facebook
Artificial Intelligence Research

How to explore an MDP efficiently: Exploration-Exploitation Dilemma in Bandits

Pirotta Matteo
Facebook AI Research

Acknowledgments

Special thanks to Alessandro Lazaric for providing these slides from the RL class we teach in Paris.

Sequential resource allocation

Clinical trials

- K treatment for a given symptom (with unknown effect)
- What treatment should be allocated to the next patient based on responses observed on previous patients ?

Online advertisement

- K adds that can be displayed
- Which add should be displayed for a user, based on the previous clicks of previous (similar) users ?

Stochastic Multi-Armed Bandit

At each round $t \in \{1, \ldots, n\}$, the learning agent

- chooses an arm a_t
- receives a reward $r_t \sim \nu_{a_t}$

Goal: maximize $\mathbb{E}\left[\sum_{t=1}^{n} r_t\right]$

A Simple Recommendation System

- A RS can recommend different genres of movies (e.g., action, adventure, romance, animation)
- Users arrive at random and *no information about the user is available*
- The RS picks a genre to recommend to the user but not the specific movies
- The feedback is whether the user *watched* a movie of the recommended genre or not
- *Objective:* design a RS that maximizes that movies watched in the recommended genre

RS as a Multi-armed Bandit

For $t = 1, \ldots, n$

1. User arrives
2. **Recommend genre** a_t
3. **Reward**

$$r_t = \begin{cases} 1 & \text{user watches movie of genre } a_t \\ 0 & \text{otherwise} \end{cases}$$

EndFor

RS as Multi-armed Bandit

The *model*

- $\nu(a)$ is a Bernoulli
- $\mu(a) = \mathbb{E}\big[r(a)\big]$ is the probability a *random* user watches a movie of genre a
- **Assumption:** $r_t \sim \nu(a_t)$ is a realization of the Bernoulli of genre a

The *objective*

- Maximize sum of reward $\mathbb{E}\Big[\sum_{t=1}^{n} r_t\Big]$

Other Examples

- Packet routing
- Clinical trials
- Web advertising
- Computer games
- Resource mining
- ...

Outline

1. Performance of a bandit algorithm

The Regret

$$R_n = \max_a \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a)\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a_t)\Big]$$

The Regret

$$R_n = \max_a \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a)\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a_t)\Big]$$

The expectation summarizes any possible source of randomness (either in r or in the algorithm)

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \max_a \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a)\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a_t)\Big]$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \max_a n\mu(a) - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a_t)\Big]$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \max_a n\mu(a) - \sum_a \mathbb{E}[T_n(a)]\mu(a)$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = n\mu(a^*) - \sum_{i=1}^{K} \mathbb{E}[T_n(a)]\mu(a)$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \sum_{a \neq a^*} \mathbb{E}[T_n(a)](\mu(a^*) - \mu(a))$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \sum_{a \neq a^*} \mathbb{E}[T_n(a)]\Delta(a)$$

The Regret

- Number of times action a has been selected after n rounds

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Regret

$$R_n = \sum_{a \neq a^*} \mathbb{E}[T_n(a)] \Delta(a)$$

- Gap $\Delta(a) = \mu(a^*) - \mu(a)$

The Regret

$$R_n = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}]\Delta_i$$

$\Rightarrow$ we only need to study the *expected number of times suboptimal* actions are selected

$\Rightarrow$ a good algorithm has $R_n = o(n)$, i.e., $R_n/n \to 0$

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly selecting the best action

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly selecting the best action

Challenge: The learner should solve two opposite problems!

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions $\Rightarrow$ **exploration**

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly selecting the best action
Challenge: The learner should solve two opposite problems!

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions $\Rightarrow$ **exploration**

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly selecting the best action $\Rightarrow$ **exploitation**
Challenge: The learner should solve two opposite problems!

The Exploration–Exploitation Dilemma

Problem 1: The environment **does not** reveal the reward of the actions not selected by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly selecting all actions $\Rightarrow$ **exploration**

Problem 2: Whenever the learner selects a **bad action**, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly selecting the best action $\Rightarrow$ **exploitation**
Challenge: The learner should solve the *exploration-exploitation* dilemma!

Explore-Then-Commit: Algorithm

Explore phase

- **For** $t = 1, \dots, \tau$
  1. **Take action** $a_t \sim \mathcal{U}(A)$ (or round robin)
  2. Observe reward $r_t \sim \nu(a_t)$
- **EndFor**
- Compute statistics for each action a

$$\widehat{\mu}_\tau(a) = \frac{1}{T_\tau(a)} \sum_{s=1}^{\tau} r_s \mathbb{I}\{a_s = a\}$$

Exploit phase

- **For** $t = 1, \dots, \tau$
  1. **Take action** $\widehat{a}^* = \arg\max_a \widehat{\mu}_\tau(a)$
  2. Observe reward $r_t \sim \nu(\widehat{a}^*)$
- **EndFor**

Explore-Then-Commit: Regret

Theorem

If explore-then-commit is run with parameter τ for n steps then it suffers a regret

$$R_n \leq \sum_{a \neq a^*} \left(\frac{\tau}{A} \Delta(a) + 2(n - \tau - 1) \exp\Big(-2\tau\Delta(a)^2 \Big) \right).$$

