Notes on Bandit Learning

Yiming Xu

1Department of Mathematics, University of Utah, Salt Lake City, 84112
These notes are born out of a reading course at the University of Utah in Spring 2020. Most of the results in the notes are taken from [LS18], while others trace back to a few papers that I ran into along the way. The purpose of the writeup is to assist my own understanding of some topics in bandit learning, meanwhile to keep a record. Any imprecision or mistakes are of my own.
# Contents

1 Stochastic Bandits .............................................. 5
   1.1 Problem Set-up ........................................... 5
   1.2 Algorithms .................................................. 6
   1.3 Lower Bounds on Regret ................................... 12
      1.3.1 Minimax Lower Bound ................................ 12
      1.3.2 Instance-Dependent Lower Bound ....................... 14
   1.4 Thompson Sampling .......................................... 15

2 Stochastic Linear Bandits .................................... 20
   2.1 Some Generalization ....................................... 20
   2.2 Stochastic Linear Bandits with a Finite and Fixed Action Set ........................................................................... 22
   2.3 Stochastic Linear Bandits with General Action Sets ...................................................................................... 26
      2.3.1 Laplace’s Method .......................................... 29
      2.3.2 Online Learning Prediction ............................... 30
   2.4 Stochastic Linear Bandits on Graphs ......................... 32
   2.5 Instance-Dependent Lower Bound ............................. 36

3 Adversarial Linear Bandits ................................... 40
   3.1 Problem Set-up ................................................... 40
   3.2 Algorithms ....................................................... 42
   3.3 Online Linear Optimization .................................. 46
   3.4 Universal Portfolio ............................................ 52

4 Bandits with Cost ................................................. 56
   4.1 Best-Arm Identification ................................. 56
      4.1.1 Best-Arm Identification with a Budget .................. 56
      4.1.2 Best-Arm Identification with a Fixed Confidence .... 63
   4.2 Budget-Limited Bandits ....................................... 65
   4.3 Multi-Fidelity Bandits .......................................... 68

5 Gittins Index .......................................................... 73
   5.1 Markov Decision Processes .................................... 73
      5.1.1 $\alpha$-Discounted Cumulative Reward .................. 74
      5.1.2 Long-Run Average Reward .............................. 76
CONTENTS

5.2 Optimal Stopping ..................................... 78
5.3 Gittins Index and Bayesian Optimality .............. 80
Chapter 1

Stochastic Bandits

1.1 Problem Set-up

Suppose that you are facing a $k$-armed slot machine. Every time you can choose one arm to play and the corresponding reward will be immediately revealed. Rewards of the arms not played in the round are assumed hidden. As players, we wish to find a strategy so that our gain is reasonably good after a given period of time, or equivalently, our regret is relatively small compared to some ‘optimal strategy’. This leads to a class of sequential decision making problems which we refer to as the bandit learning.

Of course, no good strategies exist without making assumptions. One of the simplest and most popular one is to assume that rewards generated by each arm are i.i.d. random variables. The bandit model under this assumption is called a stochastic bandit.

Supposing the number of rounds to play is given, one would probably want to spend some time exploring on different arms first to see which one yields the best reward on average, then commit to the decision concluded from the exploration. Such philosophy is often stated as the exploration-exploitation trade-off, which plays a crucial role in many bandit learning problems. In the rest of this chapter we will quantify such idea in the stochastic bandits set-up.

Some notations are needed first. Let $[k] := \{1, \cdots, k\}$ be the set of arms and $n$ be the horizon of the game. $\{x_i\}_{i \in [n]}$ denote the sequence of rewards generated by arm $i$ in $n$ rounds, which are i.i.d. random variables with mean $\mu_i$. In the following discussion we assume that centered reward distributions are 1-sub-gaussian and unstructured, in the sense that acquiring knowledge on one arm does not imply the others. A counterexample is $\mu_2 = 1 - \mu_1$ when $k = 2$. A deterministic policy is defined as an $[k]^n$-valued random vector $\pi = (\pi_i)_{i \in [n]}$ such that $\pi_i$ is a measurable function of $\{x_{is}\}_{s \in [t-1]}$. A non-deterministic policy, on the other hand, possesses some external randomness. Their difference will not be emphasized until we discuss adversarial bandits. For now, we will mainly focus on the deterministic policies for stochastic bandits. This means that which arm to play in round $t$ is completely determined by the information disclosed before $t$.

\footnote{To avoid overuse of notation, $\mu_i$ also denotes the reward distribution since many randomness considered in the notes are from parametric families indexed by $\mu_i$.}
The regret $R_n$ of $\pi$ in the environment $\mu = (\mu_1, \cdots, \mu_k)$ is defined as the average difference between the collected rewards and the best arm in hindsight:

$$R_n(\pi, \mu) := \max_{i \in [k]} \mathbb{E} \left[ \sum_{t \in [n]} (x_{t,i} - x_{t,i^*}) \right] = n \max_{i \in [k]} \mu_i - \mathbb{E} \left[ \sum_{i \in [k]} x_{t,i^*} \right]. \quad (1.1.1)$$

We will often write $R_n$ or $R_{n,\mu}$ instead of $R_n(\pi, \mu)$ when the omitted parts are clear from the context.

A few things to note. First of all, whether (1.1.1) is meaningful needs clarification. Also, it is not clear what kind of bounds would imply that a policy $\pi$ is reasonably good. The first question is hard to justify from a mathematical point of view. One could consider moving the max into the expectation so that the resulting regret is even stronger. However, this will give too much freedom to the bandits, making the analysis hard to carry out. As we shall see in the next chapter, there is a way to inject more freedom in taking maximization while keeping things under control. For now we may assume (1.1.1) as a good place to start with. For the second question, note that the trivial policy by selecting a fixed arm leads to a regret which grows linearly in $n$. Therefore, any policy achieving sub-linear regret would be reasonable, and how far this can go will be explored later in this section.

In the following analysis, let $i^* \in \arg\max_{i \in [k]} \mu_i$ be an optimal arm. For $i \in [k]$, define the sub-optimality gap of $i$ as $\Delta_i = \mu_{i^*} - \mu_i$. For $t \leq n$, define the number of rounds where $i$ is chosen before $t$ under $\pi$ as $T_i(t, \pi)$ (which again is written as $T_i(t)$ for short), which is a random variable adapted to the natural filtration. Using the tower property of expectation, one can rewrite $R_n$ as

$$R_n = \sum_{i \in [k]} \Delta_i \mathbb{E}[T_i(n)]. \quad (1.1.2)$$

Such form provides a convenient way to analyze $R_n$. Indeed, since the summation is over $i$, one often only needs to bound $\mathbb{E}[T_i(n)]$ for each $i$ under a given policy.

### 1.2 Algorithms

We mention two well-known algorithms in stochastic bandit learning: The Explore-then-Commit (ETC) algorithm and the Upper Confidence Bound (UCB) algorithm, each of which is followed by a short summary of their pros and cons, as well as some asymptotic results.

The idea behind Algorithm 1 is simple. Divide $n$ rounds into two parts: the first $mk$ rounds for exploration and the remaining for exploitation. If $m$ is small, there is a considerable chance that the quality of exploration is poor, making the exploitation procedure sub-optimal. If $m$ is large, the regret incurred in the exploration process will likely dominate. Therefore, the best $m$ is usually set at some middle point, as summarized in the following theorem.
Algorithm 1: The Explore-then-Commit Algorithm

**Input:** $m$: the number of exploration on each arm; $n$: horizon

**Output:** $\pi = (\pi_t)_{t \in [n]}$

1. **for** $t = 1, \ldots, n$ **do**
2. **if** $t \leq mk$ **then**
3. $\pi_t = \lceil t \mod k \rceil$
4. **else**
5. $\pi_t = \arg\max_{i \in [k]} \hat{\mu}_i(mk)$, where $\hat{\mu}_i(mk) = \frac{1}{m} \sum_{t=m(i-1)+1}^{m_i t} x_t$
6. **end if**
7. **end for**

**Theorem 1.2.1.** The regret $R_n$ under ETC is given by

$$R_n \leq m \sum_{i \in [k]} \Delta_i + (n - mk) \sum_{i \in [k]} \Delta_i e^{-\frac{m \Delta_i^2}{4}}. \quad (1.2.1)$$

Particularly, when $k = 2$, taking $m = \max \left\{ 1, 4 \Delta_\ast^2 \log \left( \frac{n \Delta_\ast^2}{4} \right) \right\}$ yields

$$R_n \leq \Delta_\ast + C \sqrt{n}, \quad (1.2.2)$$

where $C$ is some absolute constant.

**Proof.** The results in Theorem 1.2.1 follow directly from the tail bounds for the sum of independent sub-gaussian random variables. $\Box$

**Remark 1.2.2.** A high-probability version of the result on the pseudo-regret $\tilde{R}_n$, which is defined as

$$\tilde{R}_n = \sum_{i \in [k]} \Delta_i T_i(n),$$

can be obtained similarly:

$$\mathbb{P} \left( \tilde{R}_n \leq m \sum_{i \in [k]} \Delta_i \right) \geq 1 - \sum_{i \in [k]} e^{-\frac{m \Delta_i^2}{4}}.$$

**Remark 1.2.3.** Despite the fact that the bound given in (1.2.2) is minimax optimal (Theorem 1.3.1), how to achieve it depends on knowledge of both the sub-optimality gaps $\Delta_i$ and the horizon $n$, which may not be known in advance. In theory, it can be shown that for two-armed bandits the dependence on $\Delta$ can be removed while obtaining a sub-optimal regret bound $n^{2/3}$, and the dependence on $n$ can be resolved by a doubling trick without increasing the regret too much.
To address the dependence on the sub-optimality gaps, another algorithm called UCB was proposed. The UCB algorithm belongs to a class of algorithms where confidence intervals are used to provide optimism in exploration. The idea first appeared in the seminal work [LR85], where the asymptotics of various parametric bandits was analyzed. The UCB algorithm given below most resembles the UCB1 algorithm in [ACF02], which has a slightly different confidence level.

Algorithm 2: The Upper Confidence Bound Algorithm

Input: $\delta$: confidence level; $n$: horizon
Output: $\pi = (\pi_i)_{i \in [n]}$

1. while $t \leq n$ do
2. $\pi_i = \arg \max_{i \in [k]} \text{UCB}_i(t - 1, \delta)$, where for $i \in [k]$,

\[
\text{UCB}_i(t - 1, \delta) = \begin{cases} 
\infty & T_i(t - 1) = 0 \\
\hat{\mu}_i(t - 1) + \sqrt{\frac{2 \log \left( \frac{1}{\delta} \right)}{T_i(t - 1)}} & T_i(t - 1) > 0,
\end{cases}
\]

and $\hat{\mu}_i(t - 1) = \frac{1}{T_i(t - 1)} \sum_{s \in [t - 1]} X_{s \pi_i} \mathbb{1}_{s \pi_i = i}$.
3. end while

Algorithm 2 first explores all arms exactly once, then estimates each arm using the upper bound of its $\delta$-confidence interval obtained from the Hoeffding inequality. Intuitively, the arm chosen in round $t$ either has a large sample mean or is under-explored. A sub-optimal arm is unlikely to be played long since its optimism bonus is decreasing to zero. The key ingredient is to select a good confidence level $\delta$, which balances the trade-off between exploration and exploitation.

Theorem 1.2.4. Set $\delta = n^{-2}$. The regret under UCB is given by

\[
R_n \leq 2 \sum_{i \in [k]} \Delta_i + \sum_{i: \Delta_i > 0} \frac{16 \log n}{\Delta_i}.
\]

Proof. Let $i^* \in [k]$ be an optimal arm. It is easy to see from the algorithm that the UCB estimate for $i$ is above its true mean $\mu_{i^*}$ with high probability, say, $1 - n\delta$. On the other hand, whenever a sub-optimal arm $i$ is played $T_i$ times, its optimism bonus is $\sqrt{2 \log(1/\delta)/T_i}$, which fails to compensate for the sub-optimality gap if $T_i$ is getting too large. Precisely, for $T_i(t - 1) \geq 8\Delta_i^{-2} \log(1/\delta)$,

\[
\mathbb{P} \left( \text{UCB}_i(t - 1, \delta) > \mu_{i^*} \right) \leq e^{-\frac{T_i(t - 1) \Delta_i^2}{8}} \leq \delta.
\]

Using a union bound and (1.1.2), we see that with probability at least $1 - (n + k - 1)\delta$,

\footnote{We shall see later that for many asymptotically optimal policies, it requires each sub-optimal arm to be played at least a given number of times in order to achieve sufficient accuracy.}
or $1 - 2n\delta$.

$$\bar{R}_n \leq 8 \log \left( \frac{1}{\delta} \right) \sum_{i: \Delta_i > 0} \frac{1}{\Delta_i}. $$

On the other hand, $\bar{R}_n$ is unconditionally bounded by

$$\bar{R}_n \leq n \max_{i \in [k]} \Delta_i \leq n \sum_{i \in [k]} \Delta_i. $$

Taking expectation separately yields

$$\mathbb{E}[R_n] \leq 2n^2 \delta \sum_{i \in [k]} \Delta_i + 8 \log \left( \frac{1}{\delta} \right) \sum_{i: \Delta_i > 0} \frac{1}{\Delta_i}. $$

The proof is completed by setting $\delta = n^{-2}$. \hfill \Box

**Remark 1.2.5** (Case-dependent bound). For fixed $\mu$, the regret incurred in the UCB algorithm satisfies

$$\limsup_{n \to \infty} \frac{R_n}{\log n} \leq \frac{1}{16} \sum_{i: \Delta_i > 0} \frac{1}{\Delta_i}. $$

(1.2.3)

This is quite surprising and implies that UCB defined above is consistent. We will provide more details on this in the next section.

**Remark 1.2.6** (Uniform bound). The regret bound in Theorem 1.2.4 is loose when $\Delta_i$ is small. To resolve this, we only apply the above analysis to $i \in [k]$ whose sub-optimality gap satisfies

$$16\Delta_i^{-2} \log n \leq \frac{n}{k},$$

which is equivalent to $\Delta_i \geq 4\sqrt{k \log n/n}$. The regret for arms with sub-optimal gaps less than $4\sqrt{k \log n/n}$ will be at most $4\sqrt{n k \log n}$. This allows us to derive an improved bound which is almost distribution-free:

$$R_n \leq 2 \sum_{i \in [k]} \Delta_i + 8\sqrt{n k \log n}. $$

(1.2.4)

As we will see later, (1.2.4) is almost optimal in the minimax sense.

**Remark 1.2.7** (Achieving optimality). The confidence level in Theorem 1.2.4 depends on horizon. A possible way to make $\delta$ anytime (independent of $n$) is to choose $\delta$ adaptively, say, $\delta = (1 + t \log^2 t)^{-1}$. One could prove a similar asymptotic bound as (1.2.3), but with an improved constant 2, which exactly matches the asymptotic lower bound in Theorem 1.3.2.

On the other hand, an algorithm termed the Minimax Optimal Strategy in the Stochastic case (MOSS), can be used to achieve minimax optimality, see [AB09; DP16]. This
may require choosing \( \delta \) in an arm-dependent fashion. For example, in [DP16], for arm \( i \) in round \( t \), \( \delta_{i,t} = \max\{1, n^2 \sum_{t' = 1}^{t-1} T_{i,t'} \} \). It was shown that the regret in this case satisfies

\[
R_n \leq \sum_{i \in [k]} \Delta_i + 38 \sqrt{kn}.
\]

As we will see, this is the best possible minimax lower bound (with a different constant) for stochastic bandits. However, to achieve optimality one may suffer from a significant increase in the variance of the pseudo-regret. More details can be found in Section 9.3 in [LS18].

**Remark 1.2.8** (Alternative estimators). In Algorithm 2, the Hoeffding inequality is used to compute the confidence intervals, which can be loose sometimes. For example, consider the Bernoulli bandits whose means are close to 0 or 1. In such situations, one could apply the Chernoff bound instead, which gives entropy-based asymmetric confidence intervals, see the KL-UCB algorithm in [GC11]. The corresponding regret will be improved by a taming factor (which usually depends on variance). Also, UCB works without the sub-gaussian assumption on rewards. One could replace the sample mean estimator by the median-of-means estimator, see the Exercise 2.2.9 in [Ver18]. For any distributions whose second moment exists, the regret is similar to what we have in Theorem 1.2.4 but with a worse constant factor.

We end this section by introducing two more algorithms: the \( \epsilon \)-Greedy algorithm and the Elimination algorithm. The \( \epsilon \)-Greedy algorithm is a non-deterministic algorithm which uses external randomization for exploration. The Elimination algorithm, which can be viewed as a multi-phase implementation of some existing algorithms, will reappear in later chapters.

We first introduce the \( \epsilon \)-Greedy algorithm:

**Algorithm 3:** The \( \epsilon \)-Greedy Algorithm

*Input:* \( (\epsilon_t)_{t \in [n]} \): exploration parameters; \( n \): horizon

*Output:* \( \pi_t = (\pi_t)_{t \in [n]} \)

1. if \( t \leq k \) then
2. \( \pi_t = t \)
3. else if \( k < t \leq n \) then
4. Choose \( \pi_t \) as \( \arg \max_{i \in [k]} \hat{\mu}_i(t-1) \) with probability \( 1 - \epsilon_t \), otherwise sample it uniformly from \([k]\), where \( \hat{\mu}_i(t-1) = \frac{1}{t(t-1)} \sum_{s \in [t-1]} x_{i,s} I_{s=t} \).
5. end if

As \( t \) becomes large, the chance of exploration should diminish to a good regret bound. By carefully choosing \( \epsilon_t \) as a decreasing function of \( t \), we can get a regret bound comparable to the previous algorithms. More details are summarized in the following theorem.

**Theorem 1.2.9.** Let \( \Delta = \min_{i: \Delta_i > 0} \Delta_i \) and \( R_n \) be the regret incurred in Algorithm 3. Then,
1.2. ALGORITHMS

- If \( \liminf_{t \to \infty} \epsilon_t = \epsilon > 0 \), then
  \[
  \liminf_{n \to \infty} \frac{R_n}{n} = \frac{\epsilon}{k} \sum_{i \in [k]} \Delta_i.
  \]

- If \( \epsilon_t = \min\{1, Ct^{-1} \Delta^{-2} k\} \), where \( C \) is some sufficiently large constant, then
  \[
  R_n \leq C' \sum_{i \in [k]} \left( \Delta_i + \frac{\Delta_i}{\Delta_i^2} \log \left( e \vee \frac{n \Delta_i^2}{\epsilon} \right) \right),
  \]
  where \( C' \) is some absolute constant.

