3 Multi-armed bandits and beyond

Note takers: Achddou J., Barrier, Chapuis, Ganassali, Haddouche, Marthe, Saad, Tarrade, Van Assel

Instructor: Shipra Agrawal (Columbia University)

3.1 Introduction

When learning from sequential interactions, there is a tradeoff between:

- information and rewards:
- learning and optimization,
- exploration and exploitation.

In a nutshell, your goal is to get a maximal reward at each round and you have to choose between:

- **exploitation**: choosing an option that ensures you a good reward using information that you gathered in the past,
- **exploration**: choosing an option that has been less profitable in the past but on which you did not collect enough information, which might cost you a bad immediate reward but will allow you to understand better the system and thus obtain better rewards in the future.

We will start with introducing the basic Multi-armed Bandits problem. Multi-armed Bandits and Reinforcement Learning problems deal with the tradeoffs hereabove, gathered under the term *exploration/exploitation dilemma*.

Examples:

- Learn from customer's feedback to improve what products to show on internet. In this case, the outcome is whether or not the customer purchased the product,
- Machines at the casino : which one to put money on?

  Given an amount of money, you are free to choose in which machine(s) you will put it. Your goal is to maximize the money you will earn in total.

  Are you gonna try only one machine? Or all of them uniformly at the beginning, and then stop to use the bad ones?

  When to stop trying (exploration) and start playing (exploitation)?

Stochastic Multi-armed Bandit problem (MAB). We consider a setting of online decisions: at every round $t \in [T] = \{1, \dots, T\}$, we pull one arm $i_t \in \{1, \dots, N\}$ out of N arms using past information.

As a feedback, for each arm $i \in [N]$, a reward $r_{i,t}$ is generated i.i.d. from a **fixed but unknown distribution** with support in $[0, 1]$ and mean μ_i. The learner only observes the reward $r_t = r_{i_t,t}$ of the pulled arm i_t. The mean of the reward at time t (knowing that arm i_t has been selected) is thus μ_{i_t}.

The goal is to minimize the *regret* compared to the best arm $i^* = \arg\max_i \mu_i$:

$$\text{Regret}(T) = \mathbb{E}\left[\sum_{t \in [T]} \mu^* - \mu_{i_t} \right] = T\mu^* - \sum_{t \in [T]} \mathbb{E}[\mu_{i_t}]$$

where $\mu^* = \mu_{i^*} = \max_i \mu_i$ (note that i_t is a random variable, hence the expectation in the definition of $R(T)$).

If we know the best arm i^*, we can play $i_t = i^*$ at every round and get the optimal reward $T\mu^*$ (and regret 0): the regret is defined as the difference between what one could ideally have obtained and what we actually got at the end.

For each arm i, denoting by $\Delta_i = \mu^* - \mu_i$ its gap and by $k_i(T)$ its number of pulls up to time T, the expected regret can be rewritten as

$$\text{Regret}(T) = \sum_{i \neq i^*} \Delta_i \mathbb{E}[k_i(T)]$$

Of course, a strategy that learns something from the data will diminishes the observation frequency of any sub-optimal arm, hence its regret might be sublinear. At the opposite, if the proportions of pulls of sub-optimal arms do not evolve with time, the strategy does not learn anything and the regret is linear.

Outline. We will first cover basic algorithms involving UCB and Thompson Sampling. Then we will see useful generalizations: contextual bandits, bandits with constraints, assortment of bandits. We will finally study bandit techniques for MDP/RL.

The need for exploration. We will use a toy example to highlight how randomness forces the use of exploration.

We consider $N = 2$ arms: blue and red, with respective means $\mu_1 = 1.1$ and $\mu_2 = 1$. The optimal expected reward in this case is $1.1 \times T$.

A first natural strategy consists to simply pull the arm with the current best estimate (MLE/empirical mean) of unknown mean. This strategy is called Follow-The-Leader.

But we will see that initial trials can be misleading. Assume for instance that arm red is a Dirac distribution at 1: the associated sequence of rewards will be $1, 1, 1, \ldots$. If arm blue has quite a large variance, there is a positive probability that its empirical mean after a few trials will be lower than 1.

On this event denoted by E, you will then pull the red arm at every time step (as it empirical mean 1 will never go down under the empirical mean of the blue arm. Thus your expected regret after T steps will satisfy

$$\mathrm{Regret}(T) \gtrsim \mathbb{P}(E) \times 0.1T$$

hence the regret will be linear in T: the strategy fails to learn anything on event E. To correct this misbehaviour you have to **pay attention to exploration**!

As already explained, a good algorithm will have to balance between:

- **exploitation**: play the empirical mean reward maximizer,
- **exploration**: play less explored actions to ensure the convergence of empirical estimates.