- Difficult to tune: τ should be adjusted depending on n and $\Delta(a)$
- Worst-case w.r.t. $\Delta(a)$: $R_n = O(n^{2/3})$ (for $\tau = n^{2/3}$)

Explore-Then-Commit: Regret Analysis

- Regret decomposition

$$R_n = \sum_{t=1}^{\tau} \mathbb{E}\big[\nu(a^*) - \nu(a_t)\big] + \sum_{t=\tau+1}^{n} \mathbb{E}\big[\nu(a^*) - \nu(\widehat{a}^*)\big]$$

- During *explore* phase

$$\sum_{t=1}^{\tau} \mathbb{E}\big[\nu(a^*) - \nu(a_t)\big] = \frac{\tau}{A} \sum_{a \neq a^*} \Delta(a)$$

- During *exploit* phase

$$\begin{aligned}
\sum_{t=\tau+1}^{n} \mathbb{E}\big[\nu(a^*) - \nu(\widehat{a}^*)\big] &= (n-\tau-1) \sum_{a \neq a^*} \mathbb{P}\big[\widehat{a}^* = a\big] \Delta(a) \\
&= (n-\tau-1) \sum_{a \neq a^*} \mathbb{P}\big[\forall a' :\ \widehat{\mu}_\tau(a) \geq \widehat{\mu}_\tau(a')\big] \Delta(a) \\
&\leq (n-\tau-1) \sum_{a \neq a^*} \mathbb{P}\big[\widehat{\mu}_\tau(a) \geq \widehat{\mu}_\tau(a^*)\big] \Delta(a)
\end{aligned}$$

Explore-Then-Commit: Regret Analysis

Proposition (Chernoff-Hoeffding Inequality)

Let $X_i \in [a_i, b_i]$ *be* n *independent r.v. with mean* $\mu_i = \mathbb{E}X_i$. *Then*

$$\mathbb{P}\Big[\Big|\sum_{i=1}^{n}(X_i - \mu_i)\Big| \geq \epsilon\Big] \leq 2\exp\Big(-\frac{2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\Big).$$

Explore-Then-Commit: Regret Analysis

- Probability of error

$$\begin{aligned}\mathbb{P}\big[\widehat{\mu}_\tau(a) \geq \widehat{\mu}_\tau(a^*)\big] &= \mathbb{P}\big[\widehat{\mu}_\tau(a) - \mu(a) \geq \widehat{\mu}_\tau(a^*) - \mu(a^*) + \Delta(a)\big] \\ &\leq \mathbb{P}\big[\widehat{\mu}_\tau(a) - \mu(a) \geq \Delta(a)/2\big] + \mathbb{P}\big[\mu(a^*) - \widehat{\mu}_\tau(a^*) \geq \Delta(a)/2\big]\end{aligned}$$

- Hoeffding bound for random variables $r_t \in [0, 1]$

$$\mathbb{P}\big[\widehat{\mu}_\tau(a) \geq \widehat{\mu}_\tau(a^*)\big] \leq 2\exp\Big(-2\tau\Delta(a)^2\Big)$$

ϵ-greedy: Algorithm

- **For** $t = 1, \ldots, n$
  1. **Take action**

$$a_t = \begin{cases} \mathcal{U}(A) & \text{with probability } \epsilon_t \text{ (\textit{explore})} \\ \arg\max_a \widehat{\mu}_t(a) & \text{with probability } 1 - \epsilon_t \text{ (\textit{exploit})} \end{cases}$$

  2. Observe reward $r_t \sim \nu(a_t)$
  3. Update statistics for action a_t

$$T_t(a_t) = T_{t-1}(a_t) + 1$$

$$\widehat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a_t\}$$

- **EndFor**

ϵ-greedy: Regret

Theorem

If ϵ-greedy is run with parameter $\epsilon_t = \frac{CA}{\Delta_{\min}^2 n}$ *for n steps then it suffers a regret*

$$R_n \leq O\left(\frac{A\log(n)}{\Delta_{\min}}\right),$$

where $\Delta_{\min} = \min_a \Delta(a)$.

- Difficult to tune: optimal ϵ depends on knowledge of Δ
- Sharply separate exploration and exploitation
- Keep selecting very bad arms with some probability

Soft-max (aka Exp3): Algorithm

- **For** $t = 1, \dots, n$
  1. **Take action**

$$a_t \sim \frac{\exp\left(\widehat{\mu}_t(a)/\tau\right)}{\sum_{a'} \exp\left(\widehat{\mu}_t(a')/\tau\right)}$$