The Elimination algorithm is a multi-phase version of the ETC algorithm. It divides the horizon \( n \) into several phases, with each one explored by an ETC algorithm. At the end of a phase, the arms which are identified sub-optimal are removed from the action set so that future exploration will only be on the arms that are likely to be optimal. The algorithm given below was first analyzed in [AO10].

**Algorithm 4: The Elimination Algorithm**

**Input:** \( L \): the number of phase; \( m_\ell \): phase exploration sequence

**Output:** \( \pi = (\pi_\ell)_{\ell \in [n]} \)

1. **Initialization:** \( A_1 = [k] \)
2. **for** \( \ell = 1, \ldots, L \) **do**
3. Choose each arm \( i \in A_\ell \) exactly \( m_\ell \) times
4. For \( i \in A_\ell \), compute the average reward \( \hat{\mu}_{i, \ell} \) for \( i \) from the current phase
5. \( A_{\ell+1} = \{ i \in A_\ell : \hat{\mu}_{i, \ell} + 2^{-\ell} \geq \max_{j \in A_\ell} \hat{\mu}_{j, \ell} \} \)
6. **end for**

**Theorem 1.2.10.** Choose \( m_\ell = 4^{\ell+1} \log(\ell (\ell + 1) kn) \). Then the regret in Algorithm 4 satisfies

\[
R_n \leq 2 \sum_{i \in [k]} \Delta_i + 128 \log n \sum_{i : \Delta_i > 0} \frac{1}{\Delta_i}.
\]

**Proof.** Let \( i^* \in [k] \) be an optimal arm. It is easy to check that with high probability, \( i^* \) is never eliminated:

\[
\mathbb{P} \left( i^* \notin A_\ell \text{ for some } \ell \leq L \right) \leq \sum_{\ell=1}^{L-1} \mathbb{P} \left( i^* \notin A_{\ell+1} | i^* \in A_\ell \right)
\]

\[
\leq \sum_{\ell=1}^{L-1} k e^{-\frac{m_\ell 4^{-\ell}}{4}} \leq \frac{1}{n}.
\]

From now on condition on the event that \( i^* \) is never eliminated. For any sub-optimal \( i \),
define \( \ell_i = \min \{1 \leq \ell \leq L : 2^{-\ell} \leq \Delta_i / 2\} \). Then,
\[
\mathbb{P} \left( \text{There exists an sub-optimal } i \text{ s.t. } i \in \mathcal{A}_{\ell_i+1} \right)
\leq \sum_{i : \Delta_i > 0} \mathbb{P} \left( i \in \mathcal{A}_{\ell_i+1} \right) \leq \sum_{i : \Delta_i > 0} e^{-\frac{n \cdot \Delta_i \ell_i}{4}} \leq \frac{1}{n}.
\]
Therefore, with probability at least \( 1 - 2/n \), a sub-optimal arm \( i \) is played at most
\[
M_i = \sum_{i=1}^{\ell_i} m_i = \sum_{i=1}^{\ell_i} 4^{\ell_i} \log(\ell_i + 1) n k)
\leq \sum_{i=1}^{\ell_i} 3 \cdot 4^{\ell_i+1} \log(k n) \leq \frac{128}{\Delta_i^2} \log n,
\]
where for convenience we assume that \( \max \{\ell_i + 1, k\} \leq n \). Therefore,
\[
R_n \leq \frac{2}{n} \cdot n \sum_{i=1}^{\ell_i} \Delta_i + 128 \log n \sum_{i : \Delta_i > 0} \frac{1}{\Delta_i}
= 2 \sum_{i \in [k]} \Delta_i + 128 \log n \sum_{i : \Delta_i > 0} \frac{1}{\Delta_i}.
\]
The proof is complete.

**Remark 1.2.11.** Similar to Remark 1.2.6, one can obtain a universal regret bound for the regret in Algorithm 4:
\[
R_n = O \left( \sum_{i \in [k]} \Delta_i + \sqrt{nk \log n} \right).
\]

Using almost the same proof but with a different choice on \( m_i \), one can improve \( R_n \) to
\[
R_n = O \left( \sum_{i \in [k]} \Delta_i + \sqrt{nk \log k} \right).
\]

### 1.3 Lower Bounds on Regret

#### 1.3.1 Minimax Lower Bound

We introduce the minmax lower bound in this section, which can help us understand the best universal behavior of stochastic bandit algorithms. Let \( \mathcal{E} \) be the environment class consisting of the admissible \( \mu \). The worst-case regret for a policy \( \pi \) is defined as
\[
R_n(\pi, \mathcal{E}) := \sup_{\mu \in \mathcal{E}} R_n(\pi).
\]
1.3. LOWER BOUNDS ON REGRET

The minimax regret $R^*_n$ is the best worst-case regret among the possible policies:

$$R^*_n := \inf_{\pi} R_n(\pi, \mathcal{E}).$$

(1.3.1)

A good lower bound for (1.3.1) will illustrate the limit in stochastic bandits learning. We introduce a natural way to approach this problem. The general idea is to show that for any $\pi \in \Pi$, there exist two bandits $\mu, \mu' \in \mathcal{E}$ such that $R_n,\mu(\pi)$ and $R_n,\mu'(\pi)$ cannot both be small at the same time. Intuitively, a good instance for $\mu$ is bad for $\mu'$, and vice versa. This often implies that the $\mu$ and $\mu'$ are distant from each other in the parameter space. On the other hand, $\pi$ should not distinguish $\mu$ from $\mu'$ so that $E_\mu \approx E_{\mu'}$, which requires $\mu$ to be close to $\mu'$. Therefore, a trade-off position is desired.

**Theorem 1.3.1.** Let $\mathcal{E}$ be the class of normalized Gaussian bandits with mean $\mu = (\mu_i)_{i \in [k]} \in [0, 1]^k$. Suppose that $n \geq k - 1$. Then for any policy $\pi$ there exists some $\mu \in \mathcal{E}$ such that

$$R_n,\mu(\pi) \geq \frac{1}{27} \sqrt{(k - 1)n}.$$  

(1.3.2)

There are several ways to prove this theorem. Here we describe two of them inspired from the Le Cam’s method and the Assouad’s method, which are standard techniques in the minimax theory of statistics, see [Yu97].

**Proof based on Le Cam’s method.** Fix a policy $\pi$. Let $\Delta \in [0, 1/2]$ be some parameter to be tuned later. Consider two bandits defined as $\mu = (\Delta, 0, \cdots, 0)$ and $\mu' = \mu$ except at $i$, where $\mu_i = 2\Delta$, and $i$ is the least-played arm in $\mu$ on average: $i = \arg\min_{j > 1} E_j[T_j(n)]$. Using the Chebyshev inequality,

$$R_n,\mu(\pi) + R_n,\mu'(\pi) \geq \frac{n\Delta}{2} \left( P_{\mu} \left( T_1(n) \leq \frac{n}{2} \right) + P_{\mu'} \left( T_1(n) > \frac{n}{2} \right) \right).$$

(1.3.3)

The right-hand side of (1.3.3) can be bounded via the Bretagnolle-Huber inequality.

$$R_n,\mu(\pi) + R_n,\mu'(\pi) \geq \frac{n\Delta}{2} e^{-D_{KL}(P_\mu, P_{\mu'})} \geq \frac{n\Delta}{4} e^{-\frac{2n\Delta^2}{k-1}}.$$  

(1.3.4)

where the second inequality follows from

$$D_{KL}(P_\mu, P_{\mu'}) = \sum_{j \in [k]} E_j[T_j(n)] D_{KL}(\mu_j, \mu'_j) = E_{\mu}[T_1(n)] D_{KL}(\mu_i, \mu'_i) \leq \frac{2n}{k-1} \Delta^2.$$  

(1.3.5)

Taking $\Delta = \sqrt{(k - 1)/4n}$ yields the desired result. 

---

3The Bretagnolle-Huber inequality states that for two probability measures $P$ and $Q$ on the same probability space $(\Omega, \mathcal{F})$, and $A \in \mathcal{F}$,

$$P(A) + Q(A^c) \geq \frac{1}{2} e^{-\min\{D_{KL}(P, Q)D_{KL}(Q, P)\}}.$$  

(1.3.4)

where $D_{KL}(\cdot, \cdot)$ is the KL-divergence.
The constant 1/27 in the bound (1.3.2) can be improved to 1/9 via Assouad’s method when \( k \geq 3 \):

**Proof based on Assouad’s method.** Instead of considering only two arms as in the previous proof, we consider \( k \) arms that are mutually inconsistent. Let \( \mu^{(0)} \in \mathcal{E} \) be the all-zero vector and for \( i \in [k] \), define \( \mu^{(k)}_i \) as the vector that is \( \Delta \) at \( i \) and 0 elsewhere. Comparing \( \mu^{(i)} \) to \( \mu^{(0)} \) using the Pinsker’s inequality

\[
\frac{1}{n} \left( \mathbb{E}_{\mu^{(0)}}[T_i(n)] - \mathbb{E}_{\mu^{(i)}}[T_i(n)] \right) \leq 2\delta(\mathbb{P}_{\mu^{(0)}}, \mathbb{P}_{\mu^{(i)}}) \leq \sqrt{D_{KL}(\mathbb{P}_{\mu^{(0)}}, \mathbb{P}_{\mu^{(i)}})} = \sqrt{\frac{\Delta^2}{2} \mathbb{E}_{\mu^{(0)}}[T_i(n)]}.
\]

Summing over \( i \) together with Jensen’s inequality yields

\[
\sum_{i \in [k]} \mathbb{E}_{\mu^{(0)}}[T_i(n)] \leq n + n\sqrt{\frac{\Delta^2}{2} nk}.
\]

Therefore,

\[
\max_{i \in [k]} R_{n,\mu^{(i)}}(\pi) \geq \frac{1}{k} \sum_{i \in [k]} R_{n,\mu^{(i)}}(\pi) \geq \Delta \left( n - \frac{1}{k} \sum_{i \in [k]} \mathbb{E}_{\mu^{(0)}}[T_i(n)] \right) \geq n\Delta \left( \frac{k - 1}{k} - \sqrt{\frac{n\Delta^2}{2k}} \right).
\]

Taking \( \Delta = \sqrt{k/2n} \) completes the proof.

## 1.3.2 Instance-Dependent Lower Bound

We consider instance-dependent lower bound for policies which are known to be ‘good’. First of all, let us clarify what we mean by a good policy. A policy \( \pi \) is consistent over \( \mathcal{E} \) if for all \( \mu \in \mathcal{E} \) and \( p > 0 \),

\[
\lim_{n \to \infty} \frac{R_{n,\mu}(\pi)}{n^p} = 0.
\]

Roughly speaking, the regret of a consistent policy is less than polynomial growth. Denote by \( \Pi_c(\mathcal{E}) \) the set of consistent policies over \( \mathcal{E} \). Assuming that \( \mathcal{V} \) is unstructured, the following theorem suggests that the best one can hope for consistent policies is logarithmic growth.

**Theorem 1.3.2.** Suppose \( \mathcal{E} = \prod_{i \in [k]} \mathcal{E}_i \) and fix \( \mu \in \mathcal{E} \). Let \( i^* \) be an optimal arm. For any \( \pi \in \Pi_c(\mathcal{E}) \),

\[
\liminf_{n \to \infty} \frac{R_{n,\mu}(\pi)}{\log n} \geq \sum_{i : \Delta_i > 0} \frac{\Delta_i}{d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E})}. \tag{1.3.7}
\]

---

\(^4\)The Pinsker’s inequality states that for two probability measures \( P \) and \( Q \) on the same probability space \( (\Omega, \mathcal{F}) \), their total-variation distance (\( \delta(\cdot, \cdot) \)) is bounded by their KL-divergence (\( D_{KL}(\cdot, \cdot) \)),

\[
\delta(P, Q) \leq \sqrt{\frac{1}{2} \min\{D_{KL}(P, Q), D_{KL}(Q, P)\}}. \tag{1.3.6}
\]
1.4. THOMPSON SAMPLING

where
\[ d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E}) = \inf_{\nu \in \mathcal{E} : \nu > \mu_{i^*}} D_{KL}(\mu_i, \nu). \]

**Proof.** By (1.1.2), it suffices to prove that for \( i \in [k] \) with nonzero sub-optimality gap,
\[ \liminf_{n \to \infty} \mathbb{E}_\mu[T_i(n)] \geq \frac{1}{d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E})}. \]

To show this, fix \( \varepsilon > 0 \). Consider \( \nu \in \mathcal{E} \) that agrees with \( \mu \) except at the \( i \)-th arm, which equals \( \nu_i \) such that \( D_{KL}(\mu_i, \nu_i) < d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E}) + \varepsilon \) for some \( \varepsilon > 0 \). Applying the Bretagnolle-Huber inequality with \( A = \{ T_i(n) \leq n/2 \} \),
\[ R_{n,\mu}(\pi) + R_{n,\nu}(\pi) \geq \frac{n}{2} \min \{ \Delta_i, \nu_i - \mu_{i^*} \} e^{-d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E}) + \varepsilon} \mathbb{E}_\mu[T_i(n)]. \] (1.3.8)

Since \( \pi \) is consistent in both \( \mu \) and \( \nu \), for any \( p > 0 \) and \( \varepsilon > 0 \), there exists a sufficiently large \( N \) such that for \( n > N \),
\[ R_{n,\mu}(\pi) + R_{n,\nu}(\pi) \leq \varepsilon n^p. \] (1.3.9)

Plugging (1.3.9) into (1.3.8) with some simplification yields
\[ \mathbb{E}_\mu[T_i(n)] \geq \frac{\log \left( \frac{n}{2} \min \{ \Delta_i, \nu_i - \mu_{i^*} \} \right) - \log \varepsilon - p \log n}{d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E}) + \varepsilon}. \]

Dividing both sides by \( \log n \) and setting \( n \to \infty \), then taking \( p, \varepsilon \to 0 \) finishes the proof. \( \square \)

**Remark 1.3.3.** Theorem 1.3.2 immediately implies that instance optimality when \( \delta_i = (1 + t \log^2 t)^{-1} \). Indeed, let \( \mathcal{E} \) be the class of unstructured Gaussian bandits with unit variance. For any sub-optimal \( i \), \( d_{\text{inf}}(\mu_i, \mu_{i^*}, \mathcal{E}) = \Delta_i^2/2 \) (achieved at \( \nu \) with \( \nu_i \) arbitrarily close to \( \mu_{i^*} \)). By Theorem 1.3.2,
\[ \liminf_{n \to \infty} \frac{R_{n,\mu}(\pi)}{\log n} \geq 2 \sum_{i : \Delta_i > 0} \frac{1}{\Delta_i}. \]

Combining this with Remark 1.2.7 proves the instance optimality of the choice of \( \delta_i \) in the algorithm.

### 1.4 Thompson Sampling

In this section, we introduce Thompson sampling in the context of stochastic bandits. Thompson sampling [Tho33] is well acknowledged as the first bandit algorithm which is named after the Canadian entomologist William R. Thompson. The idea behind it is similar to UCB, but adopts a Bayesian point of view. Assuming the bandit is selected from a parametric family according to a prior, Thompson sampling uses the posterior computed from the explored data to guide future exploration. Despite being
simple, Thompson sampling has not received much attention in the past eight decades. One possible reason is the long-existing gap between the empirical performance and theoretical guarantee, which was recently filled by the work [AG12] in 2012, where a frequentist-type analysis is provided for the finite-time regret. Further works on its asymptotic optimality as well as the Bayesian regret can be found in [KKM12] and [RR14], respectively. These results reinforced and further stimulated the resurgence of interest in Thompson sampling in many academic communities. Here, we will focus on some intuitive ideas behind Thompson sampling and provide an upper bound for its Bayesian regret. The discussion is based on the chapter 36 in [LS18] and [RR14].

Let $\mathcal{E}$ be the class of environments. The frequentist view on stochastic bandits is that the true bandit $\mu \in \mathcal{E}$ is fixed and unknown. In contrast, the Bayesian perspective assumes that $\mu$ is sampled from $\mathcal{E}$ according to a prior $\mathcal{Q}$ before the game starts, that is, $\mu \sim \mathcal{Q}$. The additional information on $\mathcal{Q}$ can hopefully help us decipher the true bandit through updated posterior. Of course, conditional on $\mathcal{Q}$, the cumulative regret is the same as before; however, since $\mu$ is random based on our assumption on the model, it would be more natural to consider the averaged regret defined as

$$R_{n, BR} = \mathbb{E}_\mathcal{Q}[R_{n, \mu}],$$

(1.4.1)

where $R_{n, \mu}$ is defined in (1.1.1). (1.4.1) is referred to as the Bayesian regret. Bayesian optimality (as opposed to the minimax optimality) is measured by the Bayesian regret. A major appeal of the Bayesian framework is that the optimal Bayesian strategy can often be described, if not always computed. This result is closely related to the Gittins indices which we will discuss in detail in chapter 5.