3.2 Lower bounds

An algorithm has no way to know whether an arm is sub-optimal before it plays it. Thus it will have to observe sub-optimal arms at least a few times, leading to a non-negative regret. This has been quantified in the literature by Lai and Robbins (1985): for *any* given instance $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$ of the MAB problem, any "reasonable" algorithm will play a sub-optimal arm i at least $\Omega(\frac{\log(T)}{\Delta_i^2})$ times for large T, hence a minimal regret of

$$\mathrm{Regret}(T) \gtrsim \log(T) \sum_{i \neq i^*} \frac{1}{\Delta_i}.$$

This bound is instance-dependant: it depends on the distributions of the bandit parameter $\boldsymbol{\mu}$, more specifically through the gaps Δ_i.

Remark. *Imagine a strategy that always selects $i_t = 1$: it will have a 0 regret among all bandit parameters such that $i^* = 1$, but a linear regret among all other bandit parameters. Then this is not a good strategy, and this is the kind of strategy we remove when considering only "reasonable" strategies.*

On the other hand, there also exists a worst case bound: for every algorithm, there exists a bandit parameter for which $R(T) = \Omega(\sqrt{NT})$.

3.3 The Upper Confidence Bound algorithm (Auer 2002)

The strategy. We define the empirical mean at time t for am i as follows:

$$\hat{\mu}_{i,t} = \frac{\sum_{s \in [t]} r_s \mathbf{1}_{i_s = i}}{k_i(t)}.$$

As we already discussed, the empirical mean is not sufficient to capture the need for exploration. The idea of the UCB algorithm is to combine at each time t and for each arm i both an exploitation term (the empirical mean) and an exploration term (which is a bonus that decreases with the number of pulls) into an index denoted by $\mathrm{UCB}_{i,t}$:

$$\mathrm{UCB}_{i,t} = \underbrace{\hat{\mu}_{i,t}}_{\text{exploitation term}} + \underbrace{2\sqrt{\frac{\log t}{k_i(t)}}}_{\text{exploration term}}$$

The strategy is optimistic: we know that μ_i belongs to the confidence interval $[\hat{\mu}_{i,t} \pm 2\sqrt{\frac{\log t}{k_i(t)}}]$ w.h.p. and we take the highest value of this interval as basis: $\mathrm{UCB}_{i,t}$ overestimates μ_i. While increasing the number of observations this confidence region will shrink to $\{\mu_i\}$.

The UCB algorithm plays at time t the arm with the best optimistic estimates, as explained in Algorithm 1.

Regret analysis of UCB. Recall the expression of regret:

$$\mathrm{Regret}(T) = \sum_{i \neq i^*} \Delta_i \mathbb{E}[k_i(T)]$$

We assume optimistically that for any i, $UCB_{i,t} > \mu_i$.

First we bound the number of mistakes $\mathbb{E}[k_i(T)]$ for all suboptimal arms $i \neq i^*$.

A bound of $\mathbb{E}[k_i(T)] \leq \frac{C \log T}{\Delta_i^2}$.

Arm i will be played at time t only if $\mathrm{UCB}_{i,t} > \mathrm{UCB}_{i^*,t}$.

Input: number of arms N, number of steps T

Observe each arm once
$t \leftarrow N$ **while** $t < T$ **do**
  for *each arm* i **do**
    Compute $\mathrm{UCB}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{4 \log t}{k_i(t)}}$
  end
  $i_{t+1} \leftarrow \arg\max_i \mathrm{UCB}_{i,t}$
  Observe $r_{t+1} = r_{i_{t+1},t}$
  $t \leftarrow t + 1$
end

Algorithm 1: Upper Confidence Bound

If $n_{i,t} > \frac{16 \log T}{\Delta_i^2}$, we get $|\hat{\mu}_{i,t} - \mu_i| \leq \frac{\Delta_i}{2}$ with probability $1 - \frac{1}{T^2}$ using Azuma-Hoeffding inequality, and then

$$\mathrm{UCB}_{i,t} - \hat{\mu}_{i,t} = \sqrt{\frac{4 \log t}{n_{i,t}}} \leq \frac{\Delta_i}{2}$$

With high probability, arm i will not be pulled more than $\frac{16 \log(T)}{\Delta_i^2}$ (bound on expected number of mistakes), thus with high probability

$$\mathrm{Regret}(T) \leq 16 \log(T) \sum_{i \neq i^*} \frac{1}{\Delta_i}$$

3.3.1 Thompson Sampling (Thompson 1933)

Thompson Sampling is a Bayesian algorithm. The general idea is to maintain belief about parameters (*e.g.* mean reward) of each arm. Then observe the feedback, update the belief of pulled arm in a Bayesian manner. Belief update is performed using Bayes rule: the posterior is proportional to the product of likelihood and prior. Importantly, we don't try to estimate the parameters in this setting.