  2. Observe reward $r_t \sim \nu(a_t)$
  3. Update statistics for action a_t

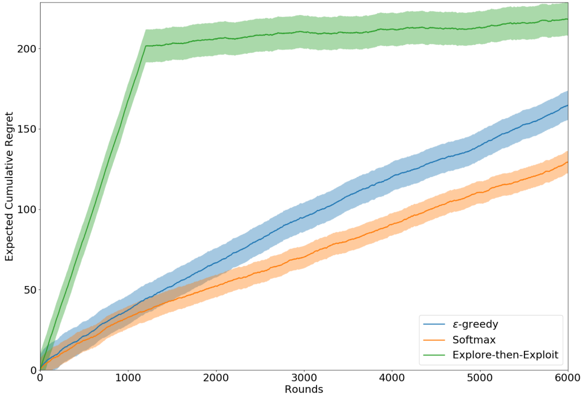
$$T_t(a_t) = T_{t-1}(a_t) + 1$$

$$\widehat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a_t\}$$

- **EndFor**

- More probability to better actions
- Temperature τ: large for exploration, small for exploitation
- Difficult to tune

Example of Regret Performance



Problem-Dependent Lower-bound

Theorem

Consider the family of multi-armed bandit problems with A Bernoulli arms and an algorithm that satisfies $\mathbb{E}[T_n(a)] = o(n^\alpha)$ for any $\alpha > 0$, any action a, and any Bernoulli MAB problem. Then for any Bernoulli MAB problem with gaps $\Delta(a) > 0$ for all $a \neq a^*$, any algorithm suffers regret

$$\lim \inf_{n\to\infty} \frac{R_n}{\log(n)} = \sum_{a \neq a^*} \frac{\Delta(a)}{kl(\mu(a), \mu(a^*))},$$

where $kl(p, q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$.

- No algorithm can achieve a regret smaller than $\Omega(\log n)$ (asymptotically)
- The ratio $\Delta(a)/\mathrm{kl}(a, a^*)$ measures the difficulty of the problem
- Algorithms such as ϵ-greedy with right tuning are optimal!

Problem-Independent Lower-bound

Theorem

Consider the family of multi-armed bandit problems with A Bernoulli arms. For any algorithm and fixed n, there exists a Bernoulli MAB problem such that

$$R_n = O\big(\sqrt{An}\big).$$

- At any finite time n, the regret may be as large as $\Omega(\sqrt{n})$

The Recipe for Effective Exp-Exp

1. Computation of estimates
2. Evaluation of uncertainty
3. Mechanism to combine estimates and uncertainty
4. Select the best action (according to its combined value)

Optimism in Face of Uncertainty

"Whenever the value of an action is **uncertain**, consider its *largest plausible* value, and then select the *best action*."

Optimism in Face of Uncertainty

"Whenever the value of an action is **uncertain**, consider its *largest plausible* value, and then select the *best action*."

Missing ingredient: uncertainty of our estimates

Measuring Uncertainty

Proposition (Chernoff-Hoeffding Inequality)

Let $X_i \in [a, b]$ *be* n *independent r.v. with mean* $\mu = \mathbb{E}X_i$. *Then*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mu\right| > (b-a)\sqrt{\frac{\log 2/\delta}{2n}}\right] \leq \delta$$

The Recipe of UCB

1. Computation of estimates

2. Evaluation of uncertainty

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a) - \mu(a)\right| \leq \sqrt{\frac{\log(1/\delta)}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a) - \mu(a)\right| \leq \sqrt{\frac{\log(1/\delta)}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \rho \sqrt{\frac{\log(1/\delta_t)}{T_t(a)}}$$

4. Select the best action (according to its combined value)

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a) - \mu(a)\right| \leq \sqrt{\frac{\log(1/\delta)}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \rho\sqrt{\frac{\log(1/\delta_t)}{T_t(a)}}$$

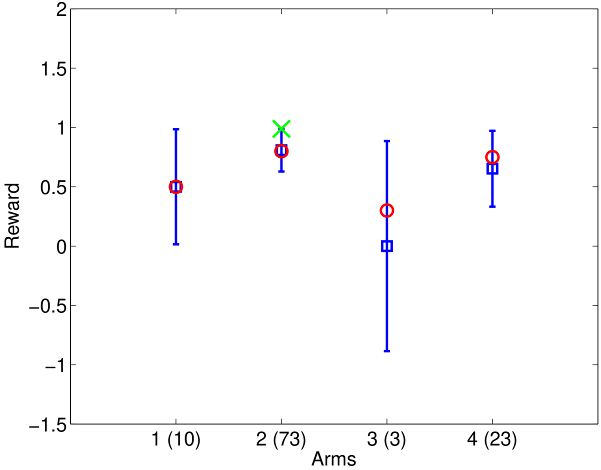
4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

UCB: Algorithm

- **For** $t = 1, \dots, n$
  1. Compute upper-confidence bound
$$B_t(a) = \widehat{\mu}_t(a) + \rho\sqrt{\frac{\log(1/\delta_t)}{T_t(a)}}$$
  2. **Take action** $a_t \arg\max_a B_t(a)$
  3. Observe reward $r_t \sim \nu(a_t)$
  4. Update statistics for action a_t
$$T_t(a_t) = T_{t-1}(a_t) + 1$$
$$\widehat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a_t\}$$
- **EndFor**

UCB: Algorithm



UCB: Regret

Theorem

Consider a MAB problem with A Bernoulli arms with gaps $\Delta(a)$. If UCB is run with $\rho = 1$ and $\delta_t = 1/t$ for n steps, then it suffers a regret

$$R_n = O\left(\sum_{a \neq a^*} \frac{\log(n)}{\Delta(a)}\right)$$

Consider a 2-action MAB problem, then for any fixed n, in the worst-case (w.r.t. Δ) UCB suffers a regret

$$R_n = O\left(\sqrt{n \log(n)}\right)$$

- It (almost) matches the lower bounds
- It does not require any prior knowledge about the MAB, apart from the range of the r.v.
- The big-O hides a few numerical constants and n-independent additive terms

UCB: Proof Sketch

Disclaimer: this is a slightly suboptimal proof, but it provides an easy path.