Thompson sampling can be described as follows. Assuming data $x_{ss}$ has been observed for $s \leq t$, one can compute the posterior distribution on $\mu$ using the prior: $\mathcal{Q}(\mu|x_{ss}, 1 \leq s \leq t)$. Thompson sampling directly draws a new bandit according to the posterior, and sets $\pi_{t+1}$ as the optimal arm in the random realization. This randomization forces exploration until the posterior is concentrated, in which case it is expected to close to $\mu$. The details of the algorithm are as follows:

**Algorithm 5:** Thompson Sampling

- **Input:** $n$: horizon; $\mathcal{Q}$: prior distribution on the bandit
- **Output:** $\pi = (\pi_t)_{t \in [n]}$

1. for $t = 1, \ldots, n$
2. Sample $\nu \sim \mathcal{Q}(.|x_{ss}, 1 \leq s \leq t-1)^t$
3. $\pi_t = \text{arg max}_{i \in [k]} \nu_i$
4. end for

*Parameters $\mu$ may not always represent the true mean in different models. However, it is always the case that the true mean is a function (link function) of $\mu$. So in general one should sample $\nu$ from the posterior distribution of the mean. Here for simplicity, we just take the function to be identity for illustration.*

Before analyzing Algorithm 5, we give some intuition why the posterior distribution is helpful for inference on $\mu$. For convenience we assume that the bandit is unstructured. Let $f_i(x; \nu)$ be the density of the reward of arm $i$ with parameter $\nu$. For $i \in [k]$, let $n_i$ be the number of pulls on $i$, and denote the corresponding rewards by $z_{i1}, \ldots, z_{in_i}$. We assume that $n_i$’s are deterministic. In this case, the posterior distribution $p$ given the
explored data is
\[ p(v) \propto \mathbb{Q}(v) \prod_{i \in [k]} \prod_{t \in [n_i]} f_i(z_{it}; v_i). \]

Therefore,
\[
\log \frac{p(\mu)}{p(v)} = \log \frac{\mathbb{Q}(\mu)}{\mathbb{Q}(v)} + \sum_{i \in [k]} \sum_{t \in [n_i]} \log \frac{f_i(z_{it}; \mu_i)}{f_i(z_{it}; v_i)} \\
\approx \log \frac{\mathbb{Q}(\mu)}{\mathbb{Q}(v)} + \sum_{i \in [k]} n_i \mathbb{E}_{z \sim f_i(\cdot; \mu_i)} \left[ \log \frac{f_i(z; \mu_i)}{f_i(z; v_i)} \right] \\
= \log \frac{\mathbb{Q}(\mu)}{\mathbb{Q}(v)} + \sum_{i \in [k]} n_i D_{KL}(\mu_i, v_i),
\]

so that
\[
\frac{p(\mu)}{p(v)} \approx \frac{\mathbb{Q}(\mu)}{\mathbb{Q}(v)} e^{\sum_{i \in [k]} n_i D_{KL}(\mu_i, v_i)}.
\]

where \( D_{KL}(\cdot, \cdot) \) is the KL-divergence. This observation has the following consequences:

- When \( i \) has been pulled many times, the \( i \)-th marginal of \( p(v) \) is concentrated, thus encouraging future pulls to explore other directions. When all arms have been well-explored, \( p(v) dv \approx \delta_\mu \).

- The convergence rate of \( p(v) dv \) to \( \delta_\mu \) would be slow if \( \mathbb{Q}(\mu) \) is small, so the bandit in this case is hard to learn from the posterior, hence a large \( R_{n,\mu} \) is expected. On the other hand, \( \mathbb{Q}(\mu) \) being small suggests that \( \mu \) is unlikely to be obtained under \( \mathbb{Q} \). When averaging over \( \mathbb{Q} \), \( R_{n,\mu} \) should be not too large.

These observations give some informal justification for the promising performance of Thompson sampling. The next theorem provides a rigorous upper bound for the Bayesian regret under appropriate assumptions:

**Theorem 1.4.1.** In the Bayesian bandit set-up, assume that for \( v \in \mathcal{E} \) and \( i \in [k] \), \( v_i \) has its mean in \([0, 1] \) and is 1-sub-gaussian after centering. If \( k \leq n \log n / 4 \), then the Bayesian regret in Thompson sampling is bounded by

\[ R_{n,\text{BR}} \leq 9 \sqrt{kn \log n}. \]

**Proof.** Let \( i^* = \arg \max_{i \in [k]} \mu_i \) and \( \mathcal{F}_i = \sigma(\pi_1, x_{1,\pi_1}, \ldots, \pi_i, x_{i,\pi_i}) \). The Bayesian regret can be written as

\[ R_{n,\text{BR}} = \sum_{i \in [n]} \mathbb{E}[\mu_i^* - \mu_{\pi_i}]. \quad (1.4.2) \]

Conditional on \( \mathcal{F}_{i-1} \), \( i^* \) and \( \pi_i \) have the same distribution\(^5\) (arg max of two conditional independent samples from the posterior). \((1.4.2)\) can therefore be bounded via centralization. Fix \( \delta > 0 \). A good choice is to use UCB\(_{\pi_i}(t - 1, \delta)\), which has the same

\[^5\]It should be warned that \( \mu_{\pi_i} \) and \( \mu_{\pi_i} \) have different distributions since the former is always no less than the latter.
distribution as \( UCB_i(t-1, \delta) \) given \( P_{t-1} \). Therefore, it follows from the tower property that

\[
R_{n, \text{BR}} = \sum_{i \in [n]} \left( \mathbb{E}[\mu_i - UCB_i(t-1, \delta)] + \mathbb{E}[UCB_{x_t}(t-1, \delta) - \mu_i] \right).
\]

With probability at least \( 1 - kn\delta \), for \( i \in [k] \) and \( t \in [k] \),

\[
0 \leq UCB_i(t-1, \delta) - \mu_i \leq 2 \sqrt{\frac{2 \log \left( \frac{1}{\delta} \right)}{T_i(t-1)}}.
\]

Therefore,

\[
R_{n, \text{BR}} \leq 2kn^2\delta + \sum_{i \in [k]} \int_{0}^{T_i(n)} 2 \sqrt{\frac{2 \log \left( \frac{1}{\delta} \right)}{x}} \, dx \leq 2kn^2\delta + \sum_{i \in [k]} \sqrt{32n T_i(n) \log \left( \frac{1}{\delta} \right)} \leq 2kn^2\delta + \sqrt{32nk \log \left( \frac{1}{\delta} \right)}.
\]

where the last step follows from Jensen’s inequality. Setting \( \delta = n^{-2} \) completes the proof. \( \square \)

**Remark 1.4.2.** As opposed to frequentist-based methods, Thompson sampling is often more robust when delay of observation happens. Also, from a computational perspective, Thompson sampling is relatively more convenient when the subroutine of frequentist-based algorithms requires solving complicated optimization problems, see Remark 2.3.5.

**Remark 1.4.3.** The idea of using direct sampling from posterior distributions can be further extended. For example, one may consider seeking an optimistic estimate for the true mean as in the UCB algorithm, which can be done by calculating the quantile of the marginal posterior for each arm. Such idea first appeared in [KCG12] and the
corresponding algorithm is called the Bayes-UCB:

**Algorithm 6: Bayes-UCB**

**Input:** $n$: horizon; $Q$: prior distribution on the bandit; $c$: parameters of the quantile

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>for $t = 1, \ldots, n$ do</td>
</tr>
<tr>
<td>2:</td>
<td>for $i = 1, \ldots, k$ do</td>
</tr>
<tr>
<td>3:</td>
<td>Calculate the posterior marginal $Q_i$ for arm $i$ from $Q(</td>
</tr>
<tr>
<td>4:</td>
<td>$q_i = \max \left{ r : Q_i((-\infty, r]) \leq 1 - \frac{1}{t} \frac{1}{\log n} \right}$ // Calculate the $(1 - t^{-1}(\log n)^{-c})$-quantile for arm $i$</td>
</tr>
<tr>
<td>5:</td>
<td>end for</td>
</tr>
<tr>
<td>6:</td>
<td>$\pi_t = \arg \max_{i \in [k]} q_{it}$</td>
</tr>
<tr>
<td>7:</td>
<td>end for</td>
</tr>
</tbody>
</table>

Analysis of Algorithm 6 in the frequentist setting for Bernoulli bandits with Beta priors is given in [KCG12]. The idea follows from our intuition provided before but was made rigorous using the tail bounds for Beta distributions. Numerical results in the paper demonstrated a superior performance of Bayes-UCB over some other frequentist-based algorithms. It also showed that the Bayes-UCB strategy, when used with a sparsity-inducing prior, can be numerically approximated using Markov Chain Monte Carlo simulations.
Chapter 2

Stochastic Linear Bandits

2.1 Some Generalization

Recall the definition of regret given in (1.1.1). In many practical situations, not only is the chosen reward observed by the player but also the side information associated with it. Such extra knowledge makes the model more complicated and as a consequence, the best possible policy in hindsight could be more optimal than the one used in (1.1.1). In this chapter, we will extend stochastic bandits to a more general class of bandit problems where side information can be appropriately included, and understand their learning strategies. The discussion closely follows the one in [LS18]. As a starter, let us consider a stochastic model where contextual information is available.

Contextual structures can be made of good use in bandit learning. Based on the classical stochastic bandit model, we further assume that players have access to the present context before selecting an arm in each round. Precisely, let $\mathcal{C}$ be the set of contexts and $\pi = (\pi_t)_{t \in [n]} \in [k]^n$ be a policy. $\pi_t$ is measurable with respect to the past decisions/rewards as well as the contexts revealed up to time $t$. The rewards collected by the player are

$$x_t = r(c_t, \pi_t) + \eta_t \quad t \in [n],$$

(2.1.1)

where $c_t \in \mathcal{C}$ (deterministic or random) is the context in round $t$, $r(\cdot, \cdot) : \mathcal{C} \times [k] \to \mathbb{R}$ is the reward function$^1$, and $\eta_t$ is the noise satisfying

$$\mathbb{E}[e^{\delta \eta_t}|c_1, \pi_1, x_1, \cdots, c_t, \pi_t] \leq e^{\frac{\delta^2}{2}}.$$  

(2.1.2)

It is easy to check from (2.1.2) that $\eta_t$ is sub-gaussian conditional on the history. (2.1.1) and (2.1.2) together lead to the so-called stochastic contextual bandits. Due to the lack

---

$^1$The reward function, which is the mean of the reward distribution, is analogous to $\mu$ in chapter 1 but has two indices.
of knowledge of \( r \), the regret is characterized by
\[
R_n = \mathbb{E} \left[ \sum_{c \in C} \max_{i \in [k]} r(c, i) \right] - \mathbb{E} \left[ \sum_{t \in [n]} x_t \right].
\] (2.1.3)

The regret given in (2.1.3) implicitly agrees that the best policy in each round is to play the arm with the largest expected reward in the present context, which is most meaningful if the dependence between contexts and actions is mild. A counterexample is when an optimal action at the beginning leads to unfavorable contexts in the subsequent rounds. The ideal \( c_t \) under our consideration is either deterministic or independent of the past actions. Conditional on \( (c_t)_{t \in [n]} \), \( R_n \) becomes a sum of regret of \( \Phi((\theta, \phi(c, i))) \), where \( \Phi \) is the link function, see [Fil+10; LLZ17]. We will briefly discuss this in a remark at the end of next section.

We next discuss some policies in stochastic linear bandits, depending on different assumptions on \( \mathcal{A}_t \). Note that \( \mathcal{A}_t \) is finite from the perspective of stochastic contextual bandits and the kernel map. However, in the general set-up of stochastic linear bandits, \( \mathcal{A}_t \) is allowed to be infinite. Regardless of varying assumptions on \( \mathcal{A}_t \), the regret defined
in (2.1.5) is based on the lack of information of \( r \) so that one always needs to efficiently estimate for \( \theta \) in the exploration process. Nevertheless, the efficiency here can have different meanings for different types of \( \mathcal{A}_t \), which needs to be discussed in a case-dependent manner.

### 2.2 Stochastic Linear Bandits with a Finite and Fixed Action Set

We begin our analysis for stochastic linear bandits whose action set \( \mathcal{A}_t = \mathcal{A} \) does not change with time. We also assume that \( |\mathcal{A}| = k \). Let \( a_t \in \mathcal{A} \) be the feature vector associated with the action under some policy \( \pi \) at time \( t \). The exploration process suffices to estimate \( \theta \), and a natural estimator is the least-squares estimator. One may use the Ridge estimator if having invertibility issues. Precisely, in round \( t \), \( \theta \) is estimated by \( \hat{\theta}_t \), where \( \hat{\theta}_t \) solves the normal equations:

\[
\begin{pmatrix}
(a_1, \ldots, a_{t-1}) \\
A_{t-1}^T
\end{pmatrix} \begin{pmatrix}
\hat{\theta}_t \\
A_{t-1}
\end{pmatrix} = \begin{pmatrix}
(x_1) \\
A_{t-1}
\end{pmatrix}.
\]

Note that the design matrix \( A_{t-1} \) depends on the past actions. Since \( \mathcal{A} \) is fixed, a good policy in the exploration stage should efficiently estimate the marginals of \( \theta \) along the directions in \( \mathcal{A} \). In other words, for any \( \epsilon, \delta > 0 \), we wish to find a \( \pi \) such that

\[
T_{\pi}(\epsilon, \delta) := \inf \left\{ t : \max_{a \in \mathcal{A}} \mathbb{P}\left( |\langle \hat{\theta}_t - \theta, a \rangle| > \epsilon \right) \leq \delta \right\}
\]

(2.2.1)

is as small as possible. This is equivalent to the optimal experimental design problem in statistics. To see this, note that for fixed \( a \in \mathcal{A} \),

\[
\mathbb{P}\left( |\langle \hat{\theta}_t - \theta, a \rangle| > \sqrt{2\|a\|_{V_{t-1}}^2 \log \left( \frac{2}{\delta} \right)} \right) \leq \delta,
\]

where \( V_{t-1} = A_{t-1}^T A_{t-1} \). Setting

\[
\sqrt{2\|a\|_{V_{t-1}}^2 \log \left( \frac{2}{\delta} \right)} \leq \epsilon
\]

and solving for \( t \) yields

\[
\|a\|_{V_{t-1}} \leq \sqrt{\frac{\delta^2}{2} \log^{-1} \left( \frac{2}{\delta} \right)}.
\]

(2.2.2)
2.2. STOCHASTIC LINEAR BANDITS WITH A FINITE AND FIXED ACTION SET

Since $V_t^{-1} \preceq V_{t-1}^{-1}$, an upper bound for (2.2.1), which we denote by the same notation, is

$$T_\epsilon(x, \delta) = \inf \left\{ t : \max_{a \in A} \|a\|_{V_t^{-1}} \leq \sqrt{\frac{\epsilon^2}{2} \log^{-1} \left( \frac{2}{\delta} \right)} \right\}. \tag{2.2.3}$$

Let $P(A)$ be set of probability measures on $A$. There is a one-to-one correspondence between policies and $P(A)$ asymptotically (The frequency of choosing an action $a$ in $t$ rounds under $\pi$ can be roughly viewed as the probability $p(a)$). Denote by $p$ the probability measure associated with $\pi$. It is easy to verify that

$$T_\epsilon(x, \delta) = \max_{a \in A} \|a\|_{A^{-1}(p)}^2 + 1 \quad A(p) := \mathbb{E}_p[aa^T]$$

Hence, as $t \to \infty$, minimizing (2.2.3) over the possible policies is equivalent to the following optimization problem:

$$\min_{p \in P(A)} \max_{a \in A} \|a\|_{A^{-1}(p)}^2. \tag{2.2.4}$$

The term $\max_{a \in A} \|a\|_{A^{-1}(p)}^2$ is often denoted by $g(p)$, and (2.2.4) is referred to as the $G$-optimal design problem. (2.2.4) can be solved using the following theorem due to Kiefer and Wolfowitz [KW85]:

**Theorem 2.2.1** (Kiefer-Wolfowitz). Let $A \subset \mathbb{R}^d$ be a compact set and $\text{span}(A) = \mathbb{R}^d$. Then

$$p^* \text{ solves } (2.2.4) \iff p^* \in \arg \max_{p \in P(A)} \log \det A(p) \iff g(p^*) = d.$$ 

Moreover, there exists a $p^*$ such that $\text{supp}(p^*) \leq d(d + 1)/2$.

**Remark 2.2.2.** The geometric interpretation of the set $\left\{ \|x\|_{A^{-1}(p)} \leq d \right\}$ based on duality is the minimum volume enclosing ellipsoid for the set $A$. When $A$ is convex, this is known as John’s ellipsoid.