We pull the arm by sampling from the posterior probability of being the best arm. Note that this is different than choosing the arm that is the most likely to be the best.

The main intuition of maintaining Bayesian posteriors is the following:

- When the number of trials increases, the posterior concentrates on the true parameters. This phenomenon enables exploitation, as the mode of the posterior captures the maximum likelihood estimate.
- Moreover, uncertainty is high when the number of trials is small. This variance captures the uncertainty about the arms and enables exploration.

Example of Bernoulli rewards with Beta priors. In the case of Bernoulli rewards, we pick the Beta distribution since it is *conjugate* (it is important because drawing a point according to updated Bayesian posterior may be costly in the general case and often requires MCMC methods). If you take $\text{Beta}(\alpha, \beta)$ as a prior, then the posterior is updated as follows:

- $\text{Beta}(\alpha + 1, \beta)$ if you observe 1.
- $\text{Beta}(\alpha, \beta + 1)$ if you observe 0.

Note that every time you observe a sample, the variance decreases[1].

We start with a $\text{Beta}(1, 1)$ distribution as prior belief for every arm. Then in round t:

- for every arm i, sample $\theta_{i,t}$ independently from current posterior $\text{Beta}(S_{i,t} + 1, F_{i,t} + 1)$, where:
$$S_{i,t} = \sum_{s \in [t-1]} \mathbf{1}_{r_s=1} \mathbf{1}_{i_s=i} \quad \text{and} \quad F_{i,t} = \sum_{s \in [t-1]} \mathbf{1}_{r_s=0} \mathbf{1}_{i_s=i},$$
- play arm $i_{t+1} = \arg\max_i \theta_{i,t}$,
- observe reward and update the Beta posteriors

$$F_{i,t+1} = \begin{cases} F_{i,t} & \text{if } i \neq i_{t+1} \\ F_{i,t} + \mathbf{1}_{r_{t+1}=0} & \text{if } i = i_{t+1} \end{cases} \quad \text{and} \quad S_{i,t+1} = \begin{cases} S_{i,t} & \text{if } i \neq i_{t+1} \\ S_{i,t} + \mathbf{1}_{r_{t+1}=1} & \text{if } i = i_{t+1} \end{cases}$$

Example of continuous rewards with Gaussian priors. We take a standard $\mathcal{N}(0, 1)$ prior. The reward likelihood is $\mathcal{N}(\hat{\mu}, 1)$ such that the posterior after n independent observations simply takes the form $\mathcal{N}(\hat{\mu}, \frac{1}{n+1})$ where $\hat{\mu}$ is the empirical mean.

Start with $\mathcal{N}(0, v^2)$ prior belief for every arm. Then in round t:

- for every arm i, sample $\theta_{i,t}$ independently from current posterior $\mathcal{N}(\hat{\mu}_{i,t-1}, \frac{v^2}{n_i(t-1)+1})$,
- play arm $i_t = \arg\max_i \theta_{i,t}$,
- observe reward and update the empirical mean $\hat{\mu}_{i,t}$.

Remark: In practice Thompson sampling seems to be more efficient in general than UCB since UCB involves the optimistic assumption of the overestimation of the mean which may not be realistic.

Why does it work? For the sake of simplicity, we come back to the two arms example: we consider two arms with $\mu_1 \geq \mu_2$, $\Delta = \mu_1 - \mu_2$. In this case we directly have that if arm 2 is pulled, the regret is Δ.

We want to bound the number of pulls of arm 2 by $\frac{\log T}{\Delta^2}$ to get a $\frac{\log T}{\Delta}$ regret bound.

[1] the variance of $\text{Beta}(\alpha, \beta)$ is $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

How many pulls of arm 2 are actually needed?

After $n \geq \frac{16\log(T)}{\Delta^2}$ pulls of arm 2 and arm 1.

Using Azuma-Hoeffding, one has : $|\hat{\mu}_i - \mu_i| \leq \sqrt{\frac{\log(T)}{n}} \leq \frac{\Delta}{4}$ with high probability. So the arms are well separated.

Beta posteriors are well separates: their mean is $\frac{\alpha_i}{\alpha_i+\beta_i} = \hat{\mu}_i$ and standard deviation about

$$\frac{1}{\sqrt{\alpha+\beta}} = \frac{1}{\sqrt{n}} \leq \frac{\Delta}{4}$$

Thus the 2 arms can be distinguished and arm 2 will not be pulled anymore. Hence the importance of verifying both arms have been pulled enough to ensure the consistency of our result.