Define the (high-probability) event *[statistics]*

$$\mathcal{E} = \left\{ \forall a, t \ \left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \sqrt{\frac{\log 1/\delta}{2T_t(a)}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.

UCB: Proof Sketch

Disclaimer: this is a slightly suboptimal proof, but it provides an easy path.

Define the (high-probability) event *[statistics]*

$$\mathcal{E} = \left\{ \forall a, t \;\; \left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \sqrt{\frac{\log 1/\delta}{2T_t(a)}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
If at time t we select action a then *[algorithm]*

$$B_t(a) \geq B_t(a^*)$$

UCB: Proof Sketch

Disclaimer: this is a slightly suboptimal proof, but it provides an easy path.

Define the (high-probability) event *[statistics]*

$$\mathcal{E} = \left\{ \forall a, t \;\; \left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \sqrt{\frac{\log 1/\delta}{2T_t(a)}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
If at time t we select action a then *[algorithm]*

$$\widehat{\mu}_t(a) + \sqrt{\frac{\log 1/\delta}{T_t(a))}} \geq \widehat{\mu}_t(a^*) + \sqrt{\frac{\log 1/\delta}{T_t(a^*)}}$$

UCB: Proof Sketch

Disclaimer: this is a slightly suboptimal proof, but it provides an easy path.

Define the (high-probability) event *[statistics]*

$$\mathcal{E} = \left\{ \forall a, t \ \left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \sqrt{\frac{\log 1/\delta}{2T_t(a)}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
If at time t we select action a then *[algorithm]*

$$\widehat{\mu}_t(a) + \sqrt{\frac{\log 1/\delta}{T_t(a))}} \geq \widehat{\mu}_t(a^*) + \sqrt{\frac{\log 1/\delta}{T_t(a^*)}}$$

On the event $\mathcal{E}$ we have *[math]*

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{T_t(a)}} \geq \mu(a^*)$$

UCB: Proof Sketch

Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$, thus

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{(T_n(a) - 1)}} \geq \mu(a^*)$$

UCB: Proof Sketch

Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$, thus

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{(T_n(a) - 1)}} \geq \mu(a^*)$$

Reordering *[math]*

$$T_n(a) \leq \frac{\log(1/\delta)}{\Delta(a)^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nK\delta$.

UCB: Proof Sketch

Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$, thus

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{(T_n(a) - 1)}} \geq \mu(a^*)$$

Reordering *[math]*

$$T_n(a) \leq \frac{\log(1/\delta)}{\Delta(a)^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nK\delta$.
Moving to the expectation *[statistics]*

$$\mathbb{E}[T_n(a)] = \mathbb{E}[T_n(a)\mathbb{I}\mathcal{E}] + \mathbb{E}[T_n(a)\mathbb{I}\mathcal{E}^C]$$

UCB: Proof Sketch

Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$, thus

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{(T_n(a) - 1)}} \geq \mu(a^*)$$

Reordering *[math]*

$$T_n(a) \leq \frac{\log(1/\delta)}{\Delta(a)^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nK\delta$.
Moving to the expectation *[statistics]*

$$\mathbb{E}[T_n(a)] \leq \frac{\log(1/\delta)}{2\Delta(a)^2} + 1 + n(nK\delta)$$

UCB: Proof Sketch

Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$, thus

$$\mu(a) + 2\sqrt{\frac{\log 1/\delta}{(T_n(a) - 1)}} \geq \mu(a^*)$$

Reordering *[math]*

$$T_n(a) \leq \frac{\log(1/\delta)}{\Delta(a)^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nK\delta$.
Moving to the expectation *[statistics]*

$$\mathbb{E}[T_n(a)] \leq \frac{\log(1/\delta)}{2\Delta(a)^2} + 1 + n(nK\delta)$$

Trading-off the two terms $\delta = 1/n^2$, we obtain

$$\mathbb{E}[T_n(a)] \leq \frac{\log n}{\Delta_i^2} + 1 + K$$

Tuning the ρ Parameter

Theory

- $\rho < 1$, polynomial regret w.r.t. n
- $\rho \geq 1$, logarithmic regret w.r.t. n

Tuning the ρ Parameter

Theory

- $\rho < 1$, polynomial regret w.r.t. n
- $\rho \geq 1$, logarithmic regret w.r.t. n