**Remark 2.2.3.** According to Theorem 2.2.1, solving the $G$-optimal design problem is the same as finding the maximizer for $\log \det A(p)$, which is a convex optimization problem and can be approximately solved via Frank-Wolfe algorithm: For $a \in A$ and $a_k = \arg \max_{a \in A} \|a\|_{A^{-1}(p_k)}^2$,

$$p_{k+1}(a) = \left( 1 - \frac{1}{\beta} \left\| a_k \|_{A^{-1}(p_k)} - 1 \right\| \right) p_k(a) + \frac{1}{\beta} \left\| a_k \|_{A^{-1}(p_k)} - 1 \right\| a=a_k.$$ 

If $p_0$ is a uniform distribution on $A$, then $g(p_k) \leq (1 + \epsilon)d$ for $k = O(d \log \log |A| + d/\epsilon)$.
Once (2.2.4) is (approximately) solved, one can use the solution to construct a near-optimal policy for exploration. Algorithms in Section 1 like ETC naturally extend to this case. However, similar problems still exist as the best trade-off point usually depends on the sub-optimality gaps. An alternative approach is to take the exploration adaptively together with a phase-elimination process, which is given below:

**Algorithm 7: Phase-elimination with G-optimal Exploration Algorithm**

**Input:** $A$: action set; $\delta$: confidence level

**Output:** $A_{1 \to n}$

1. **Initialization:** $\ell = 0$, $A_1 = A$, $T_0 = 0$, $d_0 = d$ and $N_0 = 0$
2. **while** $N_{\ell} \leq n$ **do**
3. $\ell = \ell + 1$
4. $p_\ell \in \arg\max_{p \in P(A_\ell)} \log \det A_\ell(p)$ and $\text{supp}(p_\ell) \leq d_{\ell-1}(d_{\ell-1} + 1)/2$
5. $d_\ell = \min \{ |A_\ell|, d \}$
6. **for** $a \in A_\ell$ **do**
7. $T_\ell(a) = 2\varepsilon^{-2}d_\ell p_\ell(a) \log \left( \frac{2k\ell(\ell + 1)}{\delta} \right)$
8. **end for**
9. $T_\ell = \sum_{a \in A} T_\ell(a)$
10. **for** $t \in [N_{\ell-1} + 1, N_{\ell-1} + T_\ell]$ choose action $a \in A_\ell$ $T_\ell(a)$ times. Denote the design matrix by $A_\ell$
11. $\hat{\theta}_\ell = (A_\ell^T A_\ell)^{-1} A_\ell^T(x_{T_\ell+1}, \ldots, x_{T_\ell})^T$
12. $A_{\ell+1} = \{ a \in A_\ell : \max_{b \in A_\ell} \langle \hat{\theta}_\ell, b - a \rangle \leq 2\varepsilon_\ell \}$
13. $N_\ell = N_{\ell-1} + T_\ell$
14. **end while**

Assume additionally that the largest sub-optimality gap is bounded by 1. Then the regret analysis of Algorithm 7 can be summarized in the following theorem:

**Theorem 2.2.4.** The psudo-regret $\tilde{R}_n = n \max_{a \in A} \langle \theta, a \rangle - \sum_{t \in [n]} \langle \theta, a_t \rangle$ in Algorithm 7 satisfies

\[
\Pr \left( \tilde{R}_n > 4 \sqrt{ \left( 16n \log \left( \frac{nk}{\delta} \right) + d \right) d } \right) \leq \delta.
\]

Moreover, taking $\delta = n^{-1}$ and using that $\tilde{R}_n \leq n$,

\[
R_n \leq 4 \sqrt{ \left( 32n \log \left( \frac{nk}{\delta} \right) + d \right) d }.
\]

**Proof.** Algorithm 7 guarantees that with a high probability, an optimal action $a^*$ is
2.2. STOCHASTIC LINEAR BANDITS WITH A FINITE AND FIXED ACTION SET

always in \( A \) and sub-optimal arms can only survive finitely many times. In fact,
\[
P\left(a^* \notin A_{\ell+1} \mid a^* \in A_\ell\right) \\
\leq \sum_{\ell \in \mathbb{N}} \sum_{a \in A_{\ell+1} \cap A_\ell} \left( P\left(|\hat{\theta}_{\ell+1} - \theta, a^*| > \varepsilon_{\ell+1}\right) + P\left(|\hat{\theta}_{\ell+1} - \theta, a| > \varepsilon_{\ell+1}\right) \right) \leq \frac{\delta}{k}.
\]

On the other hand, let \( \Delta_a = (\theta, a^* - a) \) be the sub-optimality gap between \( a \) and \( a^* \) and define \( \ell_a = 2 + \left\lceil \log \Delta_a - 1 \right\rceil \) (base 2). Then,
\[
P\left(a \in A_{\ell_a}\right) \leq P\left(|\hat{\theta}_{\ell_a} - \theta, a^*| > \varepsilon_{\ell_a}\right) + P\left(|\hat{\theta}_{\ell_a} - \theta, a| > \varepsilon_{\ell_a}\right) \leq \frac{\delta}{k}.
\]

Taking a union bound gives
\[
P\left(a^* \in A_\ell \text{ for all } a \in A \text{ with } \Delta_a > 0 \right) \geq 1 - \delta. \quad (2.2.5)
\]

We now estimate \( \bar{R}_n \) conditional on the event defined in (2.2.5). Let \( \Delta > 0 \) be some parameter to be tuned later. We first consider the actions played in the first \( n \) rounds with sub-optimality gap less than \( \Delta \). It is clear that the cumulative loss incurred by these actions is bounded by \( n \Delta \). On the other hand, the actions with sub-optimality gap greater than \( \Delta \) can be played only in the first \( \ell_\Delta \) rounds, where \( \ell_\Delta = 2 + |\log \Delta^{-1}| \) (base 2). The regret of these actions is bounded by the total loss incurred in the first \( \ell_\Delta \) rounds of exploration, which is further bounded \(^2\) by
\[
\sum_{\ell = 1}^{\log n / 4 \ell_\Delta} 4 \varepsilon_{\ell} T_{\ell} \leq \frac{64}{\Delta} d \log \left( \frac{2k \log n}{\delta} \right) + 4d.
\]

Therefore,
\[
\bar{R}_n \leq \inf_{0 < \Delta < 1} \left\{ n \Delta + \frac{64}{\Delta} d \log \left( \frac{2k \log n}{\delta} \right) + 4d \right\} \leq 16 \sqrt{nd \log \left( \frac{2k \log n}{\delta} \right)} + 4d
\]
\[
\leq 32 \sqrt{nd \log \left( \frac{k \log n}{\delta} \right) + d} \leq \sqrt{16 n \log \left( \frac{nk}{\delta} \right) + d} d.
\]

The proof is thus complete. \( \Box \)

**Remark 2.2.5.** One advantage of Algorithm 7 is that it behaves well even when the model is misspecified. Particularly, if the observation \( x_i \) is contaminated by some noise of magnitude less than \( \varepsilon \), then it can be proved that
\[
R_n = O \left( \sqrt{nd \log(nk)} + \varepsilon n \sqrt{d \log n} \right).
\]

\(^2\)Conditional on (2.2.5), the largest sub-optimality gap in the \( \ell \)-th round is at most \( 4\varepsilon_{\ell} \) when \( \ell > 1 \). The same holds also for \( \ell = 1 \) since \( 1 \leq 4\varepsilon_1 \).
Remark 2.2.6. Algorithm 7 can be viewed as the generalized version of Algorithm 4 for (linear) structured stochastic bandits. The uniform exploration in each phase is replaced with an effective allocation given by the Kiefer-Wolfowitz Theorem. Indeed, Theorem 2.2.4 implies Theorem 1.2.10 by taking $\mathcal{A}$ as canonical bases in $\mathbb{R}^d$ and $d = k$.

2.3 Stochastic Linear Bandits with General Action Sets

We now consider stochastic linear bandits with general action sets, meaning that $\mathcal{A}_t$ can be both time-varying and infinite. This requires us to find all-rounded confidence sets adaptively. Suppose for now that $\theta$ is still estimated by the least-squares estimator. A possible choice for confidence sets is a sequence of balls weighted by the design matrices, similar to the ones used in the classical linear regression. However, the analysis here is more intricate since the design matrix is data-dependent. We also provide a different approach to constructing the confidence sets based on online learning prediction, which allows one to utilize the prior knowledge of $\theta$ in the construction. Except for the confidence sets, the general algorithm resembles the UCB algorithm in Section 1. This implies that $\theta$ is estimated stepwise and the decision is made under optimistic bias. When the exploration data is small, the design matrix may not be invertible. Therefore, it is thus necessary to use regularized least-squares estimators to avoid invertibility issues (at the beginning). Particularly, let $\lambda > 0$ be a regularization parameter and our estimation for $\theta$ at time $t$ is given by

$$
\hat{\theta}_t = \left( \lambda I + A_{t-1}^T A_{t-1} \right)^{-1} A_{t-1}^T \begin{pmatrix} x_1 \\
\vdots \\
x_{t-1} \end{pmatrix}.
$$ (2.3.1)

Suppose that the confidence set at $t$ is an $V_{t-1}(\lambda)$-weighted ball centered at $\hat{\theta}_t$\(^3\), i.e.,

$$
C_t = \{ z \in \mathbb{R}^d : \| z - \hat{\theta}_t \|_{V_{t-1}(\lambda)} \leq \beta_t \},
$$

where $\beta_t \geq 1$ is an increasing sequence computed from $A_1, a_1, x_1, \ldots, A_{t-1}, a_{t-1}, x_{t-1}$. Suppose also that for fixed $n$ and $\delta$, the choice of $\beta_t$ makes $\{C_t\}_{t\in[\ell]}$ form a uniform\(^4\) $(1 - \delta)$-confidence set:

$$
\mathbb{P}(\theta \in \cap_{t=1}^n C_t) \geq 1 - \delta.
$$ (2.3.2)

\(^3\)Taking $\hat{\theta}_t$ as the regularized least-squares estimator is convenient and gives heuristics on the shape of the confidences sets. Other reasonable estimators also work in the following analysis.

\(^4\)The reason why one needs uniform confidence sets is because the algorithm is built adaptively so that estimations at different steps are different.
Practicality of (2.3.2) will be justified in a moment. A version of the UCB algorithm for stochastic linear bandits can then be described as follows:

**Algorithm 8:** UCB Algorithm for Stochastic Linear Bandits (LinUCB)

**Input:** \( \{A_t\}_{t \in [n]} \): action sets; \( \delta \): confidence level; \( \lambda \): regularization parameter

**Output:** \( (a_t)_{t \in [n]} \)

1: **Initialization:** \( t = 1, V_1(\lambda) = \lambda I \) and \( a_1 \in A_1 \)
2: **while** \( t \leq n \) **do**
3: \( \hat{\theta}_{t+1} = V_t^{-1}(\lambda)A_t^T(x_1, \cdots, x_t)^T \)
4: \( C_{t+1} = \{ z \in \mathbb{R}^d : \| z - \hat{\theta}_{t+1} \|_{V_t(\lambda)}^2 \leq \beta_{t+1} \} \)
5: \( a_{t+1} \in \arg\max_{a \in A_{t+1}} \sup_{z \in C_{t+1}} \langle z, a \rangle \)
6: \( V_{t+1}(\lambda) = V_t(\lambda) + a_{t+1}a_t^T \)
7: \( t = t + 1 \)
8: **end while**

Algorithm 8 is also known as Linear Reinforcement Learning (LinRel)[DHK08; RT10]. Before discussing in detail how \( \beta_t \)'s are chosen, we give a general result on the asymptotic regret that Algorithm 8 can achieve, as summarized in the following theorem:

**Theorem 2.3.1.** Assume that for some \( L > 0 \),

\[
\sup_{a \in \bigcup_{t \in [n]} \mathcal{A}_t} \left\{ |\langle \theta, a \rangle|, \frac{\| a \|_2}{L} \right\} \leq 1. \tag{2.3.3}
\]

For convenience we also assume that \( \mathcal{A}_t \) is closed. Then, with probability at least \( 1 - \delta \), the pseudo-regret \( \bar{R}_n = \sum_{t=1}^n \sup_{a \in A_t} \langle \theta, a - a_t \rangle \) in Algorithm 8 satisfies that

\[
\bar{R}_n \leq 2 + \sqrt{8n\beta_n \log \left( \frac{\det V_n(\lambda)}{\det V_1(\lambda)} \right)} \leq 2 + \sqrt{8d n\beta_n \log \left( \frac{d \lambda + nL^2}{d \lambda} \right)}. \tag{2.3.4}
\]

**Proof.** It suffices to check that (2.3.4) holds on the event defined in (2.3.2). The assumptions in the theorem imply that \( \mathcal{A}_t \) is compact thus there exists \( z_t \in \mathcal{A}_t \) such that \( \langle z_t, a_t \rangle = \sup_{z \in \mathcal{A}_t} \langle z, a_t \rangle \). Then,

\[
\bar{R}_n = \sum_{t=1}^n \sup_{a \in A_t} \langle \theta, a - a_t \rangle \overset{(2.3.3)}{\leq} 2 + \sum_{t=2}^n \min_{\theta \in A_t} \left\{ \sup_{a \in A_t} \langle \theta, a - \langle z_t, a_t \rangle \rangle + \langle z_t - \theta, a_t \rangle \right\} \leq 0 \quad \text{on (2.3.2)}
\]

\[
\overset{(2.3.2)}{\leq} 2 + 2 \sum_{t=2}^n \min_{a \in A_t} \left\{ \sqrt{\beta_n \| a_t \|_{V_{t-1}(\lambda)}^2}, 1 \right\} \beta_n \leq \beta_n \leq 2 + 2 \sum_{t=2}^n \min_{a \in A_t} \left\{ \sqrt{\beta_n \| a_t \|_{V_{t-1}(\lambda)}^2}, 1 \right\}
\]

\[
\overset{\beta_n \geq 1}{\leq} 2 + 2 \sqrt{n\beta_n \sum_{t=2}^n \log (1 + \| a_t \|_{V_{t-1}^2})} \leq 2 + 2 \sqrt{2n \beta_n \sum_{t=2}^n \log (1 + \| a_t \|_{V_{t-1}^2})}.
\]
CHAPTER 2. STOCHASTIC LINEAR BANDITS

where the last step uses the inequality \( \min\{1, u\} \leq 2 \log(1 + u) \) for \( u \geq 0 \). Since \( V_t \) is obtained from \( V_{t-1} \) through a rank-1 update, it is easy to check that

\[
\prod_{t=2}^{n} \left( 1 + \|a_t\|_{V_{t-1}}^2 \right) = \frac{\det V_n(\lambda)}{\det V_1(\lambda)}.
\]

Plugging this into the above estimate completes the proof of the first part in (2.3.4). The second part follows immediately by noting that \( \det V_n(\lambda) \leq (\text{tr}(V_n(\lambda))/d)^d \leq (\lambda + nL^2/d)^d \).

\[\square\]

Remark 2.3.2. The constant 2 comes from exploring the first action arbitrarily. It does not affect the asymptotic result and will be omitted later for convenience.

Remark 2.3.3. The above proof shows that how \( \theta \) is estimated in (2.3.1) does not really matter, as long as the confidence set is weighted by \( V_{t-1} \) (where \( V_{t-1} \) is updated in the rank-1 manner) and centered at the estimation. One may use other reasonable estimators in both theoretical analysis and practice, see section 2.3.2.

Remark 2.3.4 (Generalized stochastic linear bandits). Similar idea could be extended to generalized stochastic linear bandits. Suppose that the conditional reward of an action \( a \) is given by \( \Phi(\langle a, \theta \rangle) \), where \( \Phi \) is the link function. In this case, \( \theta \) could be estimated by the quasi-maximum likelihood estimate \( \hat{\theta}_{t+1} \):

\[
\sum_{a \in \mathcal{A}} \left( x_t - \Phi(\langle a, \hat{\theta}_{t+1} \rangle) \right) a_t = 0. \tag{2.3.5}
\]

However, in order to apply Algorithm 8 one needs to obtain confidence sets as in (2.3.2), which is difficult for the MLE in practice. An alternative approach, as observed in [Fil+10], is that the solution to the optimization in step 5 in Algorithm 8 is equivalent to \( \arg \max_{a \in \mathcal{A}_{t+1}} \Phi(\langle a, \hat{\theta}_{t+1} \rangle) + \sqrt{\frac{1}{t+1} \||a||_{V_{t-1}(\lambda)}^2}} \) when \( \Phi \) is identity. This leads to the so-called UCB-GLM algorithm, which is given below:

Algorithm 9: UCB-GLM Algorithm

\[\begin{align*}
\text{Input:} & \quad \{\mathcal{A}_t\}_{t \in [n]}: \text{action sets; } \beta_t: \text{exploration parameter; } \lambda: \text{regularization parameter} \\
\text{Output:} & \quad (a_t)_{t \in [n]} \\
1: & \quad \text{Initialization: } t = 1, V_1(\lambda) = \lambda I \text{ and } a_1 \in \mathcal{A}_1 \\
2: & \quad \text{while } t \leq n \text{ do} \\
3: & \quad \text{Solve (2.3.5) for } \hat{\theta}_{t+1} \\
4: & \quad a_{t+1} \in \arg \max_{a \in \mathcal{A}_{t+1}} \Phi(\langle a, \hat{\theta}_{t+1} \rangle) + \sqrt{\frac{1}{t+1} \||a||_{V_{t-1}(\lambda)}^2}} \\
5: & \quad V_{t+1}(\lambda) = V_t(\lambda) + a_{t+1}a_{t+1}^T \\
6: & \quad t = t + 1 \\
7: & \quad \text{end while}
\end{align*}\]

The specific choice for \( \beta_t \) with theoretical guarantee can be found in [Fil+10; LLZ17]. The penalty term can be viewed as the variance, so that exploration is encouraged when the variance is large.

\[\begin{align*}
5 & \text{This is the true maximum likelihood estimate under appropriate exponential family assumptions.}
\end{align*}\]
Remark 2.3.5 (Thompson sampling for stochastic linear bandits). Thompson sampling can also be used for stochastic linear bandits. Assuming that the true parameter is generated from some distribution $Q$, then the algorithm can be described as follows:

Algorithm 10: Thompson Sampling for Stochastic Linear Bandits

**Input:** $n$: horizon; $Q$: prior distribution on the bandit

**Output:** $(a_t)_{t \in [n]}$

1: for $t = 1, \cdots, n$ do
2: Sample $\theta \sim Q(\cdot| a_s, x_s, 1 \leq s \leq t - 1)$
3: $a_t = \arg \max_{a \in A_t} \langle \theta, a \rangle$
4: end for

We next give two ways of finding valid $\beta_t$'s to achieve (2.3.2), both of which use the maximal inequality for self-normalized martingales.

