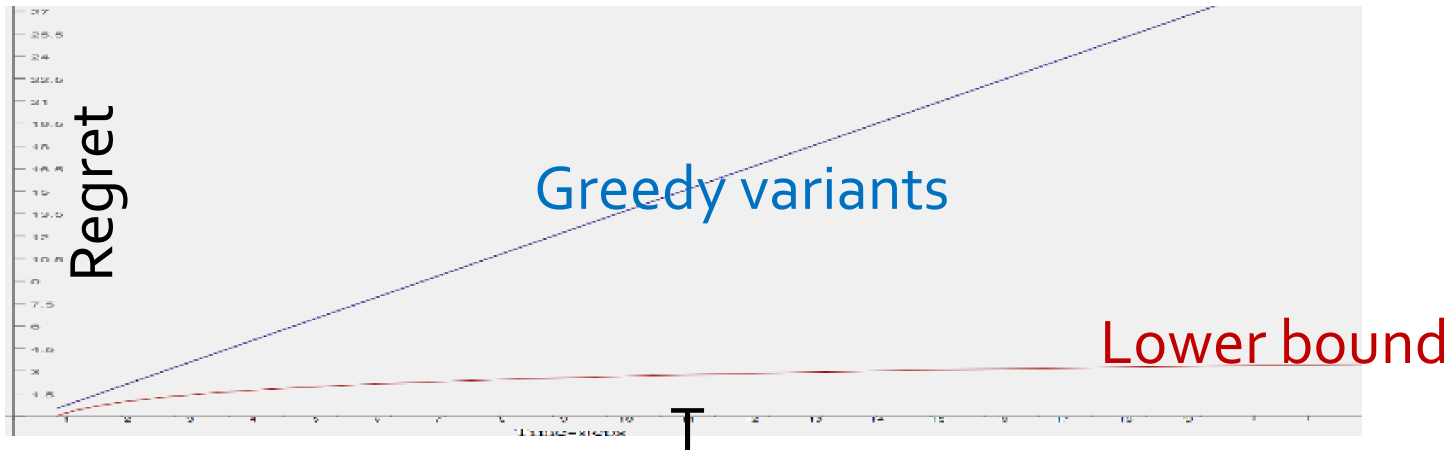
Maximize $\sum_t r_t$ ≡ Minimize regret L_T

No-Regret Strategies

Exercise: Prove that Greedy variants have linear regret

- *Greedy* and ***ϵ****-Greedy* have linear regret $L_T \geq Const \cdot T$
- No matter the algorithm, lower bound on regret is [Lai and Robbins]

$$\lim_{T\to\infty} L_T \geq Const' \cdot \log T$$



Can we write an algorithm with $L_T \leq Const'' \cdot \log T$?

Greedy variants have linear regret

Regret: $L_T = T\boldsymbol{E}[r|a^*] - \sum_t \boldsymbol{E}[r|a_t]$

Insight 1: Greedy exploits too much! $\Rightarrow \Pr(a_t \neq a^*) \geq c$

Insight 2: ϵ-Greedy explores too much! $\Pr(a_t \neq a^*) \geq c'$

Regret Minimization Principle

Optimism in the face of uncertainty

To achieve low regret, we only need to identify an optimal arm!

- Good algorithm should not play sub-optimal arms too often…

- So:
  - ➢ Use collected data to eliminate arms that "very likely" are sub-optimal
  - ➢ Choose the most optimistic remaining option

Upper Confidence Bound Algorithm

[Auer et al]

1. [Initialization] For each arm $\boldsymbol{a}$, maintain $\boldsymbol{n_a}$ and $\hat{\boldsymbol{r}}_{\boldsymbol{a}}$
2. [Initialization] For first $\boldsymbol{k}$ rounds, play each arm once.