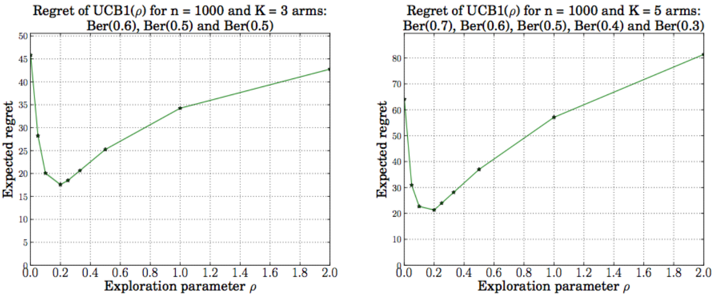
Practice: $\rho = 0.2$ is often the best choice

Tuning the ρ Parameter

Theory

- $\rho < 1$, polynomial regret w.r.t. n
- $\rho \geq 1$, logarithmic regret w.r.t. n

Practice: $\rho = 0.2$ is often the best choice



Improvements: UCB-V

Idea: use *empirical Bernstein bounds* for more accurate c.i.

Improvements: UCB-V

Idea: use *empirical Bernstein bounds* for more accurate c.i.

Algorithm

- Compute the *score* of each arm i

$$B_t(a) = \widehat{\mu}_t(a) + \rho \sqrt{\frac{\log(t)}{T_t(a)}}$$

- Select action

$$a_t = \arg\max_a B_t(a)$$

- Update the statistics $T_t(a_t)$, $\widehat{\mu}_t(a_t)$

Improvements: UCB-V

Idea: use *empirical Bernstein bounds* for more accurate c.i.

Algorithm

- Compute the *score* of each arm i

$$B_t(a) = \widehat{\mu}_t(a) + \sqrt{\frac{2\widehat{\sigma}_t^2(a)\log t}{T_t(a)}} + \frac{8\log t}{3T_t(a)}$$

- Select action

$$a_t = \arg\max_a B_t(a)$$

- Update the statistics $T_t(a_t)$, $\widehat{\mu}_t(a_t)$ and $\widehat{\sigma}_t^2(a_t)$

Improvements: UCB-V

Idea: use *empirical Bernstein bounds* for more accurate c.i.

Algorithm

- Compute the *score* of each arm i

$$B_t(a) = \widehat{\mu}_t(a) + \sqrt{\frac{2\widehat{\sigma}_t^2(a)\log t}{T_t(a)}} + \frac{8\log t}{3T_t(a)}$$

- Select action

$$a_t = \arg\max_a B_t(a)$$

- Update the statistics $T_t(a_t)$, $\widehat{\mu}_t(a_t)$ and $\widehat{\sigma}_t^2(a_t)$

Regret

$$R_n \leq O\Big(\frac{1}{\Delta}\log n\Big)$$

Improvements: UCB-V

Idea: use *empirical Bernstein bounds* for more accurate c.i.

Algorithm

- Compute the *score* of each arm i

$$B_t(a) = \widehat{\mu}_t(a) + \sqrt{\frac{2\widehat{\sigma}_t^2(a)\log t}{T_t(a)}} + \frac{8\log t}{3T_t(a)}$$

- Select action

$$a_t = \arg\max_a B_t(a)$$

- Update the statistics $T_t(a_t)$, $\widehat{\mu}_t(a_t)$ and $\widehat{\sigma}_t^2(a_t)$

Regret

$$R_n \leq O\Big(\frac{\sigma^2}{\Delta}\log n\Big)$$

Improvements: KL-UCB

Idea: use even tighter c.i. based on *Kullback–Leibler divergence*

$$\mathsf{kl}(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}$$

Improvements: KL-UCB

Idea: use even tighter c.i. based on *Kullback–Leibler divergence*

$$\mathsf{kl}(p,q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

Algorithm: Compute the *score* of each arm i (convex optimization)

$$B_t(a) = \max \Big\{ q \in [0,1] : T_t(a) \mathsf{kl}\big(\widehat{\mu}_t(a), q\big) \leq \log(t) + c \log(\log(t)) \Big\}$$

Improvements: KL-UCB

Idea: use even tighter c.i. based on *Kullback–Leibler divergence*

$$\mathsf{kl}(p,q) = p \log \frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$$

Algorithm: Compute the *score* of each arm i (convex optimization)

$$B_t(a) = \max\Big\{ q \in [0,1] : T_t(a)\mathsf{kl}\big(\widehat{\mu}_t(a), q\big) \leq \log(t) + c\log(\log(t)) \Big\}$$

Regret: pulls to suboptimal arms

$$\mathbb{E}\big[T_n(a)\big] \leq (1+\epsilon)\frac{\log(n)}{\mathsf{kl}(\mu(a),\mu(a^*))} + C_1\log(\log(n)) + \frac{C_2(\epsilon)}{n^{\beta(\epsilon)}}$$

where $d(\mu_i,\mu^*) \geq 2\Delta_i^2$

Measuring Uncertainty

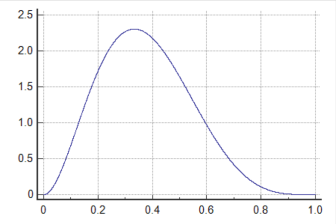
- Assume that $r_t(a)$ are distributed as Bernoulli for all actions a with parameter $\mu(a)$
- Define a prior $\mu(a) \sim \text{Beta}(\alpha_0, \beta_0)$
- After t rewards, compute the posterior for action a as $\text{Beta}\big(\alpha_t(a), \beta_t(a)\big)$ with