Extension to multiple arms. One has the following kind of results:

$$\mathbb{P}(a_t = a^*|F_{t-1}) \geq \frac{p}{1-p}\mathbb{P}(a_t = a|F_{t-1})$$

where p is the probability of anti-concentration of posterior sample for the best arm.

Best arm gets played roughly every $1/p$ plays of arm a.

- p can be lower bounded by Δ_a in general but it actually goes to 1 exponentially fast with increase in number of trials of best arm,
- cannot accumulate from arm a without playing a^* sufficiently.

3.4 Useful generalizations of the basic MAB problem

Different generalizations could be useful depending on the application:

- pulling more than one arm at a time
- having unknown distributions
- changing the feedback (having it censored for instance)
- having a goal different than reward maximization

3.4.1 Handling context in MAB

In this part, we will only consider linear contextual bandit. They make sense in a lot of application, for instance in content based recommendation, where customers and product can be described by their features. It allows for an easier way of dealing with a large amount of products and customer types, and the features will allow to make profit of similarities across product or users.

Linear contextual bandits :

- N arms (possibly very large),
- a d-dimensional context (feature vector) $x_{i,t}$ for every arm i, time t
- Linear parametric model, with parameter θ
- algorithm picks $x_t \in \{x_{1,t}, \dots, x_{N,t}\}$, observe $r_t = x_t \cdot \theta + \eta_t$
- Optimal arm depends on context: $x_t^* = \arg\max_i x_{i,t} \cdot \theta$
- Goal : minimize regret $\sum_t (x_t^* - x_t) \cdot \theta$

UCB for contextual bandits Linear regression is used to approximate the parameter :

- least square solution $\hat{\theta}_t$ of set of equation $x_s \dot{\theta} = r_s$
- $\hat{\theta}_t \simeq B_t^{-1} \left(\sum_{s=1}^{t-1} x_s r_s \right)$ where $B_t = I + \sum_{s=1}^{t-1} x_s x_s^\top$

With high probability, we have the bound : $\|\theta - \hat{\theta}_t\|_{B_t} \leq C\sqrt{d \log(Td)}$

Remark. *The bound doesn't depend on the number of arm, only on the dimension.*

The algorithm proceeds as follows. At time t:

- Observe the context $x_{i,t}$ for different arms $i = 1, \dots, N$
- Compute optimistic parameter estimates and confidence intervals for every arm
- Choose the best arm according to the most optimistic estimates

$$\arg\max_i \max_\theta \theta^\top x_{i,t} \text{ such that } \left| |\theta - \hat{\theta}| \right|_{B_t} \leq C\sqrt{d \log(Td)}$$

for *each* $t = 1, \dots, T$ **do**
  Observe set $A_t \subseteq [N]$, and context $x_{i,t}$ for all $i \in A_t$
  Play arm $I_t = \arg\max_{i \in A_t} \max_{z \in C_t} z^\top x_{i,t}$ with C_t as defined
  Observe r_t. Compute C_{t+1}
end

Algorithm 2: LinUCB algorithm

Proof.

$$\text{Regret}(T) = \sum_{t=1}^{T} (x_t^* \cdot \theta - x_t \cdot \theta)$$

with $x_t^* = x_{i^*,t}$, $i^* = \arg\max_i x_{i,t}^\top \theta$, and $x_t = x_{i_t,t}$

For the first part, we always make use optimism :

$$
\begin{aligned}
R(T) &\leq \sum_{t=1}^{T} (x_t^* \cdot \theta - x_t \cdot \theta) \qquad \text{with high P} \\
&= \sum_{t=1}^{T} x_t (\tilde{\theta}_t - \theta) \\
&\leq \sum_{t=1}^{T} \sqrt{x_t^\top B_t^\top x} \sqrt{(\tilde{\theta}_t - \theta)^\top B_t (\tilde{\theta}_t - \theta)} \\
&= \sum_{t=1}^{T} \|x_t\|_{B_t^{-1}} \underbrace{\left\|\tilde{\theta} - \theta\right\|_{B_t}}_{\leq \sqrt{Cd \log(Td)}} \\
&= \sum_{t=1}^{T} \|x_t\|_{B_t^{-1}} \sqrt{Cd \log(Td)}
\end{aligned}
$$

Moreover, by the Elliptical Potential Lemma (see e.g. Lattimore and Szepesvári [2020], Chapter 20) :

$$
\sum_t \|x_t\|_{B_t^{-1}} = x_t^\top B_t^{-1} x_t = \tilde{O}(\sqrt{d})
$$

which finishes the proof. □

Remark. *Thompson Sampling for linear contextual bandits uses the (Gaussian) Bayesian Linear Regression to sequentially maintain a posterior distribution over the unknown parameter θ. Regret guarantees currently show a slight suboptimality: $R(T) = O(d^{3/2}\sqrt{T})$ but it is still unclear whether this is due to an artefact in the proof or if that extra $\sqrt{d}$ should be here for more fundamental reasons.*

3.4.2 Assortement selection as multi-armed bandit

The customer response to the recommended assortment **may depend** on the combination of items and not just the marginal utility of each item, in the assortment.

