The Exploration-Exploitation Dilemma

A. LAZARIC (*SequeL Team @INRIA-Lille*)
ENS Cachan - Master 2 MVA

SequeL – INRIA Lille

The Exploration-Exploitation Dilemma

The Exploration-Exploitation Dilemma

Tools

Stochastic Multi-Armed Bandit

Contextual Linear Bandit

Other Multi-Armed Bandit Problems

Learning the Optimal Policy

For $i = 1, \ldots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)
    3.1 Take action a_t ***according to a suitable exploration policy***
    3.2 Observe next state x_{t+1} and reward r_t
    3.3 Compute the temporal difference δ_t (e.g., Q-learning)
    3.4 Update the Q-function

$$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

    3.5 Set $t = t + 1$

   EndWhile

EndFor

Learning the Optimal Policy

For $i = 1, \ldots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)

    3.1 ***Take action*** $a_t = \arg\max_a Q(x_t, a)$

    3.2 Observe next state x_{t+1} and reward r_t

    3.3 Compute the temporal difference δ_t (e.g., Q-learning)

    3.4 Update the Q-function

$$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

    3.5 Set $t = t + 1$

    EndWhile

EndFor

Learning the Optimal Policy

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)

   3.1 ***Take action*** $a_t = \arg\max_a Q(x_t, a)$

   3.2 Observe next state x_{t+1} and reward r_t

   3.3 Compute the temporal difference δ_t (e.g., Q-learning)

   3.4 Update the Q-function

   $$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

   3.5 Set $t = t + 1$

   EndWhile

EndFor

$\Rightarrow$ ***no convergence***

Learning the Optimal Policy

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)

    3.1 ***Take action*** $a_t \sim \mathcal{U}(A)$

    3.2 Observe next state x_{t+1} and reward r_t

    3.3 Compute the temporal difference δ_t (e.g., Q-learning)

    3.4 Update the Q-function

    $$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

    3.5 Set $t = t + 1$

    EndWhile

EndFor

Learning the Optimal Policy

For $i = 1, \ldots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)

    3.1 ***Take action*** $a_t \sim \mathcal{U}(A)$

    3.2 Observe next state x_{t+1} and reward r_t

    3.3 Compute the temporal difference δ_t (e.g., Q-learning)

    3.4 Update the Q-function

    $$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

    3.5 Set $t = t + 1$

    EndWhile

EndFor

$\Rightarrow$ ***very poor rewards***

The Exploration-Exploitation Dilemma

Tools

Contextual Linear Bandit

Stochastic Multi-Armed Bandit

Other Multi-Armed Bandit Problems

Concentration Inequalities

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

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

Concentration Inequalities

Proof.

$$\begin{aligned}
\mathbb{P}\Big(\sum_{i=1}^{n} X_i - \mu_i \geq \epsilon\Big) &= \mathbb{P}(e^{s\sum_{i=1}^{n} X_i - \mu_i} \geq e^{s\epsilon}) \\
&\leq e^{-s\epsilon}\mathbb{E}[e^{s\sum_{i=1}^{n} X_i - \mu_i}], \quad \text{Markov inequality} \\
&= e^{-s\epsilon}\prod_{i=1}^{n}\mathbb{E}[e^{s(X_i-\mu_i)}], \quad \text{independent random variables} \\
&\leq e^{-s\epsilon}\prod_{i=1}^{n} e^{s^2(b_i-a_i)^2/8}, \quad \text{Hoeffding inequality} \\
&= e^{-s\epsilon + s^2\sum_{i=1}^{n}(b_i-a_i)^2/8}
\end{aligned}$$

If we choose $s = 4\epsilon/\sum_{i=1}^{n}(b_i - a_i)^2$, the result follows.
Similar arguments hold for $\mathbb{P}\big(\sum_{i=1}^{n} X_i - \mu_i \leq -\epsilon\big)$.

Concentration Inequalities

Finite sample guarantee:

$$\mathbb{P}\Big[\underbrace{\Big|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\Big|}_{deviation} > \underbrace{\epsilon}_{accuracy}\Big] \leq \underbrace{2\exp\Big(-\frac{2n\epsilon^2}{(b-a)^2}\Big)}_{confidence}$$

Concentration Inequalities

Finite sample guarantee:

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > (b-a)\sqrt{\frac{\log 2/\delta}{2n}}\right] \leq \delta$$

Concentration Inequalities

Finite sample guarantee:

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > \epsilon\right] \leq \delta$$

if $n \geq \frac{(b-a)^2 \log 2/\delta}{2\epsilon^2}$.

The Exploration-Exploitation Dilemma

Reducing RL down to Multi-Armed Bandit

Definition (Markov decision process)

*A **Markov decision process** is defined as a tuple $M = (X, A, p, r)$:*

- ~~*X is the state space,*~~
- *A is the action space,*
- ~~*$p(y|x,a)$ is the transition probability*~~
- ~~*$r(x,a,y)$ is the reward of transition (x,a,y)*~~
  $\Rightarrow r(a)$ is the reward of action a

Notice

For coherence with the bandit literature we use the notation

- $i = 1, \dots, K$ set of possible actions
- $t = 1, \dots, n$ time
- I_t action selected at time t
- $X_{i,t}$ reward for action i at time t

Learning the Optimal Policy

Objective: learn the optimal policy π^* ***as efficiently as possible***

Learning the Optimal Policy

Objective: learn the optimal policy π^* ***as efficiently as possible***

For $t = 1, \ldots, n$

1. ~~Set~~ $\cancel{t = 0}$
2. ~~Set initial state~~ $\cancel{x_0}$
3. ~~**While**~~ ~~(~~$\cancel{x_t}$ ~~not terminal)~~
   3.1 Take action a_t
   3.2 Observe ~~next state~~ $\cancel{x_{t+1}}$ ~~and~~ reward r_t
   3.3 ~~Set~~ $\cancel{t = t + 1}$

   ~~**EndWhile**~~

EndFor

The Multi–armed Bandit Protocol

The learner has $i = 1, \ldots, K$ arms (actions)

At each round $t = 1, \ldots, n$

The Multi–armed Bandit Protocol

The learner has $i = 1, \ldots, K$ arms (actions)

At each round $t = 1, \ldots, n$

- At the same time

The Multi–armed Bandit Protocol

The learner has $i = 1, \ldots, K$ arms (actions)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses a vector of *rewards* $\{X_{i,t}\}_{i=1}^{K}$
  - The learner chooses an arm I_t

The Multi–armed Bandit Protocol

The learner has $i = 1, \dots, K$ arms (actions)

At each round $t = 1, \dots, n$

- At the same time
  - The environment chooses a vector of *rewards* $\{X_{i,t}\}_{i=1}^{K}$
  - The learner chooses an arm I_t
- The learner receives a reward $X_{I_t,t}$

The Multi–armed Bandit Protocol

The learner has $i = 1, \ldots, K$ arms (actions)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses a vector of *rewards* $\{X_{i,t}\}_{i=1}^{K}$
  - The learner chooses an arm I_t
- The learner receives a reward $X_{I_t,t}$
- The environment ***does not*** reveal the rewards of the other arms

The Multi–armed Bandit Game (cont'd)

The regret

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

The Multi–armed Bandit Game (cont'd)

The regret

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

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

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner

$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms

Problem 2: Whenever the learner pulls a ***bad arm***, it suffers some regret

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms

Problem 2: Whenever the learner pulls a ***bad arm***, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly pulling the best arm

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms

Problem 2: Whenever the learner pulls a ***bad arm***, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly pulling the best arm

Challenge: The learner should solve two opposite problems!

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms
$\Rightarrow$ ***exploration***

Problem 2: Whenever the learner pulls a ***bad arm***, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly pulling the best arm

Challenge: The learner should solve two opposite problems!

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms
$\Rightarrow$ ***exploration***

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

The Exploration–Exploitation Lemma

Problem 1: The environment ***does not*** reveal the rewards of the arms not pulled by the learner
$\Rightarrow$ the learner should *gain information* by repeatedly pulling all the arms
$\Rightarrow$ ***exploration***

Problem 2: Whenever the learner pulls a ***bad arm***, it suffers some regret
$\Rightarrow$ the learner should *reduce the regret* by repeatedly pulling the best arm
$\Rightarrow$ ***exploitation***

Challenge: The learner should solve the *exploration-exploitation* dilemma!