### 2.3.1 Laplace’s Method

Define $S_t = A_t^T (\eta_1, \cdots, \eta_n)^T$. Note that for $t \in [n]$,

$$
\| \hat{\theta}_t - \theta \|_{V_{t-1}^{-1}(\lambda)} = \langle V_{t-1}^{-1}(\lambda) \lambda \theta - S_{t-1}, \lambda \theta - S_{t-1} \rangle^{1/2} = \| \lambda \theta - S_{t-1} \|_{V_{t-1}^{-1}(\lambda)}^{1/2} \\
\leq \lambda \| \theta \|_{V_{t-1}^{-1}(\lambda)} + \| S_{t-1} \|_{V_{t-1}^{-1}(\lambda)}^{1/2} \leq \sqrt{\lambda} \| \theta \|_2 + \| S_{t-1} \|_{V_{t-1}^{-1}(\lambda)}^{1/2}.
$$

(2.3.6)

Therefore it suffices to obtain a high-probability bound for $\| S_{t-1} \|_{V_{t-1}^{-1}(\lambda)}$. The weight factor is given by the inverse of $V_{t-1}^{-1}(\lambda)$ which can be difficult to analyze. To get around we rewrite $\frac{1}{2} \| S_t \|_{V_t^{-1}(\lambda)}^2$ using the Legendre transform:

$$
\frac{1}{2} \| S_t \|_{V_t^{-1}(\lambda)}^2 = \max_{x \in \mathbb{R}^d} \left( \langle x, S_t \rangle - \frac{1}{2} \langle x, x \rangle_{V_t^{-1}(\lambda)} \right).
$$

Thanks to the sub-gaussian assumption on the noise, for fixed $x$, $M_t(x) = e^{\langle x, S_t \rangle - \frac{1}{2} \langle x, x \rangle_{V_t^{-1}}}$ is a non-negative supermartingale started at $e^{-\| x \|_2^2/2}$. This raises hope for a uniform bound in $t$ (as desired in (2.3.2)) by appealing to the maximal inequality. However, there is an extra maximum over $x$ which needs to be addressed first. For this we use a technique called self-normalization. Self-normalization is similar to Laplace’s method in asymptotic analysis. Note that $\| \lambda \theta + S_{t-1} \|_{V_{t-1}^{-1}(\lambda)}$ grows to infinity as $t \to \infty$ (otherwise there is no need to do the analysis here!). For large $t$, if one wants to understand $\max_{x \in \mathbb{R}^d} \log M_t(x)$, one may study

$$
\bar{M}(x) := \log \int_{x \in \mathbb{R}^d} M_t(x) p(x) dx
$$

instead, where $p(x)$ is some finite-measure whose support contains $\arg \max_{x \in \mathbb{R}^d} M_t(x)$. On the other hand, if $p(x)$ is a probability measure, averaging over $x$ preserves the
martingale property so that $\bar{M}(t)$ is also a non-negative supermartingale. Take $p(x)$ as a gaussian measure on $\mathbb{R}^d$ with covariance matrix $\lambda^{-1}I$. Then,

$$\bar{M}_t = \left( \frac{\lambda^d}{\det(\lambda I + V_t(\lambda))} \right)^{1/2} e^{\frac{1}{2} \|S_t\|_{V_t(\lambda)}^2} \geq \left( \frac{\lambda^d}{\det(2V_t(\lambda))} \right)^{1/2} e^{\frac{1}{4} \|S_t\|_{V_t^{-1}(\lambda)}^2}.$$

By Doob’s maximal inequality for non-negative supermartingales,

$$\mathbb{P}\left( \sup_{t \in \mathbb{N}} \|S_t\|_{V_t^{-1}(\lambda)} \geq 2 \sqrt{\log \left( \frac{1}{\delta} \right)} + \frac{1}{2} \log \left( \frac{\det V_t(\lambda)}{\lambda^d} \right) \right) \leq \mathbb{P}\left( \sup_{t \in \mathbb{N}} \bar{M}_t \geq \log \left( \frac{1}{\delta} \right) \right) = \mathbb{P}\left( \sup_{t \in \mathbb{N}} \bar{M}_t \geq \frac{1}{\sqrt{2d/\delta}} \right) \leq 2^{d/2} \delta \mathbb{E}[\bar{M}_0] = \delta.$$ 

This together with (2.3.6) implies that $\beta_i$ in (2.3.2) can be chosen as

$$\sqrt{\beta_i} = \sqrt{\lambda} \|\theta\|_2 + 2 \sqrt{\log \left( \frac{1}{\delta} \right)} + \frac{1}{2} \log \left( \frac{\det V_t(\lambda)}{\lambda^d} \right) \leq \sqrt{\lambda} \|\theta\|_2 + 2 \sqrt{\log \left( \frac{1}{\delta} \right)} + \frac{d}{2} \log \left( \frac{d \lambda + nL^2}{d \lambda} \right).$$

(2.3.7) Assuming $\|\theta\|_2 \leq \Theta(\sqrt{d})$, plugging (2.3.7) into (2.3.4) and integrates the tail probability yields that

$$R_n = \Theta \left( d \sqrt{n \log(nL)} \right).$$

### 2.3.2 Online Learning Prediction

Algorithm 8 (with $\hat{\theta}$ being treated as a general estimator) is closely related to the problem of online learning prediction. In online learning prediction, given an action, the learner predicts the reward using some predictor built from the history before observing the reward. The true reward is then fed back to the learner to update the predictor. In the following we will use $\hat{x}_t$ as the predicted reward at time $t$, with its accuracy is measured by the cumulative squared-loss $\sum_{t \in [n]} (\hat{x}_t - x_t)^2$. In Algorithm 8, one still has the predictor beforehand. What makes it different is the goal of learning. One needs to decide on their own which action to play and the way how this is done is by computing multiple predicted values using appropriate confidence sets.

In linear stochastic bandits, the best theoretic predictor is given by the expected reward function. For any predictor $\hat{x}_t$, define the regret of $\hat{x}_t$ relative to a linear predictor $z \in \mathbb{R}^d$ as

$$\rho_n(z) = \sum_{t \in [n]} (\hat{x}_t - x_t)^2 - \sum_{t \in [n]} (\langle z, a_t \rangle - x_t)^2.$$
2.3. STOCHASTIC LINEAR BANDITS WITH GENERAL ACTION SETS

where $a_t$ are given (either by a policy or some mysterious person). For any prior knowledge on $z$ such that $z \in \Theta \subset \mathbb{R}^d$, we say that the learner enjoys guarantee $B_n$ on $\Theta$ if $\sup_{z \in \Theta} \rho_n(z) \leq B_n$. As we will see, $B_n$ plays a crucial role in our construction of confidence sets, and allows some side-information on $\Theta$ to be incorporated.

From now on we assume that $\hat{x}_t$ is given to estimate the reward. For example, in Algorithm 8 $\hat{x}_t(\cdot) = \langle \hat{\theta}_t, \cdot \rangle$. Since the confidence sets ask for an estimate for the mean reward, for any sequence of actions $(a_t)_{t \in [n]}$, we consider $Q_n := \sum_{t \in [n]} (\hat{x}_t - \langle \theta, a_t \rangle)^2$. Using the Cosine theorem,

\[
Q_n = \rho(\theta) + 2 \sum_{t \in [n]} \eta_t(\hat{x}_t - \langle \theta, a_t \rangle) \leq B_n + 2 \sum_{t \in [n]} \eta_t(\hat{x}_t - \langle \theta, a_t \rangle) .
\]

To further bound $Q_n$ uniformly for $n \in \mathbb{N}$, observe that $e^{x^2 - \frac{1}{2}Q_t}$ is a non-negative supermartingale for every $x \in \mathbb{R}$. Using a similar self-normalization trick as in the previous analysis and setting $x = 1$ yields that

\[
P \left( \forall t \in \mathbb{N} \text{ s.t. } |m_t| \leq \sqrt{1 + Q_t} \log \left( \frac{1 + Q_t}{\delta^2} \right) \right) \geq 1 - \delta . \quad (2.3.8)
\]

Conditional on the event defined in (2.3.8), for every $t \in \mathbb{N}$,

\[
Q_t \leq B_t + 2 \sqrt{1 + Q_t} \log \left( \frac{1 + Q_t}{\delta^2} \right) .
\]

which implies

\[
Q_t \leq 1 + 2B_t + 32 \log \left( \frac{\sqrt{8 + \sqrt{1 + B_t}}}{\delta} \right) . \quad (2.3.9)
\]

Although (2.3.9) is a confidence set, it may be unbounded and does not fit into the form used in Algorithm 8. To solve this, we restrict the confidence sets to a ball $B(L)$ of radius $L$ using (2.3.3), then identify an estimator for $\theta$ based on the given predictor $\hat{x}_t$ and derive a confidence set around it. Let $\hat{\theta}_t$ be the regularized least-squares solution for the predicted values:

\[
\hat{\theta}_t = V_t^{-1}(\lambda) A_t^T \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_t \end{pmatrix} .
\]

$B_n$ is a random quantity depending on data.
We can then rewrite $Q_t + \lambda \| \hat{\theta} \|_2^2$ as

\[
\| A_t \theta - \tilde{A}_t \hat{\theta}_t + A_t \hat{\theta}_t - \hat{X}_t \|_2^2 + \lambda \| \theta \|_2^2
- 2 \lambda \langle \theta - \hat{\theta}_t, \hat{\theta}_t \rangle_{V_t(\theta)} + 2 \langle \theta - \hat{\theta}_t, \hat{\theta}_t \rangle_{V_t(\theta)} + \lambda \| \theta \|_2^2
\]

\[
= \| \theta - \hat{\theta}_t \|_{V_t(\theta)}^2 + \| A_t \hat{\theta}_t - \hat{X}_t \|_2^2 + \lambda \| \hat{\theta}_t \|_2^2.
\]

It is then clear that

\[
\begin{cases}
\{ z \in \mathbb{R}^d \cap B(L) : Q_t \leq 1 + 2B_t + 32 \log \left( \frac{\sqrt{8} + \sqrt{1 + B_t}}{\delta} \right) \} \\
\subset \{ z \in \mathbb{R}^d : Q_t + \lambda \| z \|_2^2 \leq 1 + 2B_t + 32 \log \left( \frac{\sqrt{8} + \sqrt{1 + B_t}}{\delta} \right) + \lambda L^2 \} \\
\subset \{ z \in \mathbb{R}^d : \| \theta - \hat{\theta}_t \|_{\tilde{V}_t(\theta)}^2 \leq 1 + 2B_t + 32 \log \left( \frac{\sqrt{8} + \sqrt{1 + B_t}}{\delta} \right) + \lambda L^2 \}.
\end{cases}
\]

This together with (2.3.4) implies that

\[
R_n = \mathcal{O} \left( \sqrt{nd \mathbb{E}[B_n] + L^2} \log(nL) \right) \quad \text{if } L = \Theta(1) = \left( \sqrt{nd \mathbb{E}[B_n] \log(nL)} \right).
\]

If $\theta$ is known to be $s$-sparse, we can take $\Theta = \{ z \in \mathbb{R}^d : \| z \|_0 \leq s \}$. In this case, it can be shown that $\mathbb{E}[\beta_n] \leq s \log n$. Hence we have $R_n = \mathcal{O} \left( \sqrt{nds \log^2(nL)} \right)$.

**Remark 2.3.6.** When building up the confidence sets in the sparsity case, one needs to know the sparsity level in advance, which determines the choice for $B_t$.

### 2.4 Stochastic Linear Bandits on Graphs

An interesting application of stochastic linear bandits in graph learning is the spectral bandit [Val+14]. Other related models will be summarized in the final remark at the end of this section. Let $G = (V, E)$ be an undirected graph with $d$ vertices. For convenience, we write $V = [d]$. Let $\mathcal{W}, D \in \mathbb{R}^{d \times d}$ be the similarity matrix and degree
matrix, respectively. The unnormalized Laplacian of $G$ is defined as $\mathcal{L} = D - W$. In a social network represented by $G$, $W$ measures the connections between the nodes and $\mathcal{L}$ measures the network diffusion dynamics. To formulate a bandit learning problem, we assume that $V$ is the complete action set. At each step $t$, a subset $V_t \subset V$ is selected for the player to choose an action from. This should be differentiated from the model where players are assigned with a node and he/she needs to choose an action from the given context [CGZ13]. The expected reward by pulling $i$ is given by $r_i$, with its fluctuation characterized by a $1$-sub-gaussian distribution. As before, the goal is to find a strategy such that after some time $n$, the cumulative regret (which is defined in (2.1.5)) is small. Without further assumption, this problem is nothing but a $d$-armed bandits with time-varying action sets, which can be tackled using Algorithm 8.

In many practical situations, however, the topological properties of $G$ dictate the potential shape of $r$. For example, a smoothness assumption requires that $r$ changes smoothly within the clusters of $V$, which are depicted by the spectrum of $\mathcal{L}$. To incorporate such information, we hypothesize that $r$ is a smooth graph function, which states that large components of the eigen-representation of $r$ under $\mathcal{L}$ are concentrated on the left end of the spectrum. Precisely, let $\mathcal{L} = Q\Lambda Q^T$ be the eigen-decomposition of $\mathcal{L}$, where $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d)$ with $0 = \lambda_1 \leq \cdots \leq \lambda_d$, and $Q = (q^{(1)}, \cdots, q^{(d)})$. Then $r$ can be written as

$$r = \sum_{i \in [d]} \theta^{(i)} q^{(i)}$$

and for $i \in [d],

$$r_i = \langle \theta, \bar{q}^{(i)} \rangle$$

where $\bar{q}^{(i)}$ is the $i$-th row vector of $Q$. The smoothness assumption on $r$ can be stated as $\|r\|_\Lambda$ being small. Since $\| \cdot \|_\Lambda$ is not a well-defined norm ($\lambda_1 = 0$), we therefore use the same trick as before and consider $\|r\|_{\Gamma}$ where $\Gamma = \Lambda + \lambda I$ for some small $\lambda > 0$. Let $A$ be the set of row vectors of $Q$. The smoothness assumption is equivalent to estimating $\theta$ via a ridge estimator penalized by an anisotropic norm term:

$$\hat{\theta}_t = \left( \Gamma + A_{t-1}^T A_{t-1} \right)^{-1} A_{t-1}^T \begin{pmatrix} x_1 \\ \vdots \\ x_{t-1} \end{pmatrix}$$

(2.4.1)

combined with Algorithm 8 lead to the so-called Spectral UCB algorithm, which

\footnote{One can think of $r \in \mathbb{R}^d$ and node $i$ corresponds to the canonical vector $e_i \in \mathbb{R}^d$.}
Algorithm 11: Spectral UCB

**Input:** $\mathcal{L}$: Laplacian matrix of the graph; $n$: horizon; $\lambda$: regularization parameter; $\delta$: confidence level

**Output:** $(a_t)_{t \in [n]}$

1. Compute the eigen-decomposition of $\mathcal{L}$: $\mathcal{L} = Q\Lambda Q^T$ (with eigenvalues arranged in increasing order). Let $\mathcal{A}$ denote the set of row vectors in $Q$.

2. **Initialization:** $t = 1, V_1 = I + \Lambda$ and $a_1 \in \mathcal{A}$

3. **while** $t \leq n$ **do**

4. Receive the action set $\mathcal{A}_{t+1} \subseteq \mathcal{A}$ in round $t+1$

5. $\hat{V}_{t+1} = V_t^{-1}(\lambda)A_t^T(x_1, \ldots, x_t)^T$

6. $C_{t+1} = \{z \in \mathbb{R}^d : \|z - \hat{V}_{t+1}\|_{V_t(\lambda)}^2 \leq \beta_{t+1}\}$

7. $a_{t+1} \in \arg\max_{a \in \mathcal{A}_{t+1}} \sup_{z \in C_{t+1}} \langle z, a \rangle$

8. $V_{t+1}(\lambda) = V_t(\lambda) + a_{t+1}a_{t+1}^T$

9. $t = t + 1$

10. **end while**

Assuming that the expected reward on the graph is bounded, say between $-1$ and $1$, and with $\beta_t$ chosen as in (2.3.7) to satisfy (2.3.2), the first part of (2.3.4) still holds, see Remark 2.3.3. In this case, with probability at least $1 - \delta$, the pseudo-regret $\tilde{R}_n$ in Algorithm 11 is bounded by

$$\tilde{R}_n \leq O\left(\sqrt{n} \log \left(\frac{\det V_n(\lambda)}{\delta \det V_1(\lambda)}\right)\right).$$

(2.4.2)

Directly bounding the log-determinant ratio by $O(d \log(1/\delta))$ is not optimal because $d$ is often dominant in real social networks ($d \gg n$). What can be gained by adding $\Lambda$ in the penalty is that one can derive a graph-dependent bound for the log-determinant ratio using the effective dimension. For a variety of networks of interest, the effective dimension can be computed explicitly and is much smaller than $d$.

**Definition 2.4.1** (Effective dimension). The effective dimension of $\Gamma = \lambda I + \Lambda$ is defined as

$$d_{\text{eff}} = \max \left\{ d : (d - 1)\lambda_d \leq \frac{n}{\log(1 + \frac{n}{\lambda})} \right\},$$

where $\lambda_d$ is the $d$-th smallest eigenvalue of $\mathcal{L}$.

An intuitive interpretation on the effective dimension is that it finds where the spectral gap occurs thus giving the number of clusters in the graph. Since the unknown vector only changes mildly within clusters, the effective dimension represents the true complexity in terms of learning. The following lemma allows us to bound $\log(\det V_n(\lambda)/\det V_1(\lambda))$ using $d_{\text{eff}}$:

**Lemma 2.4.2.** Let $d_{\text{eff}}$ be the effective dimension given Definition 2.4.1. Then,

$$\log \left(\frac{\det V_n(\lambda)}{\det V_1(\lambda)}\right) \leq 2d_{\text{eff}} \log \left(1 + \frac{n}{\lambda}\right).$$
2.4. STOCHASTIC LINEAR BANDITS ON GRAPHS

Proof. We will use the fact that \( \det V_n(\lambda) \) is maximized only if \( a_i \)'s are along the axes (Lemma 5 in [Val+14]). Hence,

\[
\log \left( \frac{\det V_n(\lambda)}{\det V_1(\lambda)} \right) \leq \max_{\sum_{j=1}^{n} t_j \geq 0} \sum_{i=1}^{d} \log \left( 1 + \frac{t_i}{\lambda + \lambda_i} \right),
\]

where we used that \( \|a\|_2 = 1 \) for \( a \in A \). Fix \( t_i \geq 0 \) such that \( \sum_{i \in [d]} t_i = n \). It is easy to see that

\[
\sum_{i \leq \text{eff}} \log \left( 1 + \frac{t_i}{\lambda + \lambda_i} \right) \leq d_{\text{eff}} \log \left( 1 + \frac{n}{\lambda} \right)
\]

\[
\sum_{i > \text{eff}} \log \left( 1 + \frac{t_i}{\lambda + \lambda_i} \right) \leq \sum_{i > \text{eff}} \log \left( 1 + \frac{d_{\text{eff}} \log(1 + \frac{n}{\lambda})}{n} \right) \leq d_{\text{eff}} \log \left( 1 + \frac{n}{\lambda} \right).
\]

Combining the two estimates gives the desired the result. \( \square \)

Lemma 2.4.2 and (2.4.2) together with \( \delta = n^{-1} \) implies that

\[
R_n \leq \Theta \left( \sqrt{n d_{\text{eff}} \log \left( \frac{n}{\lambda} \right)} \right).
\]

Remark 2.4.3. In the [Val+14], \( \beta_i = (2 \sqrt{d_{\text{eff}} \log(1 + t/\lambda)} + 2 \log(1/\delta) + c)^2 \), where \( c \) is the upper bound on \( \|	heta\|_C \). Another algorithm based on the idea of elimination was also proposed, and can be proved to achieve a better regret bound.