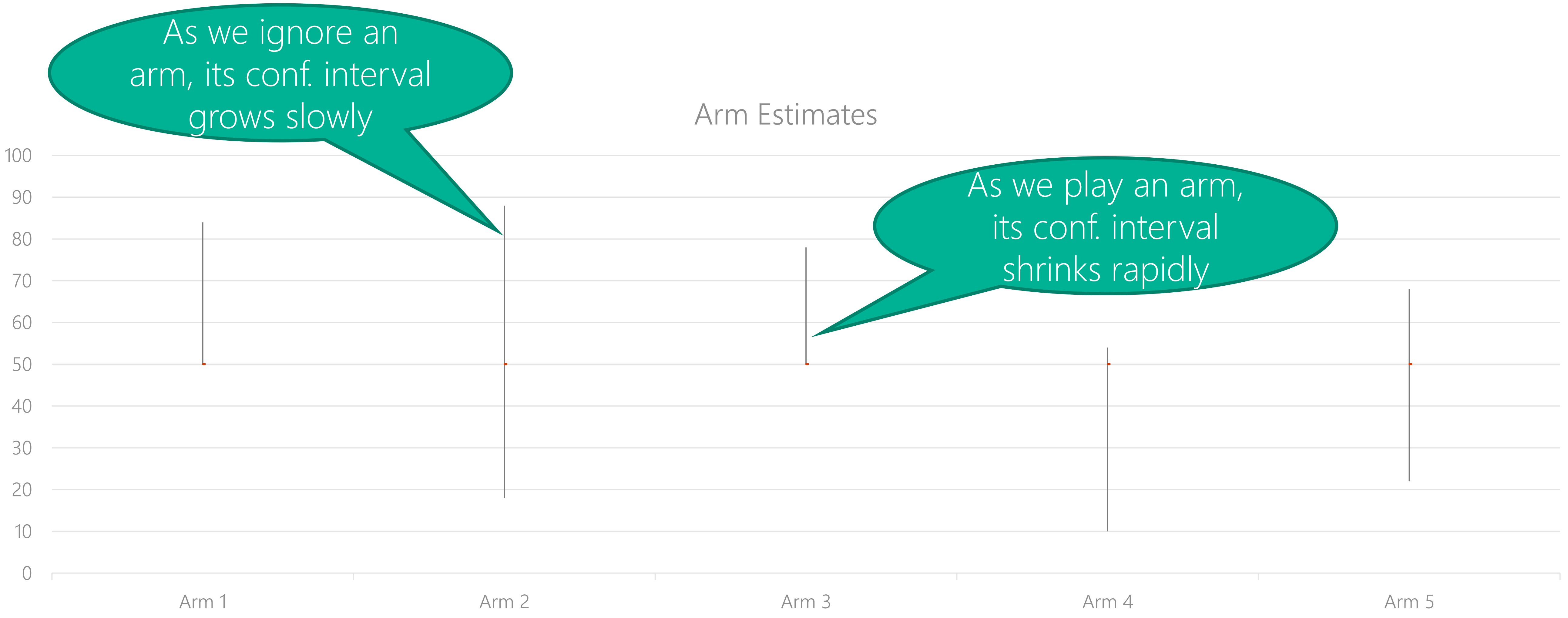
3. At round t, play $\mathrm{a_t} = \underset{a \in A}{\mathrm{argmax}} \left\{ \hat{r}_a + \sqrt{\frac{2 \log t}{n_a}} \right\}$

UCB1 achieves* logarithmic regret $\boldsymbol{L_T \leq Const'' \cdot \log T}$

Gentle proof sketch: https://jeremykun.com/2013/10/28/optimism-in-the-face-of-uncertainty-the-ucb1-algorithm/

<u>Exercise 3:</u> Implement the UCB1 algorithm

UCB Illustration



Regret Minimization Principle

Posterior Sampling

- Suppose we have prior on $\Pr(r|a)$
- ➢ Bayesian MAB

- Idea: Choose arm a according to probability that a is optimal

- This probability can be hard to compute…
- ➢ So: Sample!