The Multi–armed Bandit Game (cont'd)

Examples

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

The Stochastic Multi–armed Bandit Problem

Definition

The environment is stochastic

- *Each arm has a distribution* ν_i *bounded in* $[0,1]$ *and characterized by an expected value* μ_i
- *The rewards are i.i.d.* $X_{i,t} \sim \nu_i$ *(as in the MDP model)*

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} (n\mu_i) - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} (n\mu_i) - \sum_{i=1}^{K} \mathbb{E}[T_{i,n}]\mu_i$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(\mathcal{A}) = n\mu_{i^*} - \sum_{i=1}^{K} \mathbb{E}[T_{i,n}]\mu_i$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(\mathcal{A}) = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}](\mu_{i^*} - \mu_i)$$

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

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

The Stochastic Multi–armed Bandit Problem (cont'd)

Notation

- Number of times arm i has been pulled after n rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

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

- Gap $\Delta_i = \mu_{i^*} - \mu_i$

The Stochastic Multi–armed Bandit Problem (cont'd)

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

$\Rightarrow$ we only need to study the *expected number of pulls* of the *suboptimal* arms

The Stochastic Multi–armed Bandit Problem (cont'd)

Optimism in Face of Uncertainty Learning (OFUL)

Whenever we are *uncertain* about the outcome of an arm, we consider the *best possible world* and choose the *best arm*.

The Stochastic Multi–armed Bandit Problem (cont'd)

Optimism in Face of Uncertainty Learning (OFUL)

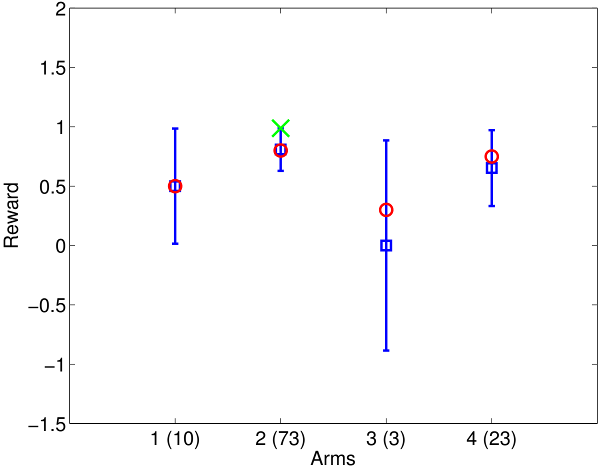
Whenever we are *uncertain* about the outcome of an arm, we consider the *best possible world* and choose the *best arm*.

Why it works:

- If the *best possible world* is correct $\Rightarrow$ *no regret*
- If the *best possible world* is wrong $\Rightarrow$ *the reduction in the uncertainty is maximized*

The Upper–Confidence Bound (UCB) Algorithm

The idea



The Upper–Confidence Bound (UCB) Algorithm

Show time!

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

At each round $t = 1, \ldots, n$

- Compute the *score* of each arm i

$$B_i = (\textit{optimistic} \text{ score of arm } i)$$

- Pull arm

$$I_t = \arg\max_{i=1,\ldots,K} B_{i,s,t}$$

- Update the number of pulls $T_{I_t,t} = T_{I_t,t-1} + 1$ and the other statistics

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

The score (with parameters ρ and δ)

$$B_i = (\textit{optimistic} \text{ score of arm } i)$$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

The score (with parameters ρ and δ)

$B_{i,s,t} =$ (*optimistic* score of arm i if pulled s times up to round t)

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

The score (with parameters ρ and δ)

$B_{i,s,t} =$ (*optimistic* score of arm i if pulled s times up to round t)

Optimism in face of uncertainty:
Current knowledge: average rewards $\hat{\mu}_{i,s}$
Current uncertainty: number of pulls s

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

The score (with parameters ρ and δ)

$$B_{i,s,t} = \text{knowledge} \underbrace{+}_{\textit{optimism}} \text{uncertainty}$$

Optimism in face of uncertainty:
Current knowledge: average rewards $\hat{\mu}_{i,s}$
Current uncertainty: number of pulls s

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

The score (with parameters ρ and δ)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log 1/\delta}{2s}}$$

Optimism in face of uncertainty:
Current knowledge: average rewards $\hat{\mu}_{i,s}$
Current uncertainty: number of pulls s

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

At each round $t = 1, \ldots, n$

- Compute the *score* of each arm i

$$B_{i,t} = \hat{\mu}_{i,T_{i,t}} + \rho \sqrt{\frac{\log(t)}{2T_{i,t}}}$$

- Pull arm

$$I_t = \arg \max_{i=1,\ldots,K} B_{i,t}$$

- Update the number of pulls $T_{I_t,t} = T_{I_t,t-1} + 1$ and $\hat{\mu}_{i,T_{i,t}}$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

Theorem

Let $X_1, \ldots, X_n$ be i.i.d. samples from a distribution bounded in $[a, b]$, then for any $\delta \in (0, 1)$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > (b-a)\sqrt{\frac{\log 2/\delta}{2n}}\right] \leq \delta$$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

After s pulls, arm i

$$\mathbb{P}\left[\mathbb{E}[X_i] \leq \frac{1}{s}\sum_{t=1}^{s} X_{i,t} + \sqrt{\frac{\log 1/\delta}{2s}}\right] \geq 1-\delta$$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

After s pulls, arm i

$$\mathbb{P}\left[\mu_i \leq \hat{\mu}_{i,s} + \sqrt{\frac{\log 1/\delta}{2s}}\right] \geq 1 - \delta$$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

After s pulls, arm i

$$\mathbb{P}\left[\mu_i \leq \hat{\mu}_{i,s} + \sqrt{\frac{\log 1/\delta}{2s}}\right] \geq 1 - \delta$$

$\Rightarrow$ UCB uses an *upper confidence bound* on the expectation

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

Theorem

For any set of K arms with distributions bounded in $[0, b]$, if $\delta = 1/t$, then UCB(ρ) with $\rho > 1$, achieves a regret

$$R_n(\mathcal{A}) \leq \sum_{i \neq i^*} \left[\frac{4b^2}{\Delta_i} \rho \log(n) + \Delta_i \left(\frac{3}{2} + \frac{1}{2(\rho - 1)} \right) \right]$$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

Let $K = 2$ with $i^* = 1$

$$R_n(\mathcal{A}) \leq O\left(\frac{1}{\Delta}\rho \log(n)\right)$$

Remark 1: the *cumulative* regret slowly increases as $\log(n)$

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

Let $K = 2$ with $i^* = 1$

$$R_n(\mathcal{A}) \leq O\left(\frac{1}{\Delta}\rho \log(n)\right)$$

Remark 1: the *cumulative* regret slowly increases as $\log(n)$
Remark 2: the *smaller the gap* the *bigger the regret*... why?

The Upper–Confidence Bound (UCB) Algorithm (cont'd)

Show time (again)!

The Worst–case Performance

Remark: the regret bound is *distribution–dependent*

$$R_n(\mathcal{A};\Delta) \leq O\left(\frac{1}{\Delta}\rho\log(n)\right)$$

The Worst–case Performance

Remark: the regret bound is *distribution–dependent*

$$R_n(\mathcal{A}; \Delta) \leq O\left(\frac{1}{\Delta}\rho \log(n)\right)$$

Meaning: the algorithm is able to *adapt to the specific problem* at hand!

The Worst–case Performance

Remark: the regret bound is *distribution–dependent*

$$R_n(\mathcal{A}; \Delta) \leq O\left(\frac{1}{\Delta}\rho \log(n)\right)$$

Meaning: the algorithm is able to *adapt to the specific problem* at hand!

Worst–case performance: what is the distribution which leads to the worst possible performance of UCB? what is the distribution–free performance of UCB?

$$R_n(\mathcal{A}) = \sup_{\Delta} R_n(\mathcal{A}; \Delta)$$

The Worst–case Performance

Problem: it seems like if $\Delta \rightarrow 0$ then the regret tends to infinity...

The Worst–case Performance

Problem: it seems like if $\Delta \to 0$ then the regret tends to infinity...
... nosense because the regret is defined as

$$R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}]\Delta$$

The Worst–case Performance

Problem: it seems like if $\Delta \to 0$ then the regret tends to infinity...
... nosense because the regret is defined as

$$R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}]\Delta$$

then if Δ_i is small, the regret is also small...