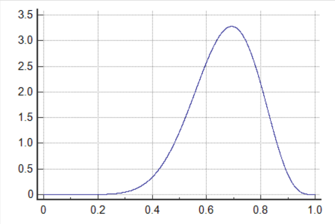
$$\alpha_t(a) = \alpha_0 + \sum_{s=1}^{t} \mathbb{I}\{a_t = a \wedge r_t = 0\} \qquad \beta_t(a) = \beta_0 + \sum_{s=1}^{t} \mathbb{I}\{a_t = a \wedge r_t = 1\}$$

Measuring Uncertainty

- Assume that $r_t(a)$ are distributed as Bernoulli for all actions a with parameter $\mu(a)$
- Define a prior $\mu(a) \sim \text{Beta}(\alpha_0, \beta_0)$
- After t rewards, compute the posterior for action a as $\text{Beta}\big(\alpha_t(a), \beta_t(a)\big)$ with

$$\alpha_t(a) = \alpha_0 + \sum_{s=1}^{t} \mathbb{I}\{a_t = a \wedge r_t = 0\} \qquad \beta_t(a) = \beta_0 + \sum_{s=1}^{t} \mathbb{I}\{a_t = a \wedge r_t = 1\}$$





The Recipe of Thompson Sampling*

1. Computation of estimates (from posterior)
2. Evaluation of uncertainty
3. Mechanism to combine estimates and uncertainty
4. Select the best action (according to its combined value)

*aka Posterior sampling

The Recipe of Thompson Sampling*

1. Computation of estimates (from posterior)

$$\widehat{\mu}_t(a_t) = \frac{\alpha_t(a)}{\alpha_t(a) + \beta_t(a)}$$

2. Evaluation of uncertainty

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

*aka Posterior sampling

The Recipe of Thompson Sampling*

1. Computation of estimates (from posterior)

$$\widehat{\mu}_t(a_t) = \frac{\alpha_t(a)}{\alpha_t(a) + \beta_t(a)}$$

2. Evaluation of uncertainty

$$\text{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

*aka Posterior sampling

The Recipe of Thompson Sampling*

1. Computation of estimates (from posterior)

$$\widehat{\mu}_t(a_t) = \frac{\alpha_t(a)}{\alpha_t(a) + \beta_t(a)}$$

2. Evaluation of uncertainty

$$\mathsf{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) \sim \mathsf{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

4. Select the best action (according to its combined value)

*aka Posterior sampling

The Recipe of Thompson Sampling*

1. Computation of estimates (from posterior)

$$\widehat{\mu}_t(a_t) = \frac{\alpha_t(a)}{\alpha_t(a) + \beta_t(a)}$$

2. Evaluation of uncertainty

$$\mathsf{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) \sim \mathsf{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

*aka Posterior sampling

TS: Algorithm

- **For** $t = 1, \dots, n$
  1. Compute upper-confidence bound

$$B_t(a) \sim \text{Beta}\big(\alpha_t(a), \beta_t(a)\big)$$

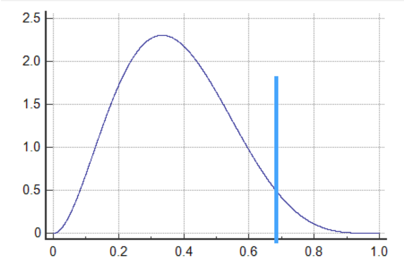
  2. **Take action** $a_t \in \arg\max_a B_t(a)$
  3. Observe reward $r_t \sim \nu(a_t)$
  4. Update statistics for action a_t

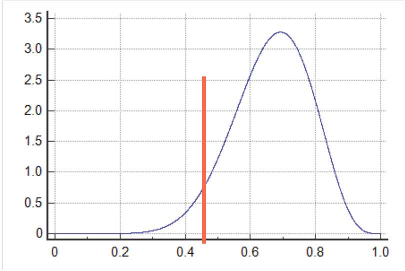
$$\alpha_t(a_t) = \alpha_{t-1}(a_t) + \mathbb{I}\{r_t = 0\}$$

$$\beta_t(a_t) = \beta_{t-1}(a_t) + \mathbb{I}\{r_t = 1\}$$

- **EndFor**

TS: Algorithm





TS: Regret

Theorem

Consider a MAB problem with A Bernoulli arms with gaps $\Delta(a)$. If UCB is run with $\rho = 1$ and $\delta_t = 1/t$ for n steps, then it suffers a regret

$$R_n = O\left((1+\epsilon)\sum_{a \neq a^*} \frac{\Delta(a)\log(n)}{kl(\mu(a), \mu(a^*))}\right)$$

- It matches the lower bound
- It requires defining a prior on the actions

A Simple Recommendation System

- A RS can recommend *specific movies*
- Users arrive at random and *no information about the user is available*
- The RS picks a movie to the user
- The feedback is whether the user *watched* the or not
- *Objective:* design a RS that maximizes that number of movies watched in the recommended genre