Posterior Sampling Example

Beta-Bernoulli Example for Drug Discovery

- Drug discovery example
- ➢ Rewards are $Bernoulli(p_a)$
- ➢ Assume $p_a \sim Beta(1,1)$ [Prior]
- ➢ Posterior of Beta-Bernoulli is also Beta! [Conjugate families]

- For each arm a, maintain $\#live_a, \#die_a$
- At round t,
- ➢ $\hat{p}_a \sim Beta(1 + \#live_a, 1 + \#die_a)$
- ➢ Play $a_t = \underset{a \in A}{argmax}\, \hat{p}_a$

Posterior sampling also achieves* logarithmic regret bound

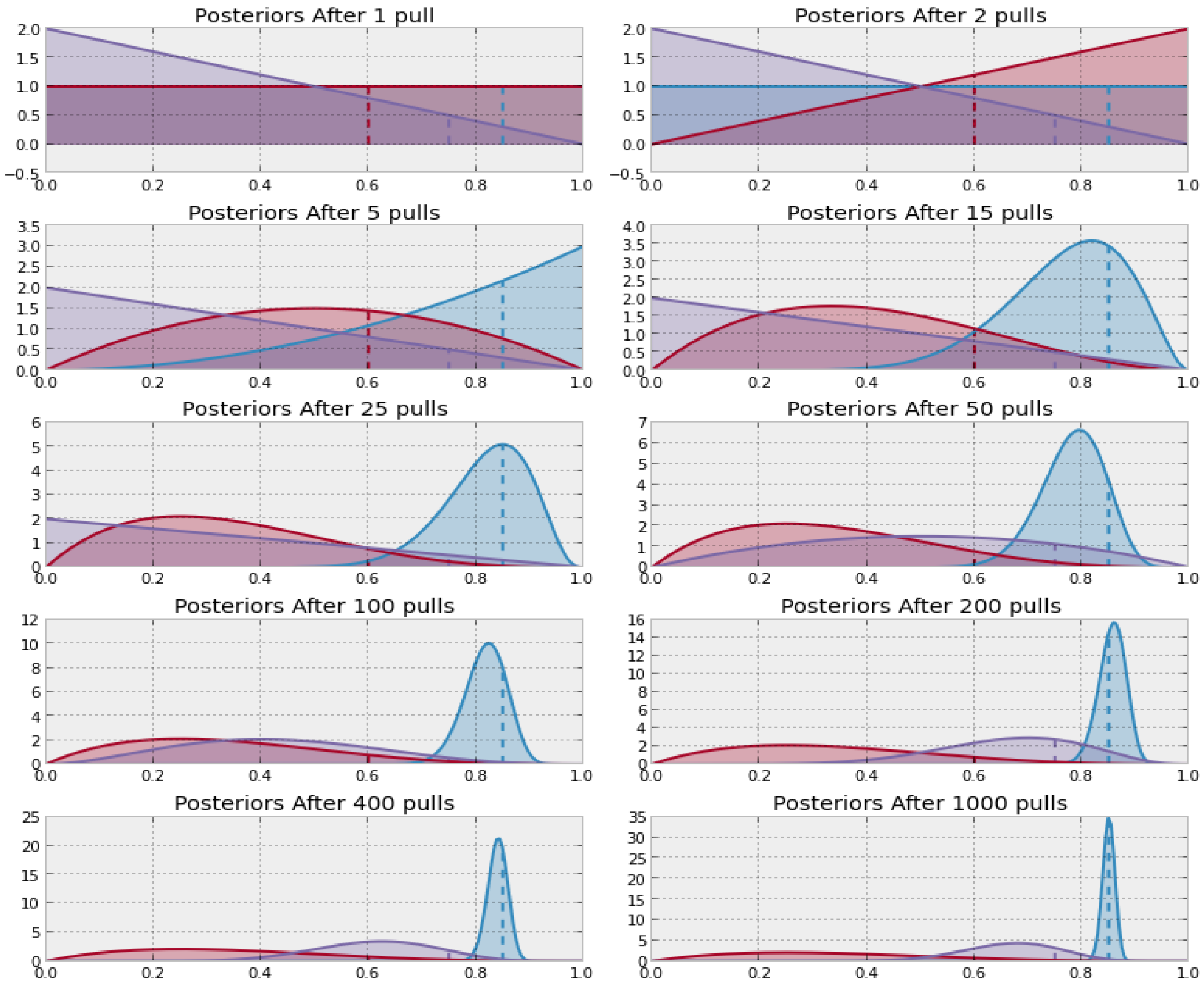
<u>Exercise 4:</u> Implement the Beta-Bernoulli Posterior Sampling algorithm