The Worst–case Performance

Problem: it seems like if $\Delta \to 0$ then the regret tends to infinity...
... nosense because the regret is defined as

$$R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}]\Delta$$

then if Δ_i is small, the regret is also small...
In fact

$$R_n(\mathcal{A}; \Delta) = \min \left\{ O\left(\frac{1}{\Delta} \rho \log(n) \right), \mathbb{E}[T_{2,n}]\Delta \right\}$$

The Worst–case Performance

Then

$$R_n(\mathcal{A}) = \sup_{\Delta} R_n(\mathcal{A};\Delta) = \sup_{\Delta} \min\left\{ O\left(\frac{1}{\Delta}\rho\log(n)\right), n\Delta \right\} \approx \sqrt{n}$$

for $\Delta = \sqrt{1/n}$

Tuning the confidence δ of UCB

Remark: UCB is an *anytime* algorithm ($\delta = 1/t$)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log t}{2s}}$$

Tuning the confidence δ of UCB

Remark: UCB is an *anytime* algorithm ($\delta = 1/t$)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log t}{2s}}$$

Remark: If the time horizon n is known then the optimal choice is $\delta = 1/n$

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log n}{2s}}$$

Tuning the confidence δ of UCB (cont'd)

Intuition: UCB should pull the suboptimal arms

- *Enough*: so as to understand which arm is the best
- *Not too much*: so as to keep the regret as small as possible

Tuning the confidence δ of UCB (cont'd)

Intuition: UCB should pull the suboptimal arms

- *Enough*: so as to understand which arm is the best
- *Not too much*: so as to keep the regret as small as possible

The confidence $1-\delta$ has the following impact (similar for ρ)

- *Big* $1-\delta$: high level of *exploration*
- *Small* $1-\delta$: high level of *exploitation*

Tuning the confidence δ of UCB (cont'd)

Intuition: UCB should pull the suboptimal arms

- *Enough*: so as to understand which arm is the best
- *Not too much*: so as to keep the regret as small as possible

The confidence $1 - \delta$ has the following impact (similar for ρ)

- *Big* $1 - \delta$: high level of *exploration*
- *Small* $1 - \delta$: high level of *exploitation*

Solution: depending on the time horizon, we can tune how to trade-off between exploration and exploitation

UCB Proof

Let's dig into the (1 page and half!!) proof.

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

$$\mathcal{E} = \left\{ \forall i, s \ \ \left| \hat{\mu}_{i,s} - \mu_i \right| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

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

UCB Proof

Let's dig into the (1 page and half!!) proof.

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

$$\mathcal{E} = \left\{ \forall i, s \ \left| \hat{\mu}_{i,s} - \mu_i \right| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
At time t we pull arm i *[algorithm]*

$$B_{i,T_{i,t-1}} \geq B_{i^*,T_{i^*,t-1}}$$

UCB Proof

Let's dig into the (1 page and half!!) proof.

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

$$\mathcal{E} = \left\{ \forall i, s \quad \left| \hat{\mu}_{i,s} - \mu_i \right| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
At time t we pull arm i *[algorithm]*

$$\hat{\mu}_{i,T_{i,t-1}} + \sqrt{\frac{\log 1/\delta}{2T_{i,t-1}}} \geq \hat{\mu}_{i^*,T_{i^*,t-1}} + \sqrt{\frac{\log 1/\delta}{2T_{i^*,t-1}}}$$

UCB Proof

Let's dig into the (1 page and half!!) proof.

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

$$\mathcal{E} = \left\{ \forall i, s \ \left| \hat{\mu}_{i,s} - \mu_i \right| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nK\delta$.
At time t we pull arm i *[algorithm]*

$$\hat{\mu}_{i,T_{i,t-1}} + \sqrt{\frac{\log 1/\delta}{2T_{i,t-1}}} \geq \hat{\mu}_{i^*,T_{i^*,t-1}} + \sqrt{\frac{\log 1/\delta}{2T_{i^*,t-1}}}$$

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

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2T_{i,t-1}}} \geq \mu_{i^*}$$

UCB Proof (cont'd)

Assume t is the last time i is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$

UCB Proof (cont'd)

Assume t is the last time i is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$

Reordering *[math]*

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

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

UCB Proof (cont'd)

Assume t is the last time i is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$

Reordering *[math]*

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

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

$$\mathbb{E}[T_{i,n}] = \mathbb{E}[T_{i,n}\mathbb{I}\mathcal{E}] + \mathbb{E}[T_{i,n}\mathbb{I}\mathcal{E}^C]$$

UCB Proof (cont'd)

Assume t is the last time i is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$

Reordering *[math]*

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

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

$$\mathbb{E}[T_{i,n}] \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1 + n(nK\delta)$$

UCB Proof (cont'd)

Assume t is the last time i is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$

Reordering *[math]*

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

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

$$\mathbb{E}[T_{i,n}] \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1 + n(nK\delta)$$

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

$$\hat{\mu}_{i,T_{i,t-1}} + \sqrt{\frac{2\log n}{2T_{i,t-1}}}$$

UCB Proof (cont'd)

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

$$\hat{\mu}_{i,T_{i,t-1}} + \sqrt{\frac{2 \log n}{2 T_{i,t-1}}}$$

and

$$\mathbb{E}[T_{i,n}] \leq \frac{\log n}{\Delta_i^2} + 1 + K$$

Tuning the confidence δ of UCB (cont'd)

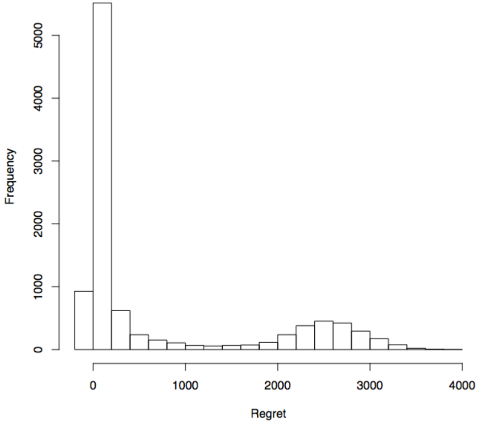
Multi–armed Bandit: the same for $\delta = 1/t$ and $\delta = 1/n$...

Tuning the confidence δ of UCB (cont'd)

Multi–armed Bandit: the same for $\delta = 1/t$ and $\delta = 1/n$...
... **almost** (i.e., in expectation)

Tuning the confidence δ of UCB (cont'd)

The value–at–risk of the regret for UCB-anytime



Tuning the ρ of UCB (cont'd)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log n}{2s}}$$

Tuning the ρ of UCB (cont'd)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log n}{2s}}$$

Theory

- $\rho < 0.5$, polynomial regret w.r.t. n
- $\rho > 0.5$, logarithmic regret w.r.t. n

Tuning the ρ of UCB (cont'd)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log n}{2s}}$$

Theory

- $\rho < 0.5$, polynomial regret w.r.t. n
- $\rho > 0.5$, logarithmic regret w.r.t. n

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

Tuning the ρ of UCB (cont'd)

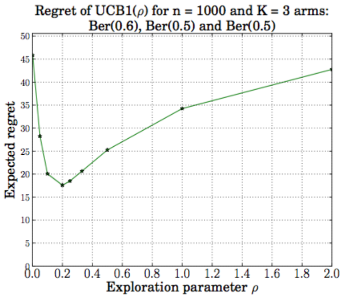
UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho\sqrt{\frac{\log n}{2s}}$$

Theory

- $\rho < 0.5$, polynomial regret w.r.t. n
- $\rho > 0.5$, 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

$$B_{i,t} = \hat{\mu}_{i,T_{i,t}} + \rho\sqrt{\frac{\log(t)}{2T_{i,t}}}$$

- Pull arm

$$I_t = \arg\max_{i=1,\dots,K} B_{i,t}$$

- Update the number of pulls $T_{I_t,t}$, $\hat{\mu}_{i,T_{i,t}}$

Improvements: UCB-V

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

Algorithm

- Compute the *score* of each arm i

$$B_{i,t} = \hat{\mu}_{i,T_{i,t}} + \sqrt{\frac{2\hat{\sigma}^2_{i,T_{i,t}} \log t}{T_{i,t}}} + \frac{8 \log t}{3 T_{i,t}}$$

- Pull arm

$$I_t = \arg \max_{i=1,\dots,K} B_{i,t}$$

- Update the number of pulls $T_{I_t,t}$, $\hat{\mu}_{i,T_{i,t}}$ and $\hat{\sigma}^2_{i,T_{i,t}}$

Improvements: UCB-V

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

Algorithm

- Compute the *score* of each arm i

$$B_{i,t} = \hat{\mu}_{i,T_{i,t}} + \sqrt{\frac{2\hat{\sigma}^2_{i,T_{i,t}} \log t}{T_{i,t}}} + \frac{8 \log t}{3 T_{i,t}}$$