RS as a Multi-armed Bandit

For $t = 1, \ldots, n$

1. User arrives
2. **Recommend movie a_t**
3. **Reward**

$$r_t = \begin{cases} 1 & \text{user watches movie } a_t \\ 0 & \text{otherwise} \end{cases}$$

EndFor

Issue: too many movies are available to collect enough feedback for each movie separately

RS as Linear Bandit

The *model*

- $\mu(a) = \mathbb{E}\big[r(a)\big]$ is the probability a *random* user watches movie a
- Each movie a is characterized by some features $\phi(a) \in \mathbb{R}^d$ (e.g., genre, release date, past rating, income)
- **Assumption**:
  - the expected value is a linear function $\mu(a) = \phi(a)^\mathsf{T}\theta^*$ (with $\theta^* \in \mathbb{R}^d$ unknown)
  - the rewards are noisy observations $r_t(a) = \mu(a) + \eta_t$ with $\mathbb{E}[\eta_t] = 0$

The *objective*

- Maximize sum of reward $\mathbb{E}\Big[\sum_{t=1}^{n} r_t\Big]$

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a_t) - \mu(a)\right| \leq \sqrt{\frac{\log(1/\delta)}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \rho\sqrt{\frac{\log(1/\delta_t)}{T_t(a)}}$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

The Recipe of UCB

1. Computation of estimates

$$\widehat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a_t) - \mu(a)\right| \leq \sqrt{\frac{\log(1/\delta)}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \rho\sqrt{\frac{\log(1/\delta_t)}{T_t(a)}}$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

Issue: $T_t(a)$ is likely to be 0 for most a, we need more **sample efficient** estimates

The Regret

$$R_n = \max_a \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a)\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(a_t)\Big]$$

$$= \mathbb{E}\Big[\sum_{t=1}^{n} \big(\phi(a^*) - \phi(a_t)\big)^\mathsf{T} \theta^*\Big]$$

Issue: a^* unlikely to be ever selected if $n \ll A$

Least-Squares Estimate of θ^*

- Least-squares estimate

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{t} \sum_{s=1}^{t} \left(r_s - \phi(a_s)^\mathsf{T} \theta \right)^2 + \lambda \|\theta\|^2$$

- Closed form solution

$$A_t = \sum_{s=1}^{t} \phi(a_s)\phi(a_s)^\mathsf{T} + \lambda I \qquad b_t = \sum_{s=1}^{t} \phi(a_s) r_s$$

$$\Rightarrow \widehat{\theta}_t = A_t^{-1} b_t$$

- Estimate of value of action a

$$\widehat{\mu}_t(a) = \phi(a)^\mathsf{T} \widehat{\theta}_t$$

Measuring Uncertainty

Proposition

Let $a_1, \dots, a_t$ any sequence of actions adapted to the filtration $\mathcal{F}_t$. If the noise η is sub-Gaussian of parameter B and the features are bounded $\|\phi(a)\|_2 \leq L$, then for any a with probability $1-\delta$

$$\left|\widehat{\mu}_t(a) - \mu(a)\right| \leq \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)},$$

where

$$\alpha_t = B\sqrt{d \log\left(\frac{1+tL/\lambda}{\delta}\right)} + \lambda^{1/2}\|\theta^*\|_2$$

- $\|\phi(a)\|_{A_t^{-1}}$ measures the correlation between $\phi(a)$ and the actions selected so far
- If $\{\phi(a)\}_a$ is an orthogonal basis for $\mathbb{R}^A$, this reduces to the MAB problem and $\|\phi(a)\|_{A_t^{-1}} = \sqrt{\frac{1}{T_t(a)}}$.

The Recipe of LinUCB

1. Computation of estimates

$$\widehat{\theta}_t = A_t^{-1} b_t \qquad \widehat{\mu}_t(a) = \phi(a)^\mathsf{T} \widehat{\theta}_t$$

2. Evaluation of uncertainty

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

The Recipe of LinUCB

1. Computation of estimates

$$\widehat{\theta}_t = A_t^{-1} b_t \qquad \widehat{\mu}_t(a) = \phi(a)^\mathsf{T} \widehat{\theta}_t$$

2. Evaluation of uncertainty

$$\left|\widehat{\mu}_t(a) - \mu(a)\right| \leq \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$

3. Mechanism to combine estimates and uncertainty

4. Select the best action (according to its combined value)

The Recipe of LinUCB

1. Computation of estimates

$$\widehat{\theta}_t = A_t^{-1} b_t \qquad \widehat{\mu}_t(a) = \phi(a)^\mathsf{T} \widehat{\theta}_t$$

2. Evaluation of uncertainty

$$\left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$

4. Select the best action (according to its combined value)

The Recipe of LinUCB

1. Computation of estimates

$$\widehat{\theta}_t = A_t^{-1} b_t \qquad \widehat{\mu}_t(a) = \phi(a)^\mathsf{T} \widehat{\theta}_t$$

2. Evaluation of uncertainty

$$\left| \widehat{\mu}_t(a) - \mu(a) \right| \leq \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \widehat{\mu}_t(a) + \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