Posterior Sampling Illustration

Beta-Bernoulli Example for Drug Discovery

- Drug discovery example
- Rewards are $Bernoulli(p_a)$
- Assume $p_a \sim Beta(1,1)$
- Suppose $k = 3$

https://dataorigami.net/blogs/napkin-folding/79031811-multi-armed-bandits



[dataorigami.net]

<u>Optional Exercise:</u> Implement the Posterior Sampling algorithm for a Gaussian prior

Further Reading

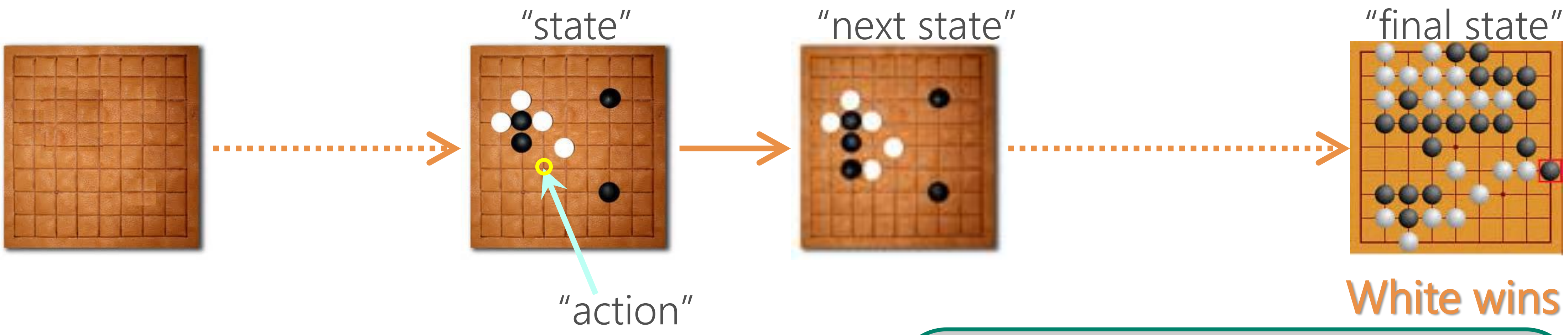
- UCB: https://jeremykun.com/2013/10/28/optimism-in-the-face-of-uncertainty-the-ucb1-algorithm/
- Thompson sampling: https://dataorigami.net/blogs/napkin-folding/79031811-multi-armed-bandits
- Finite-time Analysis of the Multi-armed Bandit Problem, Auer et al http://dl.acm.org/citation.cfm?id=599677
- An Empirical Evaluation of Thompson Sampling, Chapelle and Li https://papers.nips.cc/paper/4321-an-empirical-evaluation-of-thompson-sampling
- Tutorial, Dave Silver http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching_files/XX.pdf

Contents

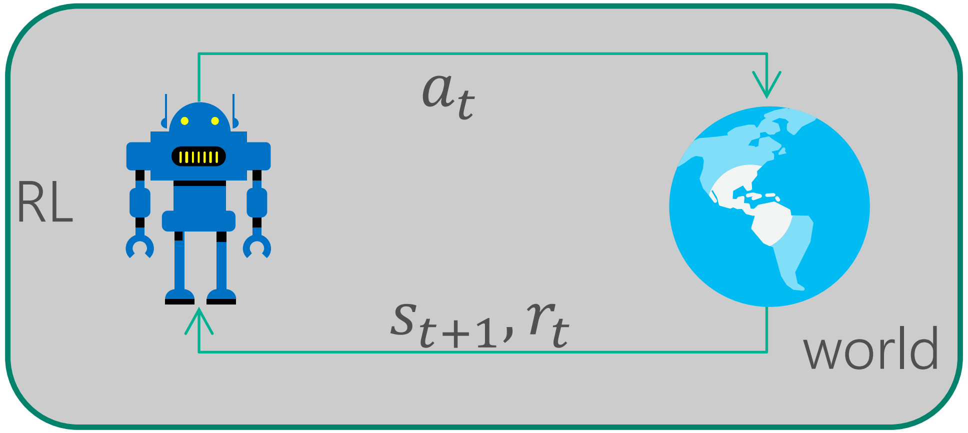
Recap: RL

Challenges: Representation, Generalization, Exploration, Temporal Credit Assignment