- Pull arm

$$I_t = \arg\max_{i=1,\dots,K} B_{i,t}$$

- Update the number of pulls $T_{I_t,t}$, $\hat{\mu}_{i,T_{i,t}}$ and $\hat{\sigma}^2_{i,T_{i,t}}$

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

$$B_{i,t} = \hat{\mu}_{i,T_{i,t}} + \sqrt{\frac{2\hat{\sigma}^2_{i,T_{i,t}} \log t}{T_{i,t}}} + \frac{8 \log t}{3 T_{i,t}}$$

- Pull arm

$$I_t = \arg \max_{i=1,\ldots,K} B_{i,t}$$

- Update the number of pulls $T_{I_t,t}$, $\hat{\mu}_{i,T_{i,t}}$ and $\hat{\sigma}^2_{i,T_{i,t}}$

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*

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

Improvements: KL-UCB

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

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

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

$$B_{i,t} = \max \Big\{ q \in [0,1] : T_{i,t} d\big(\hat{\mu}_{i,T_{i,t}}, q\big) \leq \log(t) + c \log(\log(t)) \Big\}$$

Improvements: KL-UCB

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

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

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

$$B_{i,t} = \max \Big\{ q \in [0,1] : T_{i,t} d\big(\hat{\mu}_{i,T_{i,t}}, q\big) \leq \log(t) + c \log(\log(t)) \Big\}$$

Regret: pulls to suboptimal arms

$$\mathbb{E}\big[T_{i,n}\big] \leq (1+\epsilon) \frac{\log(n)}{d(\mu_i, \mu^*)} + C_1 \log(\log(n)) + \frac{C_2(\epsilon)}{n^{\beta(\epsilon)}}$$

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

Improvements: Thompson strategy

Idea: Use a Bayesian approach to estimate the means $\{\mu_i\}_i$

Improvements: Thompson strategy

Idea: Use a Bayesian approach to estimate the means $\{\mu_i\}_i$

Algorithm: Assuming Bernoulli arms and a *Beta* prior on the mean

- Compute
$$\mathcal{D}_{i,t} = \mathrm{Beta}(S_{i,t} + 1, F_{i,t} + 1)$$
- Draw a mean sample as
$$\widetilde{\mu}_{i,t} \sim \mathcal{D}_{i,t}$$
- Pull arm
$$I_t = \arg\max \widetilde{\mu}_{i,t}$$
- If $X_{I_t,t} = 1$ update $S_{I_t,t+1} = S_{I_t,t} + 1$, else update $F_{I_t,t+1} = F_{I_t,t} + 1$

Regret:
$$\lim_{n\to\infty} \frac{R_n}{\log(n)} = \sum_{i=1}^{K} \frac{\Delta_i}{d(\mu_i, \mu^*)}$$

The Lower Bound

Theorem

For any stochastic bandit $\{\nu_i\}$, any algorithm $\mathcal{A}$ has a regret

$$\lim_{n\to\infty} \frac{R_n}{\log n} \geq \frac{\Delta_i}{\inf_\nu KL(\nu_i, \nu)}$$

The Lower Bound

Theorem

For any stochastic bandit $\{\nu_i\}$, any algorithm $\mathcal{A}$ has a regret

$$\lim_{n\to\infty} \frac{R_n}{\log n} \geq \frac{\Delta_i}{\inf_\nu KL(\nu_i, \nu)}$$

Problem: this is just asymptotic

The Lower Bound

Theorem

For any stochastic bandit $\{\nu_i\}$, *any algorithm* $\mathcal{A}$ *has a regret*

$$\lim_{n\to\infty} \frac{R_n}{\log n} \geq \frac{\Delta_i}{\inf_\nu KL(\nu_i, \nu)}$$

Problem: this is just asymptotic

Open Question: what is the finite-time lower bound?

The Exploration-Exploitation Dilemma

Tools

Stochastic Multi-Armed Bandit

Contextual Linear Bandit

Other Multi-Armed Bandit Problems

The Contextual Linear Bandit Problem

Motivating Example: news recommendation

- Different users may have different preferences
- Different news may have different characteristics
- The set of available news may change over time
- We want to minimise the regret w.r.t. the best news for each user

The *Linear* Bandit Problem

Limitations of MAB:

- Arms are independent
- Each single arm has to be tested at least once
- Regret scales linearly with K

The *Linear* Bandit Problem

Limitations of MAB:

- Arms are independent
- Each single arm has to be tested at least once
- Regret scales linearly with K

Linear bandit approach:

- Embed arms in $\mathbb{R}^d$ (each arm a is mapped to a feature vector $\phi_a \in \mathbb{R}^d$)
- The reward varies *linearly* with the arm

$$\mathbb{E}[r(a)] = \phi_a^\top \theta^*$$

where $\theta^* \in \mathbb{R}^d$ is unknown.

The *Linear* Bandit Problem

Limitations of MAB:

- Arms are independent
- Each single arm has to be tested at least once
- Regret scales linearly with K

Linear bandit approach:

- Embed arms in $\mathbb{R}^d$ (each arm a is mapped to a feature vector $\phi_a \in \mathbb{R}^d$)
- The reward varies *linearly* with the arm

$$\mathbb{E}[r(a)] = \phi_a^\top \theta^*$$

where $\theta^* \in \mathbb{R}^d$ is unknown.

Remark: if $d = A$ and $\phi_a = e_a$, then it coincides with MAB

The Linear Bandit Problem

The problem: at each time $t = 1, \ldots, n$

- The learner chooses an arm a_t and receives a reward r_{a_t}

The optimal arm: $a^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}[r(a)] = \arg\max_{a \in \mathcal{A}} \phi_a^\top \theta^*$

The regret:

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

The Linear Bandit Problem

The MAB approach: the value of an arm is estimated by $\widehat{\mu}_{i,t}$

Exploiting the linear assumption:

- Estimate θ^* using regularized least squares

$$\widehat{\theta}_n = \arg\min_{\theta} \sum_{t=1}^{n} \Big(\phi_{a_t}^\top \theta - r_t(a_t)\Big)^2 + \lambda \|\theta\|_2^2$$

The Linear Bandit Problem

The MAB approach: the value of an arm is estimated by $\widehat{\mu}_{i,t}$

Exploiting the linear assumption:

- Estimate θ^* using regularized least squares

$$\widehat{\theta}_n = \arg\min_{\theta} \sum_{t=1}^{n} \Big(\phi_{a_t}^\top \theta - r_t(a_t)\Big)^2 + \lambda \|\theta\|_2^2$$

- Closed-form solution

$$A_n = \sum_{t=1}^{n} \phi_{a_t}\phi_{a_t}^\top + \lambda I \quad b_n = \sum_{t=1}^{n} \phi_{a_t} r_t(a_t)$$

$$\Rightarrow \widehat{\theta}_n = A_n^{-1} b_n$$

The Linear Bandit Problem

The MAB approach: the value of an arm is estimated by $\widehat{\mu}_{i,t}$

Exploiting the linear assumption:

- Estimate θ^* using regularized least squares

$$\widehat{\theta}_n = \arg\min_\theta \sum_{t=1}^n \Big(\phi_{a_t}^\top \theta - r_t(a_t)\Big)^2 + \lambda \|\theta\|_2^2$$

- Closed-form solution

$$A_n = \sum_{t=1}^n \phi_{a_t}\phi_{a_t}^\top + \lambda I \quad b_n = \sum_{t=1}^n \phi_{a_t} r_t(a_t)$$

$$\Rightarrow \widehat{\theta}_n = A_n^{-1} b_n$$

- Estimate of the value of arm a

$$\widehat{r}_n(a) = \phi_a^\top \widehat{\theta}_n$$

The Linear Bandit Problem

The MAB approach: construct confidence intervals $\sqrt{\log(1/\delta)/T_{i,n}}$

Exploiting the linear assumption:

- Estimate of an arm $\widehat{r}_n(a)$ may be accurate when "similar" arms have been selected (even if $T_n(a) = 0$!)