LinUCB: Algorithm

- **For** $t = 1, \dots, n$
  1. Compute upper-confidence bound
  $$B_t(a) = \widehat{\mu}_t(a) + \alpha_t \sqrt{\phi(a)^\mathsf{T} A_t^{-1} \phi(a)}$$
  2. **Take action** $a_t \arg\max_a B_t(a)$
  3. Observe reward $r_t \sim \phi(a_t)^\mathsf{T} \theta^* + \eta_t$
  4. Update statistics
  $$A_{t+1} = A_t + \phi(a_t)\phi(a_t)^\mathsf{T}$$
  $$\widehat{\theta}_{t+1} = A_{t+1}^{-1} b_{t+1}$$
- **EndFor**

LinUCB: Regret

Theorem

Consider a linear MAB problem with actions defined in Re^d and unknown parameter $\theta^* \in \mathbb{R}^d$. If LinUCB is run with $\delta_t = 1/t$ for n steps, then it suffers a regret

$$R_n = O\big(d\sqrt{n\log(n)}\big)$$

- It depends on d but not the number of actions A
- If $A < \infty$ we can improve the bound to

$$R_n = O\big(\sqrt{dn\log(nA)}\big)$$

A Simple Recommendation System

- A RS can recommend *specific movies*
- Users arrive at random and *we have information about them*
- The RS picks a movie to the user
- The feedback is whether the user *watched* the or not
- *Objective:* design a RS that maximizes that number of movies watched in the recommended genre

RS as a Multi-armed Bandit

For $t = 1, \dots, n$

1. User arrives u_t
2. Recommend movie a_t
3. Reward

$$r_t = \begin{cases} 1 & \text{user watches movie } a_t \\ 0 & \text{otherwise} \end{cases}$$

EndFor

Issue: too many users to collect enough feedback for each user separately

RS as Contextual Linear Bandit

The *model*

- $\mu(u, a) = \mathbb{E}\big[r(u, a)\big]$ is the probability user u watches movie a
- Each user u and movie a are characterized by some features $\phi(u, a) \in \mathbb{R}^d$ (e.g., name, location, genre, release date, past rating, income)
- **Assumption**:
  - the expected value is a linear function $\mu(u, a) = \phi(u, a)^\mathsf{T} \theta^*$ (with $\theta^* \in \mathbb{R}^d$ unknown)
  - the rewards are noisy observations $r_t(u, a) = \mu(u, a) + \eta_t$ with $\mathbb{E}[\eta_t] = 0$

The *objective*

- Maximize sum of reward $\mathbb{E}\Big[\sum_{t=1}^{n} r_t\Big]$

The Regret

$$
\begin{aligned}
R_n &= \mathbb{E}\Big[\sum_{t=1}^{n} \max_{a} r_t(u_t, a)\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r_t(u_t, a_t)\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^{n} \big(\phi(u_t, a_t^*) - \phi(u_t, a_t)\big)^\mathsf{T} \theta^*\Big]
\end{aligned}
$$

Least-Squares Estimate of θ^*

- Least-squares estimate

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{t} \sum_{s=1}^{t} \left(r_s - \phi(u_s, a_s)^\mathsf{T} \theta \right)^2 + \lambda \|\theta\|^2$$

- Closed form solution

$$A_t = \sum_{s=1}^{t} \phi(u_s, a_s) \phi(u_s, a_s)^\mathsf{T} + \lambda I \qquad b_t = \sum_{s=1}^{t} \phi(u_s, a_s) r_s$$

$$\Rightarrow \widehat{\theta}_t = A_t^{-1} b_t$$

- Estimate of value of action a

$$\widehat{\mu}_t(u, a) = \phi(u, a)^\mathsf{T} \widehat{\theta}_t$$

ContextualLinUCB: Algorithm

- **For** $t = 1, \dots, n$
  1. Observe *context* u_t
  2. Compute upper-confidence bound

$$B_t(u_t, a) = \widehat{\mu}_t(u_t, a) + \alpha_t \sqrt{\phi(u_t, a)^\mathsf{T} A_t^{-1} \phi(ut_, a)}$$

  3. **Take action** $a_t \arg\max_a B_t(u_t, a)$
  4. Observe reward $r_t \sim \phi(u_t, a_t)^\mathsf{T} \theta^* + \eta_t$
  5. Update statistics

$$A_{t+1} = A_t + \phi(u_t, a_t)\phi(u_t, a_t)^\mathsf{T}$$

$$\widehat{\theta}_{t+1} = A_{t+1}^{-1} b_{t+1}$$

- **EndFor**

ContextualLinUCB: Regret

Theorem

*Consider a contextual linear MAB problem with contexts and actions defined in Re^d and unknown parameter $\theta^* \in \mathbb{R}^d$. If ContextualLinUCB is run with $\delta_t = 1/t$ for n steps, then for **any arbitrary sequence of contexts** $u_1, u_2, \ldots, u_n$ it suffers a regret*

$$R_n = O\big(d\sqrt{n\log(n)}\big)$$

Summary

- Basic exploration strategies: explore-then-commit, ϵ-greedy, softmax
- Advanced strategies: UCB, Thompson sampling
- Linear and contextual linear bandit

Bibliography

Thank you!

facebook
Artificial Intelligence Research