Ex: An assortment combining 3 types of cell phones might push the user to go for the cheapest.

Setting:

- selecting a subset $S_t \in [N]$ in each of the sequential rounds $t = 1, \ldots, T$.
- On selecting a subset S_t, reward r_t is observed with expected value $\mathbb{E}[r_t|S_t] = f(St)$ where the function $f : R^N \mapsto [0, 1]$ is unknown

Different possible structural assumptions on f (e.g. Lipschitz). But also multinomial logit choice model(MNL).

The multinomial logit choice model (Luce 1959, MfFadden 1978) is the following:

the probability that a consumer purchases product i at time t when offered an assortment S is

$$p_i(S) = \frac{e^{\theta_i}}{1 + \sum_i e^{\theta_i}} \text{ if } i \in S \cap \{0\} \text{ or } 0 \text{ otherwise} \tag{3.1}$$

i can be 0 meaning that there is no purchase.

The idea is to take into account the distribution of other arms. Pulling an arm no longer depends on its marginal distribution only. The above is a simple model to do so.

MNL bandit problem. In this setting we have N products, unknown θ_i. At every step t, we recommend an assortment S_t of size $\leq K$, observe customer response i_t, revenue r_{i_t}, and update parameter estimates. The customer's behavior is modeled by 3.1.

The goal is to optimize the total revenue $\mathbb{E}\left[\sum_t r_{i_t}\right]$, or minimize the regret compared to the optimal assortment

$$\mathcal{R}(T) := Tf(S^*) - \mathbb{E}\left[\sum_t r_t\right] = \sum_t (f(S^*) - f(S_t))$$

where $S^* = \max_S f(S)$. In many cases, even if the expected value $f(S)$ is known for all S, computing S may be intractable. Therefore, for this problem to be tractable some structural assumptions on f will be made.

Main challenges Censored feedback: feedback of product i is effected by other products in a given assortment (combinatorial: N^K choices). In other words, the response observed on offering a product i (as part of an assortment S) is not independent of other products in the assortment.

Technique to get unbiaised estimate of *individual parameters*: offer a given assortment S until no purchase: the number of times $n_S(i)$ that i is purchased in S on this process is an unbiased estimator of e^{θ_i}. Indeed, one has

$$\mathbb{E}[n_S(i)] = \frac{p_i(S)}{p_0(S)} = e^{\theta_i}.$$

Concretely, if at any round t a purchase of any item in the offered set S_t is observed, then the algorithm continues to offer the same assortment in round $t+1$, *i.e.* $S_{t+1} = S_t$. If a no-purchase

is observed in round t, then the algorithm updates the parameter estimates and makes a new assortment selection for round $t + 1$.

Then, having established confidence intervals for the parameters θ_i, we can run UCB and Thompson Sampling techniques.

UCB based algortihm Agrawal et al. [2019] acheives $O(\sqrt{NT})$ regret.

3.4.3 Bandits with constraints and non-linear aggregate utility

Generalizing MAB: we observe a non-negative reward r_t ans a cost vector c_t. The problem now becomes:

$$\max \sum_t r_t \text{ s.t. } \forall j, \sum_t c_{t,j} \leq B.$$

-> Bandits with Knapsacks Badanidiyuru et al. [2013]

A generalization of this is Bandits with convex knapsacks and concave rewards (BwCR), with convex constraints domains and concave rewards.

Bandits with convex knapsacks and concave rewards (BwCR) Agrawal and Devanur [2014] Pulling an arm i_t generates $v_t \in \mathbb{R}^d$ with unknown mean V_{i_t}.

Total number of pull constrained by T + arbitrary convex global contraints of the form $\frac{1}{n}\sum_t v_t \in S$, with S a convex set.

The goal is to maximize $f\left(\frac{1}{n}\sum_t v_t\right)$, for f an arbitrary concave function, of minimise $d\left(\frac{1}{n}\sum_t v_t, S\right)$ as one has to assure $\frac{1}{n}\sum_t v_t \in S$.

Results for UCB-like optimistic algo for BwCR We need to estimate for every arm i and coordinate j.