The Linear Bandit Problem

The MAB approach: construct confidence intervals $\sqrt{\log(1/\delta)/T_{i,n}}$

Exploiting the linear assumption:

- Estimate of an arm $\widehat{r}_n(a)$ may be accurate when "similar" arms have been selected (even if $T_n(a) = 0$!)
- Confidence intervals

$$\big|r(a) - \widehat{r}_n(a)\big| \leq \alpha_n \sqrt{\phi_a^\top A_n^{-1} \phi_a}$$

The Linear Bandit Problem

The MAB approach: construct confidence intervals $\sqrt{\log(1/\delta)/T_{i,n}}$

Exploiting the linear assumption:

- Estimate of an arm $\widehat{r}_n(a)$ may be accurate when "similar" arms have been selected (even if $T_n(a) = 0$!)
- Confidence intervals

$$\big|r(a) - \widehat{r}_n(a)\big| \leq \alpha_n \sqrt{\phi_a^\top A_n^{-1} \phi_a}$$

- Tuning of the confidence interval

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

Remark: the confidence interval reduces to MAB when all arms are orthogonal

The Linear Bandit Problem

The MAB approach – UCB: pull arm $I_t = \widehat{\mu}_{i,t} + \sqrt{\log(1/\delta)/T_{i,t}}$

Exploiting the linear assumption:

- At each time step t select arm

$$a_t = \arg\max_{a \in A} \phi_a^\top \widehat{\theta}_t + \alpha_t \sqrt{\phi_a^\top A_t^{-1} \phi_a}$$

The Linear Bandit Problem

The MAB approach – UCB: regret $O(K\log(n)/\Delta)$ or $O(\sqrt{Kn\log(K)})$

Exploiting the linear assumption:

- Regret bound

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

The Linear Bandit Problem

The MAB approach – TS:

- Compute a posterior over μ_i
- Draw a $\widetilde{\mu}_i$ from the posterior
- Select arm $I_t = \arg\max_i \widetilde{\mu}_i$

Exploiting the linear assumption:

- Regret bound

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

The *Contextual Linear* Bandit Problem

Limitations of MAB:

- The value of an arm is fixed
- No side-information / context is used

The *Contextual Linear* Bandit Problem

Limitations of MAB:

- The value of an arm is fixed
- No side-information / context is used

Contextual linear bandit approach:

- Finite arms
- Define a context $x \in \mathcal{X}$
- The reward varies *linearly* with the context

$$\mathbb{E}[r(x,a)] = \phi_x^\top \theta_a^*$$

The *Contextual Linear* Bandit Problem

Limitations of MAB:

- The value of an arm is fixed
- No side-information / context is used

Contextual linear bandit approach:

- Finite arms
- Define a context $x \in \mathcal{X}$
- The reward varies *linearly* with the context

$$\mathbb{E}[r(x,a)] = \phi_x^\top \theta_a^*$$

Extensions:

- Embed arms in $\mathbb{R}^d$ and

$$\mathbb{E}[r(x,a)] = \phi_{x,a}^\top \theta_a^*$$

- Let the arm set change over time $\mathcal{A}_t$

The Contextual Linear Bandit Problem

The problem: at each time $t = 1, \dots, n$

- User x_t arrives and a set of news $\mathcal{A}_t$ is provided
- The user x_t together with a news $a \in \mathcal{A}_t$ are described by a feature vector $\phi_{x_t,a}$
- The learner chooses a news $a_t \in \mathcal{A}_t$ and receives a reward $r_t(x_t, a_t)$

The optimal news: at each time $t = 1, \dots, n$, the optimal news is

$$a_t^* = \arg\max_{a \in \mathcal{A}_t} \mathbb{E}[r_t(x_t, a_t)]$$

The regret:

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

The *Contextual Linear* Bandit Problem

The linear regression estimate:

- $\mathcal{T}_a = \{t : a_t = a\}$
- Construct the design matrix of all the contexts observed when action a has been taken $D_a \in \mathbb{R}^{|\mathcal{T}_a| \times d}$
- Construct the reward vector of all the rewards observed when action a has been taken $c_a \in \mathbb{R}^{|\mathcal{T}_a|}$
- Estimate θ_a as

$$\hat{\theta}_a = (D_a^\top D_a + I)^{-1} D_a^\top c_a$$

The Contextual Linear Bandit Problem

Optimism in face of uncertainty: the LinUCB algorithm

- Chernoff-Hoeffding in this case becomes

$$\left|\phi_{x,a}^\top \hat{\theta}_a - r(x,a)]\right| \leq \alpha \sqrt{\phi_{x,a}^\top (D_a^\top D_a + I)^{-1} \phi_{x,a}}$$

- and the UCB strategy is

$$a_t = \arg\max_{a \in \mathcal{A}_t} \phi_{x,a}^\top \hat{\theta}_a + \alpha \sqrt{\phi_{x,a}^\top (D_a^\top D_a + I)^{-1} \phi_{x,a}}$$

The Contextual Linear Bandit Problem

The evaluation problem

- Online evaluation: too expensive
- Offline evaluation: how to use the logged data?

The Contextual Linear Bandit Problem

Evaluation from logged data

- Assumption 1: contexts and rewards are i.i.d. from a stationary distribution

$$(x_1, \dots, x_K, r_1, \dots, r_K) \sim D$$

- Assumption 2: the logging strategy is random

The Contextual Linear Bandit Problem

Evaluation from logged data: given a bandit strategy π, a desired number of samples T, and a (infinite) stream of data

Algorithm 3 Policy_Evaluator.

Inputs: $T > 0$; policy π; stream of events
$h_0 \leftarrow \emptyset$ {An initially empty history}
$R_0 \leftarrow 0$ {An initially zero total payoff}
for $t = 1, 2, 3, \ldots, T$ **do**
  repeat
    Get next event $(\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a)$
  until $\pi(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K)) = a$
  $h_t \leftarrow \text{CONCATENATE}(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a))$
  $R_t \leftarrow R_{t-1} + r_a$
end for
Output: R_T / T

The Exploration-Exploitation Dilemma

Tools

Stochastic Multi-Armed Bandit

Contextual Linear Bandit

Other Multi-Armed Bandit Problems

The Exploration-Exploitation Dilemma

Tools

Stochastic Multi-Armed Bandit

Contextual Linear Bandit

Other Multi-Armed Bandit Problems

The Best Arm Identification Problem

Motivating Examples

- Find the best shortest path in a limited number of days
- Maximize the confidence about the best treatment after a finite number of patients
- Discover the best advertisements after a training phase
- ...

The Best Arm Identification Problem

Objective: given a fixed budget n, return the best arm $i^* = \arg\max_i \mu_i$ at the end of the experiment

The Best Arm Identification Problem

Objective: given a fixed budget n, return the best arm $i^* = \arg\max_i \mu_i$ at the end of the experiment
Measure of performance: the probability of error

$$\mathbb{P}[J_n \neq i^*] \leq \sum_{i=1}^{N} \exp\big(- T_{i,n}\Delta_i^2\big)$$

The Best Arm Identification Problem

Objective: given a fixed budget n, return the best arm $i^* = \arg\max_i \mu_i$ at the end of the experiment
Measure of performance: the probability of error

$$\mathbb{P}[J_n \neq i^*] \leq \sum_{i=1}^{N} \exp\big(- T_{i,n}\Delta_i^2\big)$$

Algorithm idea: mimic the behavior of the optimal strategy

$$T_{i,n} = \frac{\frac{1}{\Delta_i^2}}{\sum_{j=1}^{N} \frac{1}{\Delta_j^2}} n$$

The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N-1$ phases. Define $(\overline{\log}(N) = 0.5 + \sum_{i=2}^{N} 1/i)$

$$n_k = \frac{1}{\overline{\log} K} \frac{n-N}{N+1-k}$$

The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N-1$ phases. Define $(\overline{\log}(N) = 0.5 + \sum_{i=2}^{N} 1/i)$

$$n_k = \frac{1}{\overline{\log} K} \frac{n-N}{N+1-k}$$

- Set of active arms A_k at phase k $(A_1 = \{1, \ldots, N\})$

The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N - 1$ phases. Define $(\overline{\log}(N) = 0.5 + \sum_{i=2}^{N} 1/i)$

$$n_k = \frac{1}{\overline{\log} K} \frac{n - N}{N + 1 - k}$$

- Set of active arms A_k at phase k ($A_1 = \{1, \ldots, N\}$)
- For each phase $k = 1, \ldots, N - 1$
  - For each arm $i \in A_k$, pull arm i for $n_k - n_{k-1}$ rounds

The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N-1$ phases. Define $(\overline{\log}(N) = 0.5 + \sum_{i=2}^{N} 1/i)$

$$n_k = \frac{1}{\overline{\log} K} \frac{n-N}{N+1-k}$$

- Set of active arms A_k at phase k ($A_1 = \{1, \ldots, N\}$)
- For each phase $k = 1, \ldots, N-1$
  - For each arm $i \in A_k$, pull arm i for $n_k - n_{k-1}$ rounds
  - Remove the worst arm

$$A_{k+1} = A_k \backslash \arg\min_{i \in A_k} \hat{\mu}_{i,n_k}$$

The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N-1$ phases. Define $(\overline{\log}(N) = 0.5 + \sum_{i=2}^{N} 1/i)$

$$n_k = \frac{1}{\overline{\log} K} \frac{n-N}{N+1-k}$$

- Set of active arms A_k at phase k ($A_1 = \{1, \ldots, N\}$)
- For each phase $k = 1, \ldots, N-1$
  - For each arm $i \in A_k$, pull arm i for $n_k - n_{k-1}$ rounds
  - Remove the worst arm

$$A_{k+1} = A_k \setminus \arg\min_{i \in A_k} \hat{\mu}_{i,n_k}$$

- Return the only remaining arm $J_n = A_N$

The Best Arm Identification Problem

The Successive Reject Algorithm

Theorem

The successive reject algorithm have a probability of doing a mistake of

$$\mathbb{P}[J_n \neq i^*] \leq \frac{K(K-1)}{2} \exp\Big(- \frac{n-N}{\overline{\log} N H_2} \Big)$$

with $H_2 = \max_{i=1,\dots,N} i \Delta_{(i)}^{-2}$.