Remark 2.4.4. There are a few other bandit models on networks utilizing the underlying graph structures. In [CGZ13], a model called A Gang of Bandits was introduced. There they assumed that each node corresponds to an independent stochastic linear bandit, and the parameters between neighboring nodes are close under the Euclidean distance. This assumption is similar to the smoothness condition in spectral bandits, i.e., the parameters are changing smoothly with respect to the graph metric. An algorithm called GOB.Lin was proposed in the same paper to factor in the smoothness assumption. Particularly, a graph Laplacian induced kernel trick was to used to map both contexts and parameters to a larger space, so that inference of the parameter on one node implies information on the others.

Both spectral bandits and a gang of bandits take the similarity matrix as given. In a subsequent work [GLZ14], the authors considered the case when pre-knowledge on such graph structure is absent, and one needs to learn it while exploring. Following the same set-up as in [CGZ13], an algorithm called the Cluster of Bandits (CLUB) was proposed, combining the ideas of LinUCB and hypothesis testing. Start by assuming the graph is a perfect cluster, or equivalently, \( G \) is a complete graph. At each step, the player receives a node and then chooses the best action based on LinUCB, with the estimated

---

\[ \text{Note: } \text{One may think this as a multi-dimensional version of stochastic bandits, where each arm is associated with a } d\text{-dimensional parameter.} \]
parameter chosen as the estimated cluster parameter associated with the node. Once a new reward is observed, one can use it to update the individual parameter of the node as well as the corresponding cluster parameter. If one notices the difference between the individual parameters of two nodes in the same cluster above some threshold (which depends on the exploration level of the nodes), then remove the edge between them. The process yields an updated graph which will be used in the next round.

### 2.5 Instance-Dependent Lower Bound

We now discuss the inevitable regret for any learning strategy in stochastic linear bandits. As in Section 1, there are two directions to answer this question. The first is to consider the minimax regret for fixed horizon \( n \), which comes down to analyzing policies whose worst regret over a class of bandits is optimal. The idea for this approach is similar to the study of stochastic bandits. However, identifying the environments that are mutually incoherent for a given policy is more complicated due to the varying geometry of the action set. We shall not go into the details here. The take-away message is the conclusion: For various types of \( A \) (independent of \( t \)), the regret incurred by applying Algorithm 8 is near-optimal (only differ by a log factor). From this perspective, the universal behavior if optimistic algorithms is almost optimal.

The other way is to consider the asymptotic instance lower bound for a consistent policy. In this case, one first fixes a bandit then sends \( n \) to infinity. This approach provides a theoretical limit of learning in infinite horizon, yet it may obscure some humongous constants that often matter in practice. The main result we are going to introduce in this section quantifies the asymptotic lower bound for consistent policies. The bound, which is solved by an optimization problem, is achievable by a well-designed policy (not guaranteed as finite-time optimal so may not be useful in practice). As a disturbing consequence, it can be shown that optimistic policies can be far from being asymptotically optimal, see [LS16].

The following theorem shows that any sub-optimal arm must be fully explored under any consistent policy.

**Theorem 2.5.1.** Consider stochastic linear bandits where noise is standard Gaussian. Assume that \( A \subset \mathbb{R}^d \) is finite, independent of time and spans \( \mathbb{R}^d \). Let \( \theta \in \mathbb{R}^d \) be the true parameter and \( (a_t)_{t \in \{1, \ldots, n\}} \) be a sequence of actions under a consistent policy over the complete environments \( \mathbb{R}^d \). Suppose that there is a unique optimal action \( a^* \) for \( \theta \). Define \( \tilde{G}_n = E[\sum_{t \in \{1, \ldots, n\}} a_t a_t^T] \). Then for any \( a \in A \setminus a^* \),

\[
\limsup_{n \to \infty} \log n \|a\|^2_{\tilde{G}_n^{-1}} \leq \frac{\Delta_a^2}{2},
\]

where \( \Delta_a = \langle \theta, a^* - a \rangle \) is the sub-optimality gap of \( a \).

---

9The estimated cluster parameter can be thought as a statistic calculated under the current null hypothesis. If some actions form a cluster, then one can treat the past exploration data on each action equally and combine them together to estimate for the common parameter shared by the actions. This should be close to the estimate on each individual arm if the hypothesis is true.
Note that the left-hand side of (2.5.1) appears as the marginal error in the \((1 - 1/n)\)-confidence interval of the least-squares estimator of \((\theta, a)\), see (2.2.2). A global consistent thus needs to estimate every sub-optimal action to the extent that the marginal error is comparable to its sub-optimality gap. In this sense we conclude that \(a\) is well-explored.

**Proof.** We first note that \(\tilde{G}_n\) is invertible for sufficiently large \(n\), otherwise some dimension has not been explored at all so that one can modify \(\theta\) to have linear regret. Let \(\Delta = \min_{a \in A \setminus \{a^*\}} \Delta_a\) and consider a new bandit defined by

\[
\theta' = \theta + \frac{\Delta_a + \epsilon}{\|a - a^\ast\|_H^2} H(a - a^\ast),
\]

where \(H\) is some positive semidefinite matrix with \(\|a - a^\ast\|_H > 0\), and \(\epsilon \in (0, \Delta]\). It is easy to check that action \(a\) under \(\theta'\) is better than \(a^\ast\) by \(\epsilon\). Therefore, applying the Bregman-Huber inequality on the event that \(T_a(n) = \sum_{t \in [n]} I_{a_t = a} \geq n/2\), with the measures being the induced measures under \(\theta\) and \(\theta'\), respectively,

\[
c_p n^p \text{ consistency} \geq R_{n, \theta} + R_{n, \theta'} \geq \frac{n}{2} \mathbb{P}_\theta \left( T_a(n) > \frac{n}{2} \right) + \mathbb{P}_{\theta'} \left( T_a(n) \leq \frac{n}{2} \right)
\]

\[
\geq \frac{n}{4} \epsilon e^{-\frac{1}{2} \frac{\epsilon^2}{\|a - a^\ast\|_H^2}}
\]

for any \(p > 0\) and \(c_p > 0\) depending only on \(p\). Rearranging terms, setting \(p \to 0\) first then \(\epsilon \to 0\) yields that

\[
\liminf_{n \to \infty} \frac{\Delta_a^2 \|a - a^\ast\|_H^2}{2 \log n \|a - a^\ast\|_H^2} \geq 1.
\]

For simplicity assume that \(\log n \|a - a^\ast\|_H^{-2} \to c \in [0, \infty]\) (so that \(\lim = \limsup\)), otherwise consider any convergent subsequence. Take \(H\) to be a limiting point of \(G_n^{-1}/\|G_n^{-1}\|\). Thanks to the consistency and the unique optimality assumptions, \(H\) is only degenerate in direction \(a^\ast\), i.e., \(H a^\ast = 0\) and \(\|a - a^\ast\|_H = \|a\|_H^2 > 0\). One can check this by applying the Sherman-Morrison formula to the matrix \(K + n a a^T\), where \(K\) satisfies \(\|K\| \leq c_p n^p\) for \(p > 0\). This step ensures that the construction of \(\theta'\) is valid. On the other hand, as \(n \to \infty\),

\[
\frac{\|a - a^\ast\|_H^2}{\|a - a^\ast\|_H^2/\|G_n^{-1}\|} = \frac{\|a - a^\ast\|_H^2/\|G_n^{-1}\|}{\|a - a^\ast\|_H^2} \to 1
\]

\[
\frac{\|G_n^{-1}\| \cdot \|a - a^\ast\|_H^2}{\|a - a^\ast\|_H^2} = \frac{\|a - a^\ast\|_H^2}{\|G_n^{-1}\|} \to 1
\]

\[
\frac{\|a\|_H^2}{\|a - a^\ast\|_H^2} \to 1
\]

\[
\frac{\|a - a^\ast\|_H^2}{\|a - a^\ast\|_H^2} \to 1
\]
Plugging these back into (2.5.2) yields the desired result.

We now see how (2.5.1) implies a lower bound on the asymptotic regret \( R_n,\theta \). First note that

\[
R_{n,\theta} = \sum_{a \in \mathcal{A} \setminus a^*} \Delta_a \mathbb{E}[T_a(n)].
\]

On the other hand, \( \bar{G}_n = \sum_{a \in \mathcal{A}} \mathbb{E}[T_a(n)] a a^T \) needs to satisfy (2.5.1) for every \( a \in \mathcal{A} \setminus a^* \). Equivalently, for \( \epsilon > 0 \) and sufficiently large \( n \), \( \log n \|a\|^2_{\bar{G}_n^{-1}} \leq \Delta_a^2 (1 + \epsilon)/2 \). If we let \( a_d(n) = (\log n)^{-1} \mathbb{E}[T_a(n)] \) and \( H_n = (\log n)^{-1} \bar{G}_n \). Then, for sufficiently large \( n \),

\[
\frac{R_{n,\theta}}{\log n} \geq \inf_{a \in \mathcal{A} \setminus a^*} \sum_{a \in \mathcal{A} \setminus a^*} \Delta_a a_d(n),
\]

with \( a \) satisfying

\[
\|a\|_{H_n^{-1}}^2 \leq \frac{1 + \epsilon}{2} \Delta_a^2 \quad \forall a \in \mathcal{A} \setminus a^*
\]

\[
\sum_{a \in \mathcal{A}} a_d(n) = \frac{n}{\log n}.
\]

With further relaxation and setting \( \epsilon \to 0 \) gives

\[
\limsup_{n \to \infty} \frac{R_{n,\theta}}{\log n} \geq \inf_{a \in [0, \infty)^{\mathcal{A}}} \max_{\sum_{a \in \mathcal{A} \setminus a^*} 2 \Delta_a^2 \|a\|_{H_n^{-1}}^2 \leq 1} \sum_{a \in \mathcal{A} \setminus a^*} \Delta_a a_d.
\]

It has been shown in [LS16] that the above equality is achievable. The first step in such an algorithm is to select a small subset \( B \) of \( \mathcal{A} \) so that information of all actions is contained inside. This can be done by taking \( B \) as a barycentric spanner of \( \mathcal{A} \). Secondly, the algorithm explores \( B \) to obtain a rough estimate on the sub-optimality gaps of all actions. The estimated gaps are then used to approximately solve for the pull counts for each sub-optimal actions. If an anomaly is detected that indicates the estimated gaps are inaccurate, the algorithm then switches to a recovery phase where it plays UCB instead.

As a consequence of Theorem 2.5.1, consistent optimistic algorithms are not asymptotically optimal. An intuitive explanation is that optimistic algorithms no longer explore actions which are identified sub-optimal, which usually takes fewer steps than get a good estimation for them. A rigorous justification is as follows:

**Example:** Consider a stochastic linear bandit with \( \theta = e_1 \in \mathbb{R}^2 \) and \( \mathcal{A} = \{e_1, e_2, e_3 := (1 - \epsilon)e_1 + \gamma e_2\} \), where \( e_1, e_2 \) are canonical bases of \( \mathbb{R}^2 \), \( \epsilon > 0 \) and \( \gamma > 1 \). A consistent UCB algorithm will no longer play \( e_2 \) once identifying it is sub-optimal (this process should not take long since \( e_1 \perp e_2 \)), but later on it will surely hesitate for a long time between \( e_1 \) and \( e_3 \) (since \( e_1 \) and \( e_3 \) are almost parallel). An asymptotically optimal policy, however, will play \( e_2 \) longer than seem necessary. This sacrifice will help better decipher the optimal action in \( e_1 \) and \( e_3 \). In fact, it can be shown in this case that
an asymptotically optimal policy has regret $R_{n,\theta} = \mathcal{O}(\log n)$. However, for the UCB, assuming that the confidence set $C_t$ is given by (2.3.7) with $\delta = n^{-1}$,

$$\theta \in C_t = \{ z \in \mathbb{R}^2, \| z - \hat{\theta}_t \|_{\mathcal{V}_t^{-1}(0)}^2 \leq c \log n \},$$

where $c$ is some constant. Then,

$$\max_{z \in C_t} \langle e_2, z \rangle = \max_{z \in C_t} \langle e_2, z - \theta \rangle \leq 2\| e_2 \|_{\mathcal{V}_t^{-1}} \sqrt{c \log n} < 1 \leq \max_{z \in C_t} \langle e_1, z \rangle$$

if $\| e_2 \|_{\mathcal{V}_t^{-1}}^2 \leq (4c \log n)^{-1}$. A sufficient condition for this is $T_{e_2}(t-1) > 4c \log n$. So conditional on $C_t$, $T_{e_2}(t-1) \leq 4c \log n + 1$. Meanwhile, $T_{e_2}(t-1)$ is bounded by $n$ outside $C_t$. Therefore,

$$\mathbb{E}[T_{e_2}(n)] \leq 4c \log n + 2.$$

Equivalently,

$$\alpha_{e_2} \leq 4c.$$

Yet a consistent policy requires $e_2$ be explored to a given extent. If this is not achieved by simply pulling $e_2$ a number of times, it must be compensated by pulling other actions which contain information about $e_2$. In our case, this means playing $e_3$ an extra number of times. However, $e_3$ contains only tangential information of $e_2$ so that it must be chosen extensively. Indeed, the consistency assumption implies that

$$\alpha_{e_3} e_3 + \alpha_{e_2} \geq 2\gamma^2,$$

which further simplifies to $\alpha_{e_3} \geq (2\gamma^2 - 4c)e^{-2}$. Hence, if $\gamma > \sqrt{2c}$, then the regret $R_n$ satisfies $R_n = \mathcal{O}(\epsilon^{-1} \log n)$. Since $\epsilon$ can be arbitrarily small, the regret incurred by the consistent UCB can be any worse than the regret of the best consistent policy.
3.1 Problem Set-up

Learning in stochastic linear bandits can be roughly viewed as a sequential decision-making problem in a random environment. In this chapter, we consider a different type of bandits called adversarial bandits, where the environment is deterministic and prefixed. Learning under such circumstances is therefore more of the game theory flavor. We will first introduce the basic set-up of the problem as well as its variants, then link them to a more general model called the adversarial linear bandit. Meanwhile, we will see that there is a deep connection between adversarial bandit learning and online regularized empirical risk minimization, for which we will provide a detailed discussion later.

Let us begin by some definitions. A $k$-armed adversarial bandit is defined as $k$ deterministic sequences $\{x_{ti}\}_{t \in [n]}$, where $i \in [k]$. We assume that $x_{ti} \in [0, 1]$ representing the loss incurred by playing arm $i$ at $t$. Loss and rewards are transferable by considering $y_{ti} = 1 - x_{ti}$. The key difference, as opposed to the stochastic bandit, is that the environment is non-random, though it can be chosen in an arbitrary manner beforehand. This implies the necessity for learners to adopt a non-deterministic policy (see section 1.1) to gain a non-trivial minimax (universal) regret. To see it, for any policy $\pi$, the regret in the environment $\{x_{ti}\}_{t \in [n]}$ is given by

$$R_n(\pi) = E \left[ \sum_{t \in [n]} x_{ti} - \min_{i \in [k]} \sum_{t \in [n]} x_{ti} \right], \tag{3.1.1}$$

where the expectation is taken with respect to $\pi$. If $\pi$ is a deterministic policy (a measurable function of the history), then we can construct an environment sequentially \(^1\) by defining $x_{1i} = 0$ and for $t \geq 1$, $x_{ti} = \|[t] = i\|_{\pi_t}$. By our construction,

$$\sum_{i \in [k]} \sum_{t \in [k]} x_{ti} = n - 1,$$

\(^1\)Our choice of construction at present will not affect the choices of the policy before.
3.1. PROBLEM SET-UP

implying that

\[ \min_{i \in [k]} \sum_{t \in [n]} x_{ti} \leq \frac{n - 1}{k}. \]

Hence, the minimax regret over the set of deterministic policies \( \Pi_d \) is lower bounded by

\[ \min_{\pi \in \Pi_d} \max_{\{x_t\} \in [0,1]^n} R_n(\pi) \geq \left(1 - \frac{1}{k}\right)(n - 1). \quad (3.1.2) \]

Therefore, in the following analysis we assume that \( \pi \) is non-deterministic. That is, at step \( t \), one first selects a distribution \( P_t \) on \([k]\) using the past observations, then samples \( \pi_t \sim P_t \). As we will see, with proper choice of \( P_t \), the regret grows sub-linearly in \( n \).

A similar problem to the adversarial bandit is bandits with expert advice. In this case, the second term in (3.1.1) is replaced by the loss incurred by following the best expert. To do this explicitly, let \( [M] \) denote the group of experts. Each \( m \in [M] \) is associated with a sequence of distributions on \([k]\), denoted by \( \{E_{tm}\}_{t \in [n]} \). At step \( t \) expert \( m \) samples its \( \pi_t^{(m)} \sim E_{tm} \) as its recommendation for the player. \( E_{tm} \) are deterministic and non-interactive\(^2\) with \( \pi \) (oblivious). The regret is defined by

\[ R_n(\pi) = \max_{m \in [M]} \mathbb{E} \left[ \sum_{t \in [n]} x_{t \pi_t} - \sum_{t \in [n]} x_{t \pi_t^{(m)}} \right] = \mathbb{E} \left[ \sum_{t \in [n]} x_{t \pi_t} - \min_{m \in [M]} \sum_{t \in [n]} \langle E_{tm}, x_t \rangle \right]. \quad (3.1.3) \]

where \( x_t = (x_{t1}, \ldots, x_{tk})^T \). In this context, a strategy \( \pi \) is a distribution \( P_t \) on \([M]\) at step \( t \), which samples \( \pi_t \) according to

\[ \mathbb{P} (\pi_t = i) = \sum_{m \in [M]} P_t(m) E_{im}(i). \]

The details of this algorithm will be analyzed together with the Exp3 algorithm in the next section. The adversarial bandit can be viewed as a special case of an expert-led bandit with \( M = k \) and \( E_{im} = \delta_m \).

We end this section by defining the adversarial linear bandits, which provides a systematic extension of adversarial bandits. Let \( A \subseteq \mathbb{R}^d \) be an action set. For the moment assume that \( A \) is finite and spans \( \mathbb{R}^d \). An adversarial linear bandit is a sequence of deterministic vectors \( \{y_t\}_{t \in [n]} \subseteq A' \), where

\[ A' = \left\{ x \in \mathbb{R}^d : \sup_{a \in A} |\langle x, a \rangle| \leq 1 \right\}. \]

The \( k \)-armed adversarial bandit (with a slightly different loss range) can be recovered by taking \( d = k \), \( A = \{e_i\}_{i \in [d]} \) and \( y_t = x_t \). The expert-led bandit, at first sight, requires

\(^2\)Experts’ advice may contain extra randomness. However, such randomness should be history-independent. Therefore, we could combine the external randomness with the distribution to get a new deterministic distribution. It therefore suffices to assume \( E_{tm} \) as deterministic.
the action set to be time-dependent in order to fit into the framework of adversarial linear bandits. As we will see in later, it is possible to do it by considering actions are distributions on the experts, see Example 3.3.4.