We are interested in the following problem, where $H_t = \{\overline{V} : \overline{V}_{ij} \in [\text{LCB}_{t,ij}, \text{UCB}_{t,ij}]\}$,

$$p_t = \arg_p \max_{\overline{V} \in H_t} f\left(\sum_i p_i \overline{V}_i\right) \tag{3.2}$$

$$\text{s.t.} \quad \min_{\overline{U} \in H_t} \text{dist}\left(\sum_i p_i \overline{U}_i, S\right) \leq 0 \tag{3.3}$$

For non-decreasing f, the inner maximizer in the objective of 3.3 will be simply the UCB estimate, therefore for the classic MAB problem this algorithm reduces to the UCB algorithm.

3.5 Bandit techniques for Markov Decision Processes

General formulation The general problem is as follows: the reward on pulling an arm (action) depends on the *current* state of the system. Each round $t = 1, \dots, T$ consists in observing the current state, taking an action, observing the reward and a new state. The solution concept is referred to as a *policy*. The general goal is to learn the state transition dynamics and the reward distributions, while optimizing the policy.

(Application: inventory management, autonomous vehicle control, robot navigation, personalized medical treatments...)

3.5.1 MDPs

at round t, the player observes state s_t, take action a_t, observe reward $r_t \in [0, 1]$ and next state s_{t+1}.

The system dynamics is given by a MDP (S, A, r, P, s_0) such that

$\mathbb{E}[r_t|s_t, a_t, H_t]\mathbb{E}[r_t|s_t, a_t] =: r_{s_t,a_t}$, and $\mathbb{P}[s_{t+1}|s_t, a_t, H_t]\mathbb{P}[s_{t+1}|s_t, a_t] =: P_{s_t,a_t}(s_{t+1})$.

Due to Markov property, there is an optimal policy $\pi : S \to A$. The goal is to minimize expected regret compared to the best stationary policy π^*, defined as follows

$$\text{Regret}(M, T) := \sum_{t=1}^{T} \left[r(s_t^*, \pi^*(s_t^*)) - r(s_t, a_t) \right] .$$

We want to learn the MDP model parameters (r, P) from observations $(s_t, a_t, r_t, s_{t+1})_{1 \le t \le T}$, while optimizing the policy for total expected reward.

In these models, regret bounds will be of the form $O(S\sqrt{TA})$.

Need for exploration Let us look at a single state MDP: the situation boils down to the classical MDP problem, for which the exploit only policy may mislead into playing bad action forever.

[example of a two-states MDP] Let us illustrate the fact that exploiting the seemingly best policy is not the optimal choice. In the above example, initializing at state 1 and playing red action forever would avoid the best action (state 2, black action), which needs a bit of 'faith' in order to be discovered! In MDPs, the exploitation is thus even more important that in classical MAB.

Communicating MDPs Caveat: MDP can get stuck on bad states for a long time, depending on the underlying graph structure.

Let us define *communicating MDPs*, which are MDPs for which there is always a way to get out of a bad state in finite time. Namely, for every pair of states s, s', there is a policy π (*that*

depends on s, s') such that using this policy starting from s, the expected time to reach s' is finite and bounded by D, called the *diameter* of the MDP.

Some useful properties:

- the optimal asymptotic average reward is independent of the starting state
- the asymptotic average reward (gain) of policy π is defined by

$$\lambda^\pi(s) := \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r(s_t, \pi(s_t))\middle| s_1 = s\right]$$

  There is a single policy π^* achieving the optimal infinite average reward

$$\forall s,\ \max_p i\lambda^\pi(s) = \lambda^{\pi^*}(s) =: \lambda^*$$

- We define the regret as the gain compared to asymptotic normal:

$$\text{Regret}(M, T) := T\lambda^* - \sum_{t=1}^{T} r(s_t, a_t).$$

Some bounds Upper confidence bounds based algorithms Auer et al. [2008], Bartlett and Tewari [2012] : worst-case regret bound $O(DS\sqrt{AT})$, lower bound $\Omega(\sqrt{DSAT})$.

Optimistic Posteriori Sampling Agrawal and Jia [2017]: worst-case regret bound in $O(DS\sqrt{AT})$.

3.5.2 UCRL: Upper confidence bound based algorithm for RL

Expected reward $R(s, a)$ for all $s \in S, a \in A$, as well as $P(s, a)$ a distribution on S. At each step, we can use r_t to update an estimator of $R(s_t, a_t)$ and s_{t+1} for $P(s_t, a_t)$

UCRL algorithm Model-based approach: maintain estimates $\hat{P}, \hat{R}$, and occasionally solve the MDP $(S, A, \hat{P}, \hat{R}, s_1)$ to find a policy, run this policy for some time to get samples, update the estimates, and iterate. We proceed in epochs:

At every epoch k, use samples to compute an optimistic MDP $(S, A, \tilde{R}, \tilde{P}, s_1)$, solve it to find an optimal policy $\tilde{\pi}$. Then, execute $\tilde{\pi}$ in epoch k, and observe samples s_t, a_t, r_t, s_{t+1}. Then, go to next epoch if $n_k(s, a) \geq 2n_{k-1}(s, a)$ for some s, a.