The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter a
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$

The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter a
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$

- Select

$$I_t = \arg\max_{B_{i,s}}$$

The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter a
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$

- Select

$$I_t = \arg\max_{B_{i,s}}$$

- At the end return

$$J_n = \arg\max_i \hat{\mu}_{i,T_{i,n}}$$

The Best Arm Identification Problem

The UCB-E Algorithm

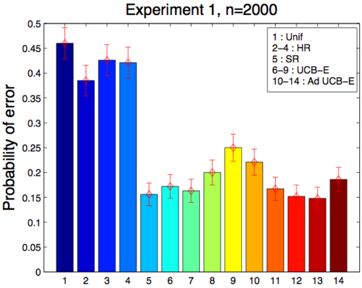
Theorem

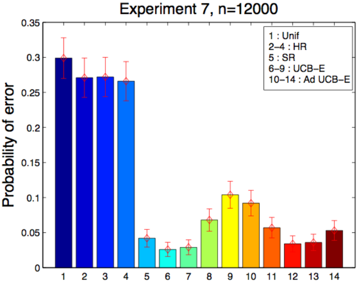
The UCB-E algorithm with $a = \frac{25}{36}\frac{n-N}{H_1}$ *has a probability of doing a mistake of*

$$\mathbb{P}[J_n \neq i^*] \leq 2nN \exp\Big(-\frac{2a}{25}\Big)$$

with $H_1 = \sum_{i=1}^{N} 1/\Delta_i^2$.

The Best Arm Identification Problem





The Active Bandit Problem

Motivating Examples

- N production lines
- The test of the performance of a line is expensive
- We want an accurate estimation of the performance of each production line

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

Notice: Given an arm has a mean μ_i and a variance σ_i^2, if it is pulled $T_{i,n}$ times, then

$$L_{i,n} = \mathbb{E}\big[(\hat{\mu}_{i,T_{i,n}} - \mu_i)^2\big] = \frac{\sigma_i^2}{T_{i,n}}$$

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

Notice: Given an arm has a mean μ_i and a variance σ_i^2, if it is pulled $T_{i,n}$ times, then

$$L_{i,n} = \mathbb{E}\big[(\hat{\mu}_{i,T_{i,n}} - \mu_i)^2\big] = \frac{\sigma_i^2}{T_{i,n}}$$

$$L_n = \max_i L_{i,n}$$

The Active Bandit Problem

Problem: what are the number of pulls $(T_{1,n}, \ldots, T_{N,n})$ (such that $\sum T_{i,n} = n$) which minimizes the loss?

$$(T^*_{1,n}, \ldots, T^*_{N,n}) = \arg \min_{(T_{1,n},\ldots,T_{N,n})} L_n$$

The Active Bandit Problem

Problem: what are the number of pulls $(T_{1,n}, \ldots, T_{N,n})$ (such that $\sum T_{i,n} = n$) which minimizes the loss?

$$(T^*_{1,n}, \ldots, T^*_{N,n}) = \arg \min_{(T_{1,n}, \ldots, T_{N,n})} L_n$$

Answer

$$T^*_{i,n} = \frac{\sigma_i^2}{\sum_{j=1}^N \sigma_j^2} n$$

The Active Bandit Problem

Problem: what are the number of pulls $(T_{1,n}, \ldots, T_{N,n})$ (such that $\sum T_{i,n} = n$) which minimizes the loss?

$$(T^*_{1,n}, \ldots, T^*_{N,n}) = \arg \min_{(T_{1,n},\ldots,T_{N,n})} L_n$$

Answer

$$T^*_{i,n} = \frac{\sigma_i^2}{\sum_{j=1}^N \sigma_j^2} n$$

$$L^*_n = \frac{\sum_{i=1}^N \sigma_i^2}{n} = \frac{\Sigma}{n}$$

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms
Measure of performance: the regret on the quadratic error

$$R_n(\mathcal{A}) = \max_i L_n(\mathcal{A}) - \frac{\sum_{i=1}^N \sigma_i^2}{n}$$

The Active Bandit Problem

Objective: given a fixed budget n, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms
Measure of performance: the regret on the quadratic error

$$R_n(\mathcal{A}) = \max_i L_n(\mathcal{A}) - \frac{\sum_{i=1}^N \sigma_i^2}{n}$$

Algorithm idea: mimic the behavior of the optimal strategy

$$T_{i,n} = \frac{\sigma_i^2}{\sum_{j=1}^N \sigma_j^2} n = \lambda_i n$$

The Active Bandit Problem

An UCB–based strategy

At each time step $t = 1, \ldots, n$

- Estimate

$$\hat{\sigma}^2_{i,T_{i,t-1}} = \frac{1}{T_{i,t-1}} \sum_{s=1}^{T_{i,t-1}} X^2_{s,i} - \hat{\mu}^2_{i,T_{i,t-1}}$$

- Compute

$$B_{i,t} = \frac{1}{T_{i,t-1}} \left(\hat{\sigma}^2_{i,T_{i,t-1}} + 5\sqrt{\frac{\log 1/\delta}{2\,T_{i,t-1}}} \right)$$

- Pull arm

$$I_t = \arg\max B_{i,t}$$

The Active Bandit Problem

Theorem

The UCB–based algorithm achieves a regret

$$R_n(\mathcal{A}) \leq \frac{98 \log(n)}{n^{3/2} \lambda_{\min}^{5/2}} + O\left(\frac{\log n}{n^2}\right)$$

The Active Bandit Problem

Theorem

The UCB–based algorithm achieves a regret

$$R_n(\mathcal{A}) \leq \frac{98 \log(n)}{n^{3/2} \lambda_{\min}^{5/2}} + O\left(\frac{\log n}{n^2}\right)$$

The Exploration-Exploitation Dilemma

Tools

Stochastic Multi-Armed Bandit

Contextual Linear Bandit

Other Multi-Armed Bandit Problems

Bonus: Reinforcement Learning

Learning the Optimal Policy

For $i = 1, \ldots, n$

1. Set $t = 0$
2. Set initial state x_0
3. **While** (x_t not terminal)
   3.1 Take action a_t ***according to a suitable exploration policy***
   3.2 Observe next state x_{t+1} and reward r_t
   3.3 Compute the temporal difference δ_t (e.g., Q-learning)
   3.4 Update the Q-function

$$\widehat{Q}(x_t, a_t) = \widehat{Q}(x_t, a_t) + \alpha(x_t, a_t)\delta_t$$

   3.5 Set $t = t + 1$
   EndWhile

EndFor

Learning the Optimal Policy

The regret in MAB

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

Learning the Optimal Policy

The regret in MAB

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

$$\Rightarrow R_n(\mathcal{A}) = \max_{\pi} \mathbb{E}\Big[\sum_{t=1}^{n} r(x_t, \pi(x_t))\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r(x_t, a_t)\Big]$$

Learning the Optimal Policy

The regret in MAB

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} X_{I_t,t}\Big]$$

$$\Rightarrow R_n(\mathcal{A}) = \max_{\pi} \mathbb{E}\Big[\sum_{t=1}^{n} r(x_t, \pi(x_t))\Big] - \mathbb{E}\Big[\sum_{t=1}^{n} r(x_t, a_t)\Big]$$

⇒ ***not correct***: actions influence the state as well!