Agent plays white

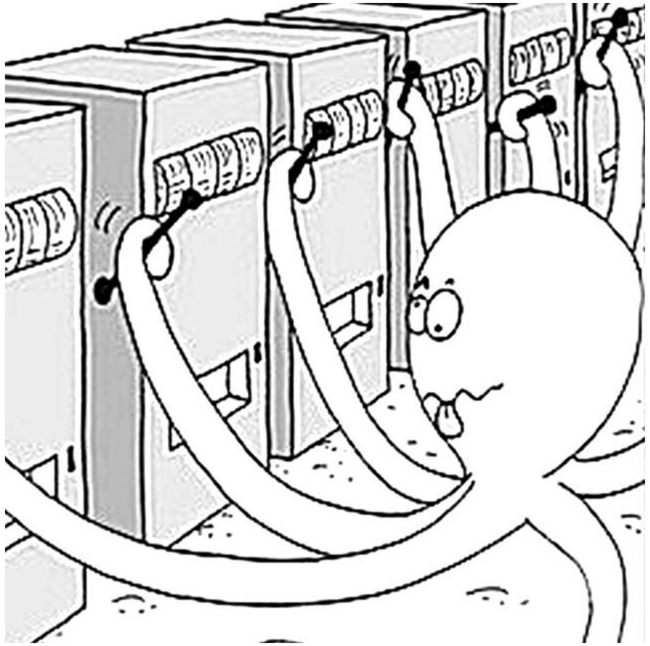


- State s_t depends on previous actions
- Typically delayed reward



Recap: Multi-Armed Bandit

Challenges: Exploration



[research.microsoft.com]

- Exploration-exploitation dilemma

Now: Contextual Bandits

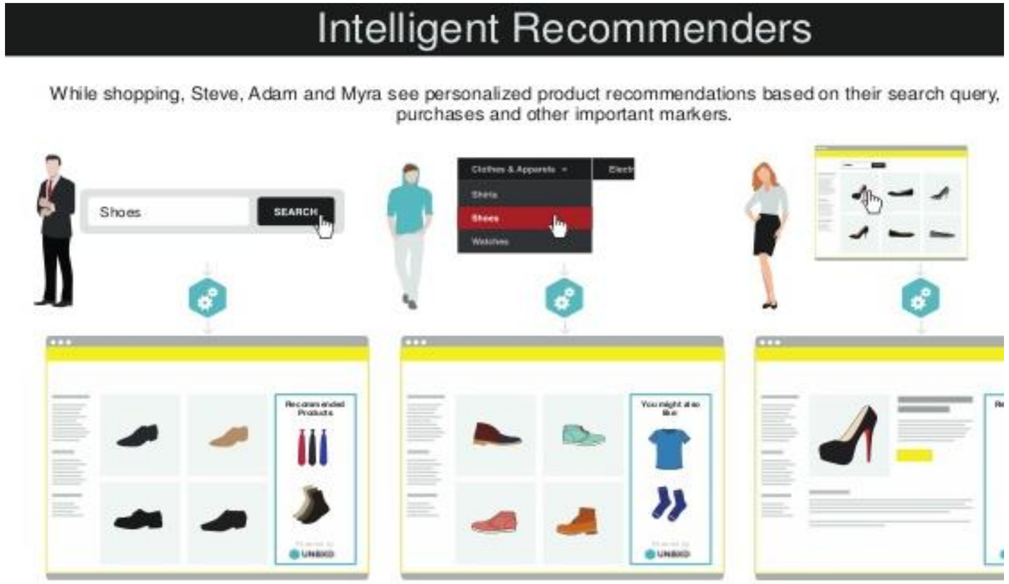
Challenges: Representation, Generalization, Exploration

- A set of k arms $A = \{a_1, \dots a_k\}$
- At each turn t:

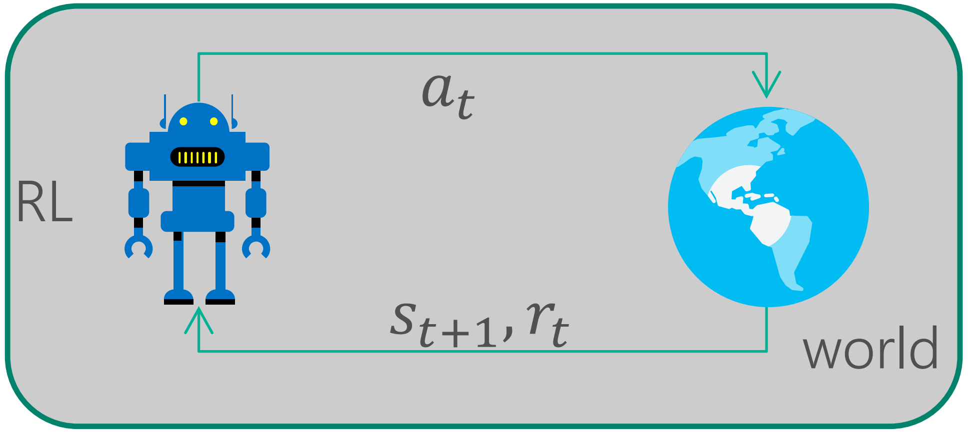
➢ Receive context $s_t \sim \Pr(S)$ [unknown distribution]

➢ Play action a_t

➢ Receive reward $r_t \sim \Pr(r | s_t, a_t)$ [unknown distribution]



[unbxd.com]



A Contextual Bandit Algorithm

LinUCB [Li et al]

- Assume linear relation between rewards and arms
  - Arms have an embedding $x_{t,a} = \phi(s_t, a)$
  - Reward $\mathbb{E}[r_{t,a} \mid x_{t,a}] = \theta_a \cdot x_{t,a}$

- Idea: Use ridge regression for $\hat{\theta}_a$ using $\left(A: \sum x_{t,a} x_{t,a}^T, b: \sum r_{t,a} \cdot x_{t,a}\right)$
- $\hat{\theta}_a = A^{-1} b$
- Add exploration bonus

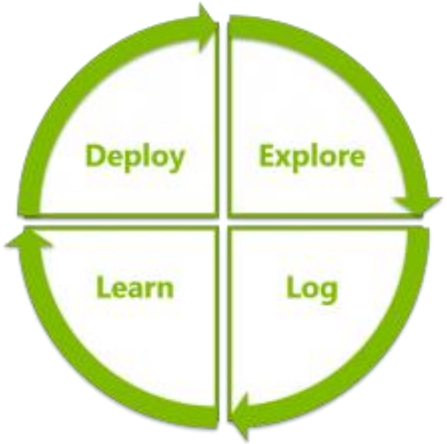
$$\underset{a}{argmax} \quad \underbrace{\hat{\theta}_a \cdot x_{t,a}}_{\text{Mean}} + \alpha \underbrace{\sqrt{x_{t,a}^T A^{-1} x_{t,a}}}_{\text{Conf.Interval}}$$

Case Studies

News Recommendation

- MSN piloted contextual bandits
- Update model every 5 mins
- ϵ-Greedy strategy for exploration ; no tuning

➢ 25% increase in clicks over static baseline!



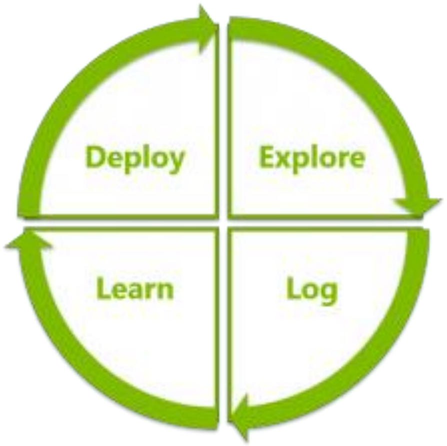
http://aka.ms/mwt

- Yahoo Front Page news recommendation

➢ LinUCB gave ~10% increase in clicks compared to baseline ϵ-Greedy

Further Reading

- Decision Service http://ds.microsoft.com



http://aka.ms/mwt

- Tutorial: http://hunch.net/~exploration_learning/
- http://www.stat.berkeley.edu/~bartlett/courses/2014fall-cs294stat260/readings.html
- LinUCB: https://arxiv.org/pdf/1003.0146.pdf