[missing: more involved description of the algorithm]

Theorem 3.1. *For any communicating MDP with unknown diameter D, we have with high probability*

$$\text{Regret}(M, T) \leq \tilde{O}(DS\sqrt{AT}),$$

where $\tilde{O}$ hides logarithmic factors in S, A, T.

Proof sketch. By the communicating property, we have that w.h.p., the extended MDP in UCRL is communicating. For this extended extended MDP, the optimal average reward $\tilde{\lambda}(s)$ is independent of s.

The average regret in an epoch k is

$$\lambda^* - \frac{1}{T_k} \sum_{t \in [T_k]} r_{s_t, a_t} = (\lambda^* - \tilde{\lambda}) + (\tilde{\lambda} - \frac{1}{T_k} \sum_{t \in [T_k]} r_{s_t, a_t}).$$

The above first term is non-postive by construction of UCRL. For the second term, we follow the same policy but on a different MDP. The bounds are obtained using concentration of transition probability vector samples from the posterior.

Bellman equation Define the value functions

$$v_\gamma^M(s) = \mathbb{E}_M \left[\sum_{k \geq 1} \gamma^{k-1} r_k \right].$$

We have

$$v_\gamma^M(s) = R(s, M(s)) + \gamma \mathbb{E}_{s' \sim P^M(s,\cdot)} \left[v_\gamma^M(s') \right],$$

which also writes

$$v_\gamma^M = R^M + \gamma P^M \cdot v_\gamma^M$$

We can show that

$$\lambda^M(s) = \lim_{\gamma \to 1} (1 - \gamma) v_\gamma^M(s)$$

Define the bias vector $h^M(s) := \mathbb{E}\left[\lim_{T \to \infty} \sum_{t=1}^T (r_t - \lambda^M(s_t)) | s_1 = s\right]$. We actually have that

$$h^M(s_1) - h^M(s_2) = \lim_{\gamma \to 1} (v_\gamma^M(s_1) - v_\gamma^M(s_2)).$$

These two equations, together with the fixed point equation satisfied by $v_\gamma^M(s)$ give that for $\gamma \in (0, 1)$,

$$(1 - \gamma) v_\gamma^M(s) = R^M(s) + \gamma \sum_{s'} P^M(s, s') (v_\gamma^M(s') - \gamma v_\gamma^M(s)).$$

Then, sending $\gamma \to 1$, one gets the Bellman equation:

$$\lambda^M(s) = R^M(s) + \sum_{s'} P^M(s, s') h^M(s') - h^M(s).$$

Bounding the difference In our context Bellman equation writes $\tilde{\lambda} - r_{s,\pi(s)} = \tilde{P}_{s,\pi(s)} \cdot \tilde{h} - \tilde{h}_s$, where $\tilde{h}$ is the bias vector of samples and satisfies $|\tilde{h}_i - \tilde{h}_j| \leq D$ for all $i, j \in S$. Thus

$$\tilde{\lambda} - \frac{1}{T_k} \sum_{t \in [T_k]} r_{s_t,a_t} = \frac{1}{T_k} \sum_{t \in T_k} \left(\tilde{P}_{s_t,a_t} \cdot \tilde{h} - \tilde{h}_{s_t} \right)$$
$$= \frac{1}{T_k} \sum_{t \in T_k} \left(\tilde{P}_{s_t,a_t} \cdot \tilde{h} - P_{s_t,a_t} \cdot \tilde{h} + P_{s_t,a_t} \cdot \tilde{h} - \tilde{h}_{s_t} \right).$$

By martingale property, $P_{s_t,a_t} \cdot \tilde{h} - \tilde{h}_{s_t} = 0$. Then, w bound the deviation of posterior sample from the true model $(\tilde{P}_{s_t,a_t} - P_{s_t,a_t}) \cdot \tilde{h}$ (posteriori variance, sample error...). Since here $\tilde{h}$ is not fixed, we need a union bound, giving a bound in $\tilde{O}(D\sqrt{S}/\sqrt{N_{s,a}})$. □

3.5.3 Posterior sampling algorithm for MDPs

A more intuitive algorithm with different techniques for regret bound proofs.

Finite state, finite action S states, A actions.