Learning the Optimal Policy

The regret in MAB

$$R_n(\mathcal{A}) = \max_{i=1,\dots,K} \mathbb{E}\Big[\sum_{t=1}^n X_{i,t}\Big] - \mathbb{E}\Big[\sum_{t=1}^n X_{I_t,t}\Big]$$

$$\Rightarrow R_n(\mathcal{A}) = \max_{\pi} \mathbb{E}\Big[\sum_{t=1}^n r(x_t, \pi(x_t))\Big] - \mathbb{E}\Big[\sum_{t=1}^n r(x_t, a_t)\Big]$$

$\Rightarrow$ ***not correct***: actions influence the state as well!

The regret in RL

$$R_n(\mathcal{A}) = \max_{\pi} \mathbb{E}\Big[\sum_{t=1}^n r(x_t^*, \pi(x_t^*))\Big] - \mathbb{E}\Big[\sum_{t=1}^n r(x_t, a_t)\Big],$$

$x_t^* \sim p\big(\cdot\,|x_{t-1}^*, \pi^*(x_{t-1}^*)\big)$

Learning the Optimal Policy

Idea: can we adapt UCB (that already works in MAB, contextual bandit) here?

Learning the Optimal Policy

Idea: can we adapt UCB (that already works in MAB, contextual bandit) here? ***Yes!***

Exploration-Exploitation in RL

- A policy π is defined as $\pi : X \to A$
- The long-term average reward of a policy is

$$\rho_\pi(M) = \lim_{n\to\infty} \mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n} r_t\right]$$

- Optimal policy

$$\pi^*(M) = \arg\max_\pi \rho_\pi(M) \implies \rho^*(M) = \rho_{\pi^*(M)}(M)$$

Exploration-Exploitation in RL

- A policy π is defined as $\pi : X \rightarrow A$
- The long-term average reward of a policy is

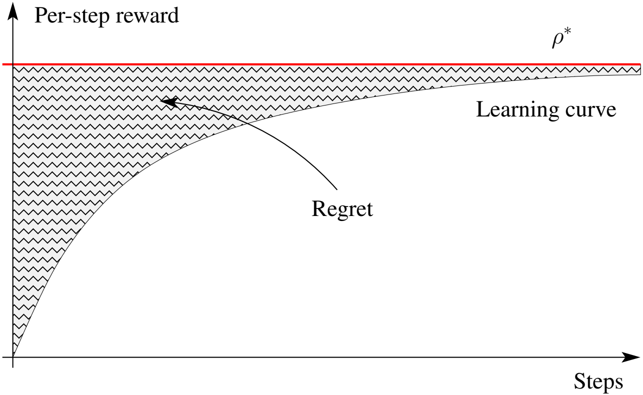
$$\rho_\pi(M) = \lim_{n\rightarrow\infty} \mathbb{E}\Big[\frac{1}{n}\sum_{t=1}^{n} r_t\Big]$$

- Optimal policy

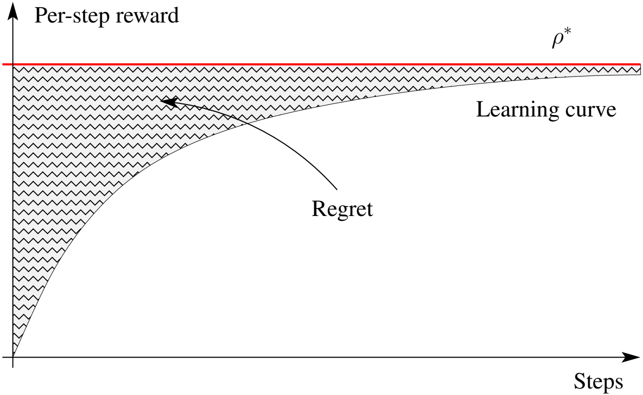
$$\pi^*(M) = \arg\max_\pi \rho_\pi(M) \implies \rho^*(M) = \rho_{\pi^*(M)}(M)$$

- Exploration-exploitation dilemma
  - *Explore* the environment to estimate its parameters
  - *Exploit* the estimates to collect reward

Exploration-Exploitation in RL

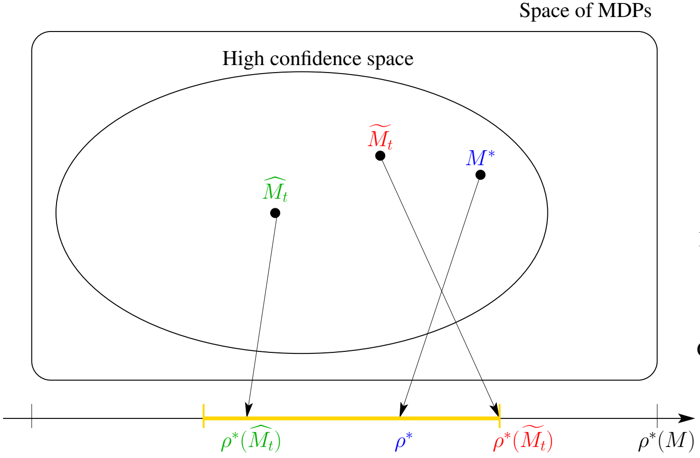


Exploration-Exploitation in RL



$$\textit{Cumulative Regret} \qquad R_n = n\rho^* - \sum_{t=1}^{n} r_t$$

Upper-confidence Bound for RL (UCRL)

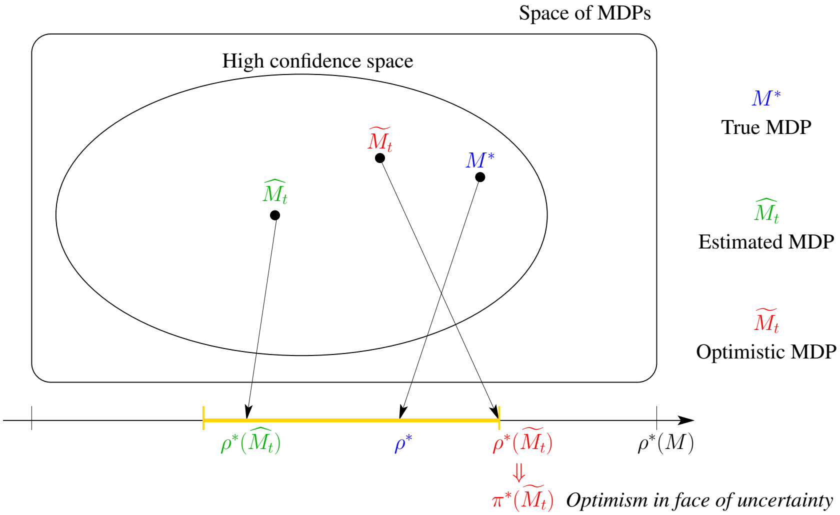


M^*
True MDP

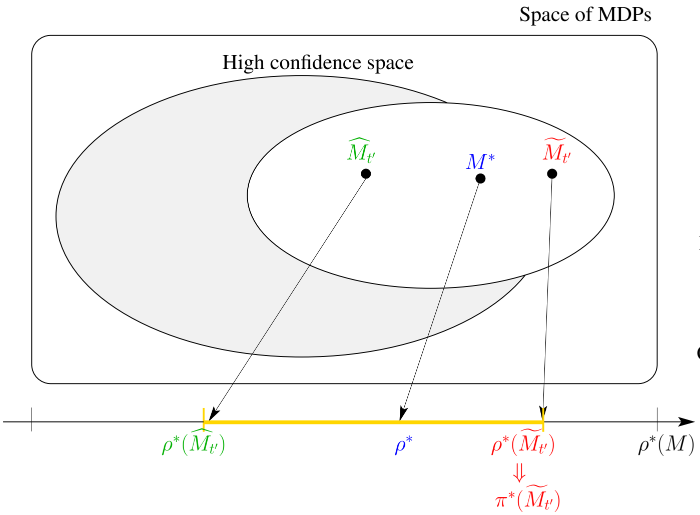
$\widehat{M}_t$
Estimated MDP

$\widetilde{M}_t$
Optimistic MDP

Upper-confidence Bound for RL (UCRL)



Upper-confidence Bound for RL (UCRL)

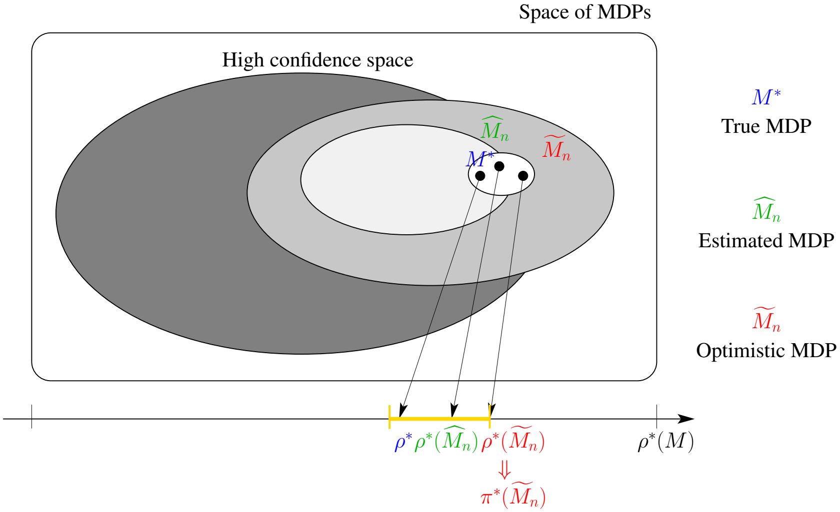


M^*
True MDP

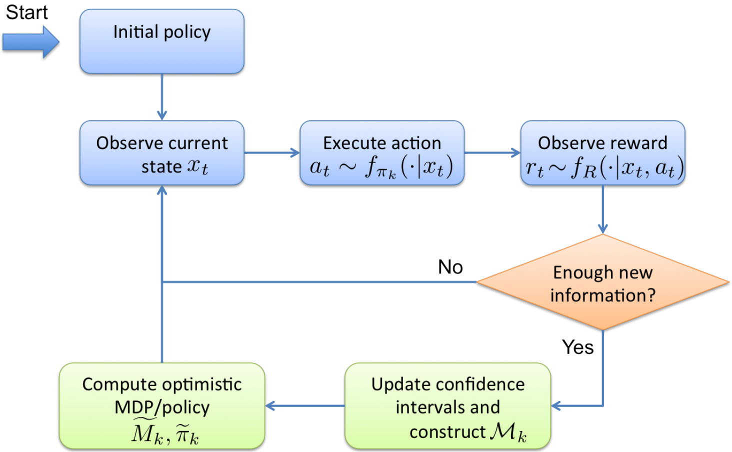
$\widehat{M}_{t'}$
Estimated MDP

$\widetilde{M}_{t'}$
Optimistic MDP

Upper-confidence Bound for RL (UCRL)



Upper-confidence Bound for RL (UCRL)



The UCRL2 Algorithm

Initialize episode k

1. Current time t_k
2. Let $N_k(x,a) = \big|\{\tau < t_k : x_t = x, a_t = a\}\big|$
3. Let $R_k(x,a) = \sum_{t=1}^{t_k} r_t \mathbb{I}\{x_t = x, a_t = a\}$
4. Let $P_k(x,a,x') = \big|\{\tau < t_k : x_t = x, a_t = a, x_{t+1} = x'\}\big|$
5. Compute $\hat{r}_k(x,a) = \frac{R_k(x,a)}{N_k(x,a)}$, $\hat{p}_k(x,a,x') = \frac{P_k(x,a,x')}{N_k(x,a)}$

Compute optimistic policy

1. Let

$$\mathcal{M}_k = \Big\{\widetilde{M} : |\tilde{r}(x,a) - \hat{r}_k(x,a)| \leq B_r(x,a); \\ \|\tilde{p}(\cdot|x,a) - \hat{p}_k(\cdot|x,a)\|_1 \leq B_p(x,a)\Big\}$$

2. Compute

$$\tilde{\pi}_k = \arg\max_{\pi} \max_{\tilde{M} \in \mathcal{M}_k} \rho(\pi; \tilde{M})$$

Execute $\tilde{\pi}_k$ **until at least one state-action space counter is doubled**

Upper-confidence Bound for RL (UCRL)

Set of *plausible MDPs* $\mathcal{M}_k = \{\widetilde{M}\}$: confidence intervals built using Chernoff bounds

$$B_r(x,a) \approx \sqrt{\frac{\log(XA/\delta)}{N_k(x,a)}}; \quad B_p(x,a) \approx \sqrt{\frac{X\log(XA/\delta)}{N_k(x,a)}}$$

Upper-confidence Bound for RL (UCRL)

Set of *plausible MDPs* $\mathcal{M}_k = \{\widetilde{M}\}$: confidence intervals built using Chernoff bounds

$$B_r(x,a) \approx \sqrt{\frac{\log(XA/\delta)}{N_k(x,a)}}; \quad B_p(x,a) \approx \sqrt{\frac{X\log(XA/\delta)}{N_k(x,a)}}$$

Computation of the *optimistic optimal policy* $\widetilde{\pi}_k$

$$\widetilde{\pi}_k = \arg\max_{\pi} \max_{\widetilde{M}\in\mathcal{M}_k} \rho_\pi(\widetilde{M})$$

The Extended Value Iteration Algorithm

Planning in average reward MDPs

- The optimal Bellman equation: optimal gain ρ^* and bias u^*

$$u^*(x) + \rho^* = \max_a \Big[r(x,a) + \sum_{x'} p(x'|x,a) u^*(x') \Big]$$

- Value iteration (given v_0)

$$v_n = \max_a \Big[r(x,a) + \sum_{x'} p(x'|x,a) v_{n-1}(x') \Big]$$

until $\mathrm{span}(v_n - v_{n-1}) \leq \epsilon$

- Guarantees of greedy policy

$$\pi_n(x) = \arg\max_a \Big[r(x,a) + \sum_{x'} p(x'|x,a) v_{n-1}(x') \Big] \Rightarrow |g^{\pi_n} - g^*| \leq \epsilon$$

The Extended Value Iteration Algorithm

Planning in optimistic average reward MDPs

- The optimal Bellman equation: optimal gain $\widetilde{\rho}$ and bias $\widetilde{u}$

$$\widetilde{u}(x) + \widetilde{\rho} = \max_a \max_{\tilde{r}(x,a)} \max_{\tilde{p}(\cdot|x,a)} \Big[\tilde{r}(x,a) + \sum_{x'} \tilde{p}(x'|x,a)\widetilde{u}(x')\Big]$$

- Value iteration (given v_0)

$$\begin{aligned} v_n &= \max_a \max_{\tilde{r}(x,a)} \max_{\tilde{p}(\cdot|x,a)} \Big[\tilde{r}(x,a) + \sum_{x'} \tilde{p}(x'|x,a) v_{n-1}(x')\Big] \\ &= \max_a \max_{\tilde{p}(\cdot|x,a)} \Big[\tilde{r}^+(x,a) + \sum_{x'} \tilde{p}(x'|x,a) v_{n-1}(x')\Big] \qquad (\tilde{r}^+ = \hat{r} + \sqrt{1/N_k}) \\ &= \max_a \Big[\tilde{r}^+(x,a) + \max_{\tilde{p}(\cdot|x,a)} \sum_{x'} \tilde{p}(x'|x,a) v_{n-1}(x')\Big] \quad \text{(simple LP)} \end{aligned}$$

- LP problem: assign highest probability from $\|\tilde{p}(\cdot|x,a) - \hat{p}(\cdot|x,a)\|_1$ to highest $v_{n-1}(x')$

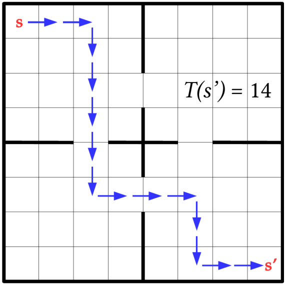
The Regret

Theorem

UCRL2 *run over n steps in an MDP with diameter D, X states and A actions suffers a regret*

$$R_n = O(DX\sqrt{An})$$

where diameter $D = \max_{x,x'} \min_{\pi} \mathbb{E}\big[T_{\pi}(x,x')\big]$.



Posterior Sampling for Reinforcement Learning (PSRL)

Initialize episode k

1. Current time t_k
2. Let $N_k(x,a) = \big|\{\tau < t_k : x_t = x, a_t = a\}\big|$
3. Compute posterior over $r(x,a)$ and $p(\cdot|x,a)$

Compute random policy

1. Let $\widetilde{M}_k = \{\widetilde{r}_k, \widetilde{p}_k\}$ such that $\widetilde{r}_k, \widetilde{p}_k$ sampled from their posteriors
2. Compute optimal policy $\tilde{\pi}_k = \arg\max_\pi \rho^\pi(\widetilde{M}_k)$

Execute $\tilde{\pi}_k$ until at least one state-action space counter is doubled

Bibliography I

Reinforcement Learning

Alessandro Lazaric
alessandro.lazaric@inria.fr
sequel.lille.inria.fr