Prior: Dirichlet $(\alpha_1, \alpha_2, ..., \alpha_i + 1, ..., \alpha_s)$ on $P_{s,a}$

After $n_{s,a} = \sum \alpha_i$ observations for a state-action pair s, a one computes the posterior $\hat{p}_{s,a}(i) = \frac{\alpha_i}{\sum_j \alpha_j} = \frac{\alpha_i}{n_{s,a}}$.

The variance is bounded by $\frac{1}{n_{s,a}}$: the more we have trials, the more the posterior concentrated around true probability.

Learning phase One maintains a Dirichlet posterior for $P_{s,a}$ for any (s, a). We start with an uninformative prior Dirichlet $(1, 1, \ldots, 1)$.

Deciding phase We first sample $\tilde{P}_{s,a}$ for any (s, a). Then the optimal policy $\tilde{\pi}$ is computed for the MDP $(S, \varphi A, \tilde{P}, r, s_0)$

Our algorithm

- For any (s, a), generate multiple $\psi = \tilde{O}(S)$ independent samples from a Dirichlet posterior for $P_{s,a}$.
- Form extended sample MDP $(S, \psi A, \tilde{P}, r, s_0)$.
- Form optimal policy $\tilde{\pi}$ and use through the epoch.

Further initial exploration: For (s, a) with very small $N_{s,a} < \sqrt{\frac{TS}{A}}$ use simple optimistic sampling that provides extra exploration.

Regret bound analysis Assumption True MDP is communicating with diameter D

- For UCRL: with high probability, extended MDP is communicating
- For posterior sampling whp extended MDP is a communicating MDP with diameter at most $2D$.

We recall that a useful property of communicating MDP is that optimal asymptotic average reward does not depend on the initial state.

The averaged regret in an epoch k is

$$\lambda^* - \frac{1}{T_k} \sum_{t \in [T_k]} r_{s_t,a_t} = (\lambda^* - \tilde{\lambda}) + (\tilde{\lambda} - \frac{1}{T_k} \sum_{t \in [T_k]} r_{s_t,a_t}).$$

Two main results: First the optimism of transition matrix on a projection is sufficient $\geq \lambda^*$ if for every s, a, $\tilde{P}_{s,a}.h^* \geq P_{s,a}.h$ if a set of samples satisfy optimism on projection to unknown bias vector h^*.

Second For any fixed bounded vector h a sample satisfies above with probability $1/S$. There is no need to know h^*! But there is a need of $O(S \log(\frac{SA}{\rho}))$ samples whp.

3.6 Learning to manage inventory

It gives a general recipe for a loose class of problems.

Overview:

You start the inventory at time t, you observe inv_t, you gather new o_t and old o_{t-L} orders. Then you have to deal with the demand d_t after dealing a new on hand inventory $I_t = inv_t + o_{t-L}$. You then observe $y_t = mon(I_t, d_t)$.

You finally incur holding and lost sales cost $h(I_t, d_t)$...

Learning an MDP In each round $t = 1..T$:

- Observe inventory I_t, past $L-1$ orders $(o_{t-L+1}, ..., o_{t-1})$.
- Decide new order $o_t \in [0, U]$
- Observe sales $y_t = \min(d_t, I_t)$ where $d_t \sim F$.
- Incur cost $\bar{C}_t = h(I_t - y_t) + p(d_t - y_t)$
- Start new inventory $I_{t+1} = I_t - y_t + o_{t-L+1}$.

Holding unobserved lost sales The actual cost is $\bar{C}_t = h(I_t - y_t) + p(d_t - y_t)$ but d_t is unknown. We then use the surrogate $C_t = h(I_t - y_t) + p(-y_t)$.

Bibliography

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006, 2014.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in Neural Information Processing Systems*, 30, 2017.

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. Mnl-bandit: A dynamic learning approach to assortment selection. *Operations Research*, 67(5):1453–1485, 2019.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n). *CoRR*, abs/1306.2119, 2013. URL http://arxiv.org/abs/1306.2119.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

Peter L Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. *arXiv preprint arXiv:1205.2661*, 2012.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2010.04.005. URL https://www.sciencedirect.com/science/article/pii/S1063520310000552.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Edoardo Di Napoli, Eric Polizzi, and Yousef Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, March 2016. doi: 10.1002/nla.2048. URL https://doi.org/10.1002/nla.2048.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *arXiv preprint arXiv:1209.1873*, 2012.

David I. Shuman, Mohammad Javad Faraji, and Pierre Vandergheynst. A multiscale pyramid transform for graph signals. *IEEE Trans. Signal Process.*, 64(8):2119–2134, 2016. doi: 10.1109/TSP.2015.2512529. URL https://doi.org/10.1109/TSP.2015.2512529.