Let \( a_t \) be the action chosen in round \( t \) under a policy \( \pi \). The regret in this case is defined by

\[
R_n(\pi) = \mathbb{E} \left[ \sum_{t \in [n]} \langle a_t, y_t \rangle - \min_{a \in A} \langle a, y_t \rangle \right]. 
\] (3.1.4)

In the rest of this chapter, we will be focused on discussing algorithms with provable sub-linear bounds for (3.1.4). Following the same convention as in the previous discussion, we will abbreviate \( R_n(\pi) \) as \( R_n \) whenever the policy is clear from the context.

### 3.2 Algorithms

We now discuss learning strategies for adversarial linear bandits. The first algorithm we are going to analyze is called Exp3, which is the abbreviation of Exponential-weight algorithm for Exploration and Exploitation. Early ideas of exponential weighting can be found in [LW89; Vov90]. In our case, Exp3 consists of two main steps which are summarized as follows: At step \( t \), we

1. Estimate the environment parameters \( y_t \) before time \( t \); and

2. Build a sampling distribution weighted by the estimated cumulative loss for each \( a \in A \).

This description tells us that past observations play an important role in future decision making. Based on the estimated cumulative loss, one can naively select the action giving the best performance (which becomes a deterministic policy afterwards!). However, such an approach can be unstable due to the adversarial nature of the problem, as suggested in (3.1.2). On the other hand, following the leader simply means no more exploration in the future. What step (2) does is essentially smoothing out the hard-thresholding decision making process. As we will see later, this is equivalent to minimizing the empirical risk function with appropriate regularization.
Let $p$ be a probability distribution on $\mathcal{A}$. We now state the Exp3 algorithm:

**Algorithm 12:** Exp3 Algorithm for Adversarial Linear Bandits

**Input:** $\mathcal{A}$: action set; $\eta$: exploration rate; $p$: mixing distribution; $\gamma$: mixing parameter

**Output:** $(a_t)_{t\in[n]}$

1. Initialization: $t = 1$, $a_1 \sim p$
2. while $t \leq n$ do
3. $\hat{Y}_t = \langle a_t, y_t \rangle \left( \sum_{a \in \mathcal{A}} P_t(a) a a^T \right)^{-1} a_t$ // Importance-weighted estimator
4. $P_{t+1} = (1-\gamma) \hat{P}_{t+1} + \gamma p$, where $\hat{P}_{t+1}(a) \propto e^{-\eta \sum_{s=1}^t \langle a, \hat{Y}_s \rangle}$
5. $a_{t+1} \sim P_{t+1}$
6. $t = t + 1$
7. end while

**Theorem 3.2.1.** Suppose $|\mathcal{A}| = k$. For carefully chosen $\gamma$ and $\eta$, the regret in Algorithm 12 satisfies

$$R_n \leq \sqrt{8(g(p) + d)n \log k},$$

where $g(p) = \max_{a \in \mathcal{A}} \|a\|_2^2 Q^{-1}(p)$. By Theorem 2.2.1, there exists some $p$ such that $g(p) = d$. Hence,

$$R_n \leq 4\sqrt{dn \log k}.$$

**Proof.** The proof is standard for exponential-weighting algorithms. It is worth mentioning that exponential weights also arise in regularized empirical risk minimization. However, the analysis there involves taking Bregman divergence projection, which cannot be computed explicitly in general.

We first note that $\hat{Y}_t$ is an unbiased estimator for $y_t$, therefore $\langle a, y_t \rangle = \mathbb{E}[\langle a, \hat{Y}_t \rangle]$ for fixed $a \in \mathcal{A}$. The trick is to consider the partition functions given by the exponential weights:

$$W_t = \sum_{a \in \mathcal{A}} e^{-\eta \sum_{s=1}^t \langle a, \hat{Y}_s \rangle}.$$

Note that

$$\hat{P}_t = \frac{P_t - \gamma p}{1 - \gamma}. \quad (3.2.1)$$

Note that for fixed $a \in \mathcal{A}$,

$$-\eta \sum_{t \in [n]} \langle a, \hat{Y}_t \rangle \leq \log W_t \leq \sum_{t \in [n]} \log \frac{W_t}{W_{t-1}} + \log k.$$
The term \( \log(W_t/W_{t-1}) \) can be estimated explicitly:

\[
\log \frac{W_t}{W_{t-1}} = \log \left( \sum_{a \in A} \hat{P}_t(a) e^{-\eta \langle a, \hat{Y}_t \rangle} \right) \\
\leq \log \left( \sum_{a \in A} \hat{P}_t(a) (1 - \eta \langle a, \hat{Y}_t \rangle + \eta^2 \langle a, \hat{Y}_t \rangle^2) \right) \\
\leq - \sum_{a \in A} \eta P_t(a) (a, \hat{Y}_t) + \sum_{a \in A} \eta^2 P_t(a) (a, \hat{Y}_t)^2,
\]

where for the \((*)\) we assumed that \( \eta \langle a, \hat{Y}_s \rangle \geq -1 \) and used that \( e^x \leq 1 + x + x^2 \) when \( x \leq 1 \). Summing over \( t \), replacing \( \hat{P}_t \) by (3.2.1) and taking expectation on both sides yields

\[
R_n \leq \log \frac{k}{\eta} + \gamma \mathbb{E} \left[ \sum_{t \in [n]} \sum_{a \in A} (P_t(a) - P_t(a)) (a, \hat{Y}_t) \right] + \frac{1}{1 - \gamma} \mathbb{E} \left[ \sum_{t \in [n]} \sum_{a \in A} \eta P_t(a) (a, \hat{Y}_t)^2 \right]
\]

\[
\leq \frac{\log k}{\eta} + 2\gamma n + \frac{\eta}{1 - \gamma} \mathbb{E} \left[ \sum_{t \in [n]} \sum_{a \in A} P_t(a)^2 (a, \hat{Y}_t)^2 \right]
\]

\[
= \frac{\log k}{\eta} + 2\gamma n + \frac{\eta}{1 - \gamma} \mathbb{E} \left[ \sum_{t \in [n]} \sum_{a \in A} P_t(a) (a, Q^{-1}(P_t) b) b^2 \right]
\]

\[
\leq \frac{\log k}{\eta} + 2\gamma n + \frac{\eta}{1 - \gamma} \mathbb{E} \left[ \sum_{t \in [n]} \sum_{a \in A} P_t(a) (a, Q^{-1}(P_t) a) \right]
\]

\[
\leq \frac{\log k}{\eta} + 2\gamma n + 2\eta nd,
\]

where the last step follows by assuming \( \gamma \leq \frac{1}{2} \). A sufficient condition for \( \eta \langle a, \hat{Y}_s \rangle \geq -1 \) is

\[
\left| \eta \langle a, Q^{-1}(P_t) a_i \rangle \right| \leq 1,
\]

which can be further strengthened as \( \left| \eta \langle a, Q^{-1}(P_t) a_i \rangle \right| \leq 1 \). Since \( Q(P_t) \succeq \gamma Q(p) \), the above condition holds if \( \eta \leq \gamma g(p) \). Taking the equality and \( \gamma = \sqrt{\frac{\log k}{2n \log(p) + 2nd}} \) finishes the proof.

\[\square\]

**Remark 3.2.2 (Extra exploration).** The mixing distribution is used to ensure that \( Q(P_t) \) is away from being singular. This is often necessary in practice, as there exist examples where setting \( \gamma = 0 \) yields very bad regret. For \( k \)-armed adversarial bandits with rewards between 0 and 1, no extra exploration is needed. Nevertheless, extra exploration can often be used to control the variance of an estimator. Together with a modified estimator, it is possible to derive high-probability bounds for the pseudo-regret [Aue+02]
(Exp3.P). Recently, it is proved in [Koc+14] that the extra exploration can be omitted by using a different estimator [Koc+14] (Exp3.IX).

**Remark 3.2.3** (Infinite action set). The above proof assumes that $\mathcal{A}$ is finite. When $\mathcal{A}$ is infinite, one can use almost the same proof by replacing $\mathcal{A}$ by its $\varepsilon$-net discretization. An alternative approach is to directly apply Algorithm 12 to the $\mathcal{A}$, with $P_t'$ and $p$ interpreted as density functions. When $P$ is log-concave, then there exists a polynomial-time algorithm to sample $P_t'$. The regret in this case is $\Theta(d\sqrt{n\log n})$, which is $\sqrt{d}$ worse than the regret in Theorem 3.2.1. More details can be found in chapter 27 in [LS18] as well as chapter 3 in [Bub11].

We next provide a similar algorithm for bandits with expert advice, which is called Exponential weighting for Exploration and Exploitation with Experts (Exp4). We shall mention that in the subsequent section, both Exp3 (for $k$-armed adversarial bandits) and Exp4 can be treated as special cases in the online regularized empirical risk minimization framework. Nevertheless, it is worthwhile giving the concrete algorithm here for its own interest. Since we are comparing our strategy to the best expert in hindsight, we will put a distribution on the experts instead of the actions. This combined with experts’ own distributions on $[k]$ leads to a averaged distribution on $[k]$, which will be used exploration and estimation. Once the estimates of the gain/loss at each arm is obtained, we can calculate the estimate of the average gain/loss incurred by following each expert’s advice. This will in turn feed back into our exponential weights for the distribution on the experts next round. Details of the algorithm are given below:

**Algorithm 13: Exp4 Algorithm for Bandits with Expert Advice**

**Input:** $n$: horizon; $\eta$: exploration rate; $M$: number of experts; $\gamma$: variance-controlling parameter (which is used in Exp3.IX); $E_t = (E_T^T/m)_{m \in [M]}$: experts’ advice

**Output:** $(\pi_t)_{t \in [n]}$

1: **Initialization** $P_1 = (\frac{1}{M}, \ldots, \frac{1}{M})$  // Initial distribution on the experts
2: **for** $t = 1, \ldots, n$ **do**
3: $\pi_t \sim P_t'$ := $P_t E_t$  // Sampling distribution
4: $\hat{Y}_t = \frac{Y_t}{P_t' + \gamma}$  // Importance-weighted estimator
5: $\hat{Y}_t = E_t \hat{Y}_t$.  // Estimated average loss of experts
6: $P_{t+1} \propto e^{-\eta \sum_{i=1}^{k} \hat{Y}_i}$  // Exponential weighting
7: **end for**

Using a similar proof as Theorem 3.2.1, one can show that with $\gamma = 0$ and $\eta = \sqrt{2\log M/nk}$, the regret in Algorithm 13 is bounded by

$$R_n \leq \sqrt{2nk \log M}.$$ 

The bound can be improved to take into account the so-called the degree of disagreement of experts by adopting adaptive learning rates. We will postpone its analysis to Example 3.3.4 in the next section.
3.3 Online Linear Optimization

In this section, we consider the online linear optimization, which provides an alternative perspective towards the adversarial linear bandits considered before. From now on we assume that $\mathcal{A} \subset \mathbb{R}^d$ is convex. Let $a_t$ be the action at time $t$. For fixed $\{y_t\}_{t \in [n]} \subset \mathcal{A}'$ and $a \in \mathcal{A}$, the relative regret to $a$ is defined by

$$R_n(a) = \sum_{t \in [n]} \langle a - a, y_t \rangle.$$  

The goal of online linear optimization can be roughly stated as finding a strategy so that $\max_{a \in \mathcal{A}} R_n(a)$ is asymptotically small. This problem is the same as the adversarial linear bandit provided that $y_t$ is explicitly given. Nevertheless, such difference can be well addressed by importance-weighted estimation, as observed in Algorithm 12. In the rest of the section we will focus on the study of online linear optimization first, then pass the results to adversarial linear bandits.

A simple strategy for online linear optimization is to follow the leader in the past. That is, the action at time $t$ is chosen as the action giving the best cumulative loss up to $t-1$:

$$a_t = \arg\min_{a \in \mathcal{A}} \sum_{s \in [t-1]} \langle a, y_s \rangle.$$  

This is also referred to as the empirical risk minimization. To see the rationale, note that

$$\sum_{t \in [n]} \langle a_{n+1}, y_t \rangle = \sum_{t \in [n-1]} \langle a_{n+1}, y_t \rangle + \langle a_{n+1}, y_n \rangle \geq \sum_{t \in [n-1]} \langle a_n, y_t \rangle + \langle a_{n+1}, y_n \rangle \geq \sum_{t \in [n]} \langle a_{t+1}, y_t \rangle,$$

which implies that

$$\max_{a \in \mathcal{A}} R_n(a) \leq \sum_{t \in [n]} \langle a_{t+1} - a_t, y_t \rangle.$$  

This bound would be vacuous unless $a_{t+1}$ does not change dramatically from $a_t$. Indeed, consider the case when $d = 1$, $\mathcal{A} = [-1, 1]$ and $y_1 = \frac{1}{2}$ and $y_t = (-1)^{t+1}$ for $t > 1$. It is easy to check that the follow the leader policy leads to $a_t = y_t$ for $t > 1$, while the best policy in this case is $a_t = -y_t$ for $t > 1$. The issue here is the same as the one we came across in (3.1.2). To mitigate the potential alternating phenomenon, we consider

---

3 In adversarial linear bandits, only the marginal $\langle a_t, y_t \rangle$ is observed in each round.

4 This usually depends on the geometry of the action set for which we assume nothing other than convexity.
3.3. ONLINE LINEAR OPTIMIZATION

the regularized version of the follow the leader algorithm (FTRL). Let \( F_1, \ldots, F_{n+1} \) be a sequence of convex functions. Define the action at time \( t \) as

\[
a_t = \arg \min_{a \in A} \sum_{s \in [t-1]} \langle a, y_s \rangle + F_t(a) + D_{F_t}(a_{t+1}, a)
\]  

(3.3.1)

The following theorem provides an upper bound for the relative regret in the FTRL:

**Theorem 3.3.1.** Suppose that \( a_t \) in (3.3.1) is well-defined for all \( t \in [n] \). Then, for fixed \( a \in A \),

\[
R_n(a) \leq \sum_{t \in [n]} \left( \langle a_t - a_{t+1}, y_t \rangle - D_{F_t}(a_{t+1}, a_t) \right) + F_{n+1}(a) - F_t(a) + \sum_{t=1}^n \left( F_t(a_{t+1}) - F_{t+1}(a_{t+1}) \right).
\]

where \( D_{F_t} \) is the Bregman divergence defined by \( D_{F_t}(x, y) = F(x) - F(y) - \nabla F(y)(x-y) \).

Particularly, if \( F_t = F / \eta_t \) and \( \eta_t \) is decreasing with \( \eta_t = \eta_{n+1} \), then

\[
R_n(a) = \frac{F(a) - \min_{b \in A} F(b)}{\eta_n} + \sum_{t \in [n]} \left( \langle a_t - a_{t+1}, y_t \rangle - \frac{D_{F_t}(a_{t+1}, a_t)}{\eta_t} \right).
\]

Note that the divergence terms appearing in the above bounds will penalize significant difference between \( a_t \) and \( a_{t+1} \).

**Proof.** We begin by writing

\[
R_n(a) = \sum_{t \in [n]} \langle a_{t+1} - a, y_t \rangle + \sum_{t \in [n]} \langle a_t - a_{t+1}, y_t \rangle.
\]

The first part on the right-hand side can be bounded similarly as in the analysis of the follow-the-leader algorithm. The only difference is that the regularization term allows more accurate estimate using the divergence:

\[
\sum_{t \in [n]} \langle a, y_t \rangle + F_{n+1}(a) \geq \sum_{t \in [n]} \langle a_{n+1}, y_t \rangle + F_{n+1}(a_{n+1})
\]

\[
= \sum_{t \in [n-1]} \langle a_{n+1}, y_t \rangle + F_n(a_{n+1}) + \langle a_{n+1}, y_n \rangle + F_{n+1}(a_{n+1}) - F_n(a_{n+1})
\]

\[
\geq \sum_{t \in [n-1]} \langle a_{n+1}, y_t \rangle + F_n(a_{n+1}) + D_{F_n}(a_{n+1}, a_n) + \langle a_{n+1}, y_n \rangle + F_{n+1}(a_{n+1}) - F_n(a_{n+1})
\]

\[
\vdots
\]

\[
\geq \sum_{t \in [n]} \langle a_{t+1}, y_t \rangle + F_t(a_t) + \sum_{t \in [n]} D_{F_t}(a_{t+1}, a_t) + \sum_{t \in [n]} (F_{t+1}(a_{t+1}) - F_t(a_{t+1})),
\]

where for the second inequality follows from the first-order optimality of \( a_n \):

\[
\Phi_n(a_{n+1}) - \Phi_n(a_n) \geq D_{\phi_n}(a_{n+1}, a_n) = D_{F_n}(a_{n+1}, a_n).
\]

Reorganizing terms yields the desired result. \( \square \)
One could also consider a local version of (3.3.1). Here we assume that $F_t = F / \eta_t$ for some decreasing sequence $\infty = \eta_1 > \eta_2 \geq \cdots \geq \eta_{n+1}$.

$$a_t = \arg\min_{a \in A} \langle a, y_{t-1} \rangle + D_{F_t}(a)$$

$$= \arg\min_{a \in A} \eta_t \langle a, y_{t-1} \rangle + D_{F_t}(a) \ .$$

(3.3.2)

(3.3.2) can be interpreted as a two-phase procedure which is often called the mirror descent (primal-dual): first map $a_{t-1}$ to the dual space via the mirror map $F$: $a_{t-1} \mapsto \nabla F(a_{t-1})$, take the gradient descent with step length $t$, and pull back to the point $\tilde{a}_t$ in the domain of $F$. Then obtain $a_t$ from $\tilde{a}_t$ by taking the Bregman divergence projection to the action set $A$. The two steps can be summarized as follows under appropriate domain conditions,

$$\tilde{a}_t = \arg\min_{a \in \text{Dom}(F)} \langle a, y_{t-1} \rangle + D_{F_t}(a, \tilde{a}_{t+1})$$

$$a_t = \text{Proj}_{D_{F_t}}(\tilde{a}_t).$$

A similar bound for the regret holds true as in the case of FTRL:

**Theorem 3.3.2.** Suppose that $a_t$ in (3.3.2) is well-defined for all $t \in [n]$. Then, for fixed $a \in A$,

$$R_n(a) = \sum_{t \in [n]} \frac{D_{F}(a_t, \tilde{a}_{t+1})}{\eta_{t+1}} + \sum_{t \in [n]} \frac{D_{F}(a_t, a_{t+1}) - D(a_t, \tilde{a}_{t+1})}{\eta_{t+1}}$$

$$\leq \sum_{t \in [n]} \frac{D_{F}(a_t, \tilde{a}_{t+1})}{\eta_{t+1}} + \sum_{t \in [n]} D_{F}(a_t, a_{t+1}) \left( \frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right).$$

**Proof.** The first-order optimality of $\tilde{a}_{t+1}$ implies that

$$y_t = \frac{1}{\eta_{t+1}} (\nabla F(a_t) - \nabla F(\tilde{a}_{t+1})).$$

Plugging into the definition of $R_n(a)$,

$$R_n(a) = \sum_{t \in [n]} \langle a_t - a, y_t \rangle = \sum_{t \in [n]} \frac{1}{\eta_{t+1}} \langle a_t - a, \nabla F(a_t) - \nabla F(\tilde{a}_{t+1}) \rangle$$

$$= \sum_{t \in [n]} \frac{1}{\eta_{t+1}} (D_{F}(a_t, \tilde{a}_{t+1}) + D_{F}(a_t, a_{t+1}) - D_{F}(a_t, \tilde{a}_{t+1}))$$

$$\leq \sum_{t \in [n]} \frac{D_{F}(a_t, \tilde{a}_{t+1})}{\eta_{t+1}} + \sum_{t \in [n]} \frac{D_{F}(a_t, a_{t+1}) - D_{F}(a_t, a_{t+1})}{\eta_{t+1}},$$

where for the last step we used that $D_{F}(a_t, a_{t+1}) \leq D_{F}(a_t, \tilde{a}_{t+1})$. Rearranging terms leads to the desired result. \qed
3.3. ONLINE LINEAR OPTIMIZATION

Results in Theorem 3.3.1 and 3.3.2 can be easily modified to apply to the randomized versions of the algorithms. Indeed, as in adversarial linear bandits, suppose that one can estimate \( y_i \) by some unbiased estimator \( \hat{Y}_t \). The decision at \( t \) should be \( a_t = \arg \min_{a \in A} \sum_{i=1}^{t-1} \langle a, \hat{Y}_i \rangle + F_i(a) := a^*_t \) or \( a_t = \arg \min_{a \in A} \langle a, \hat{Y}_{t-1} \rangle + D_F(a) := a^\dagger_t \). However, conditional on history, both algorithms will lead to fixed choices in the future so that bias cannot be resolved. A possible way to get around this issue is to sample \( a_t \) from some distribution \( P_t \) on \( A \) with mean \( a^*_t \) or \( a^\dagger_t \). This leads to the following algorithm:

**Algorithm 14: FTRL/Mirror Descent Algorithms for Adversarial Linear Bandits**

**Input:** \( A \): action set; \( F(a) \): regularization; \( \{\eta_t\}_{t \in [n]} \): learning rates

**Output:** \( (a_t)_{t \in [n]} \)

1. **Initialization:** 
   \( t = 1, a_1 = \arg \min_{a \in A} F(a) \)
2. **while** \( t \leq n \) **do**
   3. Estimate \( y_t \) by an unbiased estimator \( \hat{Y}_t \)
   4. \( \bar{a}_{t+1} = \arg \min_{a \in A} \eta_{t+1} \sum_{i \in [t]} \langle a, \hat{Y}_i \rangle + F(a) \) (FTRL)
   5. \( \bar{a}_{t+1} = \arg \min_{a \in A} \eta_{t+1} \langle a, \hat{Y}_{t-1} \rangle + D_F(a) \) (Mirror descent)
   6. Sample \( a_{t+1} \) from \( P_{t+1} \), where \( P_{t+1} \) is some distribution on \( A \) with mean \( \bar{a}_{t+1} \)
   7. **end while**

The regret bound given by Algorithm 14 follows directly from Theorem 3.3.1 and 3.3.2. Take the FTRL for instance. For each \( t \), conditional on \( \bar{a}_t \) and averaging over \( P_t \),

\[
\sum_{i \in [n]} \mathbb{E}[\langle \bar{a}_t - a, \hat{Y}_i \rangle] = \sum_{i \in [n]} \mathbb{E}[\langle \bar{a}_t - a, y_i \rangle] = \sum_{i \in [n]} \mathbb{E}[\langle a_t - a, y_i \rangle] = R_n. \tag{3.3.3}
\]

Therefore,

\[
R_n \leq \max_{a,b \in A} \frac{F(a) - F(b)}{\eta_n} + \mathbb{E} \left\{ \sum_{i \in [n]} \left[ \frac{\langle \bar{a}_t - \bar{a}_{t+1}, \hat{Y}_i \rangle}{\eta_t} - \frac{D_F(\bar{a}_{t+1}, \bar{a}_t)}{\eta_{t+1}} \right] \right\}. \tag{3.3.4}
\]

where \( H_t \) is the Hessian of \( F \) at some point \( \xi_t \) between \( \bar{a}_t \) and \( \bar{a}_{t+1} \).

We now see a few examples:

**Example 3.3.3** (\( k \)-armed adversarial bandits). Let \( d = k \) and \( A = \text{conv}(e_i)_{i \in [k]} \).

Set \( F(a) \) as the negentropy \( \sum_{i \in [k]} a^{(i)}(\log a^{(i)} - 1) \), where \( a = (a^{(i)})_{i \in [k]} \). It is easy to check

\[\text{conv}(e_i)_{i \in [k]}\]
that \( a_i = (1/k, \cdots, 1/k)^T \). For \( x \in \mathcal{A} \) and \( y \in \mathbb{R}_+^k \),

\[
D_F(x, y) = \sum_{i \in [k]} x^{(i)}(\log x^{(i)} - 1) - \sum_{i \in [k]} y^{(i)}(\log y^{(i)} - 1) - \sum_{i \in [k]} \log y^{(i)}(x^{(i)} - y^{(i)})
\]

\[
= \sum_{i \in [k]} x^{(i)} \log \frac{x^{(i)}}{y^{(i)}} + \text{terms of } y.
\]

Therefore, \( \text{Proj}_{F, \mathcal{A}}(y) = y/\|y\|_1 \). From direct computation, we find that

\[
a_i^{(t)} = e^{t \sum_{s \in [t-1]} y^{(i)} - \sum_{s \in [t-1]} y^{(i)}}. \tag{3.3.5}
\]

Replacing \( y_i \) by the importance-weighted estimator and and choosing \( P_t \) as a distribution on \( \{e_i\}_{i \in [k]} \) with \( P_t(e_i) = a_i^{(t)} \) in (3.3.4) recovers the Exp3 algorithm for \( k \)-armed bandits. Using the fact that \( F \in [1, \log k + 1] \), \( \nabla^{-2} F(a) = \text{diag}(a) \), \( \hat{Y}_t = \langle y_t, e^{A_t} \rangle P_t(A_t) e^{A_t} \) and \( P_t(A_t) \geq P_{t+1}(A_t) \),

\[
R_n \leq \log M/n + \frac{k}{2} \sum_{t \in [n]} \eta_t.
\]

Taking \( \eta_t = \sqrt{\log k/kt} \) (adaptive learning rates) yields \( R_n \leq 2\sqrt{nk \log k} \).

**Example 3.3.4** (Disagreement of experts). We demonstrate that bandits with expert advice can be viewed as a special case of adversarial linear bandits. We will use the same notation as in Algorithm 13. To see this, take \( \mathcal{A} = \text{conv}(e_m)_{m \in [M]} \). Similar to Example 3.3.3, we use the convex hull for convenience. For each \( a \in \mathcal{A} \), its expected loss is given by \( \langle E_t^a, y_t \rangle = \langle a, E_t y_t \rangle \). Therefore, the environment is given by the transformed sequence \( \{E_t y_t\}_{t \in [n]} \). Under this set-up, a natural unbiased estimator for \( E_t y_t \) is

\[
E_t y_t = \mathbb{E}[E_t \hat{Y}_t],
\]

where \( \hat{Y}_t \) is the importance-weighted estimator. Note that directly estimating \( E_t y_t \) is hard since they are closely correlated, and it is often easier to estimate the low-dimensional structure. Taking \( F(a) \) and \( P_t \) the same as in Example 3.3.3 recovers Algorithm 13. The regret \( R_n \) is bounded by (3.3.4). Moreover, in this case, the first term on the right-hand side of (3.3.4) is bounded by \( \log M/\eta_t \), and for the second term, note that it is enough to consider when

\[
\langle \hat{a}_t - \hat{a}_{t+1}, E_t \hat{Y}_t \rangle \geq 0,
\]

or equivalently,

\[
\langle \hat{a}_t, E_t^{(\sigma_j)} \rangle \geq \langle \hat{a}_{t+1}, E_t^{(\sigma_j)} \rangle.
\]
3.3. ONLINE LINEAR OPTIMIZATION

\[\langle \tilde{a}_t - \tilde{a}_{t+1}, E_t \tilde{Y}_t \rangle - \frac{D_F(\tilde{a}_{t+1}, \tilde{a}_t)}{\eta_t} = \frac{\eta_t}{2} \| E_t \tilde{Y}_t \|_{H_t}^2 = \frac{\eta_t y_{t\sigma_t}}{2 p_t^2(\pi_t)} \langle H_t^{-1} E_t^{(\pi_t)}, E_t^{(\pi_t)} \rangle \]

\[= \frac{\eta_t}{2 p_t E_t} \langle H_t^{-1} E_t^{(\pi_t)}, E_t^{(\pi_t)} \rangle \leq \frac{\eta_t}{2 p_t E_t} \sum_{m \in [M]} \left( \kappa \overline{a}_t^{(m)} + (1 - \kappa) \overline{a}_{t+1}^{(m)} \right) E_t^{(\pi_t)}(m) \]

\[\leq \frac{\eta_t \max_{m \in [M]} E_t^{(\pi_t)}(m)}{2 p_t E_t^{(\pi_t)}} \sum_{m \in [M]} \left( \kappa \overline{a}_t^{(m)} + (1 - \kappa) \overline{a}_{t+1}^{(m)} \right) E_t^{(\pi_t)}(m) \]

where \( \kappa \in [0, 1] \) is some case-dependent constant. Averaging over \( \pi_t \) and summing over \( t \) yields

\[\sum_{t \in [n]} \left[ \langle \tilde{a}_t - \tilde{a}_{t+1}, E_t \tilde{Y}_t \rangle - \frac{D_F(\tilde{a}_{t+1}, \tilde{a}_t)}{\eta_t} \right] \leq \frac{1}{2} \sum_{t \in [n]} \eta_t \sum_{i \in [k]} \max_{m \in [M]} E_t^{(i)}(m) \]

Taking \( \eta_t = \sqrt{\log M / E_t} \) and combining the two estimates together arrives the bound for \( R_n \):

\[R_n \leq C \sqrt{E_n^* \log M}, \quad (3.3.6)\]

where \( C > 0 \) is some absolute constant. The quantity \( E_n^* \) measures the cumulative degree of disagreement. It is easy to check that

\[n \leq E_n^* \leq n \min \{ k, M \}.\]

The lower bound is achieved when all the experts give exactly the same advice in each round, and the upper bound is achieved when all experts give completely different and affirmative advice. Bound (3.3.6) is encouraging since it implies that learning is easier when the experts have more aligned recommendations, which coincides with the intuition.

**Example 3.3.5** (The log barrier and first order bounds). This example is taken from an exercise in [LS18], which is adapted from the work [WL18]. It has exactly the
same context as in the Example 3.3.3 except with a different regularization function: 
\[ F(a) = -\sum_{i\in[k]} \log a(i). \] This gives a different weight matrix \( \nabla^2 F(x) = \text{diag}(x^2) \). In this case, it is easy to check that
\[
R_n(a) \leq \frac{F(a) - k \log k}{\eta_n} + \frac{1}{2} \sum_{t\in[n]} [E[y_t, e_A_t]^2].
\]

When maximizing over \( a \), the bound becomes useless since \( F(a) \) blows up when \( a \) approaches the corners of \( \mathcal{A} \). We consider a truncated version of \( \mathcal{A} \) and use the boundedness assumption to achieve a more reasonable bound for the regret. For convenience let arm 1 be the optimal action. Then,
\[
R_n \leq \min_{a\in \mathcal{A}\cap\{\min_{i\in[k]} a(i) \geq \frac{1}{n}\}} \sum_{t\in[n]} \langle a - e_1, y_t \rangle + \max_{a\in \mathcal{A}\cap\{\min_{i\in[k]} a(i) \geq \frac{1}{n}\}} R_n(a)
\]
\[
\leq k - 1 + \frac{k \log \left( \frac{e}{k} \right)}{\eta_n} + \frac{1}{2} \sum_{t\in[n]} [E[y_t, e_A_t]^2].
\]

Taking
\[
\eta_t = \sqrt{\frac{k \log \left( \frac{n^k}{k} \right)}{1 + \sum_{s\in[t-1]} (y_s, e_A_s)^2}}
\]
yields
\[
R_n \leq k - 1 + 2 \sqrt{k \left( 1 + \sum_{t\in[n]} (a_t, e_A_t)^2 \right) \log \left( \frac{n^k}{k} \right)}
\]
\[
\leq k - 1 + 2 \sqrt{k \left( 1 + R_n + \min_{i\in[k]} \sum_{t\in[n]} (a_t, e_t) \right) \log \left( \frac{n^k}{k} \right)},
\]
which further simplifies to
\[
R_n = \Theta \left( \sqrt{k \min_{i\in[k]} \sum_{t\in[n]} (a_t, e_t) \log \left( \frac{n^k}{k} \right)} \right).
\]

### 3.4 Universal Portfolio

We end this chapter with an interesting application of exponential-weighting algorithm called Universal Portfolio in finance. In fact, this strategy predates the other algorithms we have studied for adversarial linear bandits so far, and is concerned with maximizing sequential investment in long term in a stock market. For discussion we will mainly follow the paper [Cov11] where the result first appeared.

We first introduce the problem. Suppose that in a stock market, we are interested in \( k \) stocks and want to invest in them using rebalanced portfolio. A portfolio can be
thought as the convex combination coefficients that an investor uses to allocate money into different assets, and rebalanced means that the portfolio is changing with respect to the investment periods. Although this may not sound practical for some investors, it is interesting to know how good one can achieve under such circumstances. As we will see, Universal Portfolio provides a reasonable strategy whose cumulative return matches the best constant portfolio in hindsight in an asymptotic sense. Before giving more details, we will need some notation. Let \( n \) be the horizon as before. For \( t \in [n] \), define \( x_t = (x_{t1}, \cdots, x_{tk})^T \) be the stock market vector at the \( t \)-th investment period, i.e., \( x_{ti} \) is the ratio of the closing to open price for stock \( i \) at time \( t \). A rebalanced portfolio strategy is a sequence of vectors \((b_t)_{t \in [n]}\) such that \( b_t \in \mathcal{P}([k])\) and is non-anticipating\(^8\), where \( \mathcal{P}([k]) \) denotes the set of distribution on \([k]\). The total amount of gain relative to the original investment is after time \( n \) is given by

\[
S_n = \prod_{t \in [n]} b_t^T x_t = e^{\Sigma b_t^T x_t},
\]

where dependence on \( b_t \) in the notation is omitted for convenience. The optimal criterion in this case is the gain obtained by sticking to the best constant portfolio in hindsight:

\[
S_n^* = \max_{b \in \mathcal{P}([k])} S_n(b) \quad S_n(b) = \prod_{t \in [n]} b_t^T x_t = e^{\Sigma b_t^T x_t}.
\quad (3.4.1)
\]

The solution (assumed unique) of \( (3.4.1) \) is denoted by \( b_n^* \). Intuitively, \( S_n(b) \) grows exponentially with \( n \). Therefore, using Laplace’s method, one should expect that

\[
\frac{1}{n} \log S_n^* \approx \frac{1}{n} \log \left( \frac{\int_{b \in \mathcal{P}([k])} S_n(b) \, db}{\int_{b \in \mathcal{P}([k])} \, db} \right) \quad \text{as } n \to \infty. \quad (3.4.2)
\]

On the other hand, note that \( S_t(b) \) depends on the stock price in the first \( t \) rounds. To make the right-hand side of \( (3.4.2) \) into a non-anticipating strategy, write

\[
\frac{\int_{b \in \mathcal{P}([k])} S_t(b) \, db}{\int_{b \in \mathcal{P}([k])} \, db} = \prod_{t \in [n]} \frac{\int_{b \in \mathcal{P}([k])} S_t(b) \, db}{\int_{b \in \mathcal{P}([k])} \, db} = \prod_{t \in [n]} \left( \frac{\int_{b \in \mathcal{P}([k])} S_{t-1}(b) \, db}{\int_{b \in \mathcal{P}([k])} \, db} \right)^T x_t, \quad (3.4.3)
\]

where we assume that \( S_0(b) \equiv 1 \). The \( b_t \) defined above is non-anticipating, and would yield the same cumulative gain as the left-most term in \( (3.4.3) \). This strategy is called Universal Portfolio. To get an idea of the presence of exponential weights, take the action set as the probability measure simplex \( \mathcal{P}([k]) \). In each round \( t \), action \( b \in \mathcal{P}([k]) \) is sampled from some distribution whose density \( p_t(b) \) satisfies

\[
p_t(b) \propto S_{t-1}(b) = e^{\Sigma_{s \in [t-1]} \log(b^T x_s)},
\]

where \( \Sigma_{s \in [t-1]} \log(b^T x_s) \) is the cumulative log-return\(^9\) under \( b \) in \( t - 1 \) consecutive

---

\(^8\)Non-anticipating means that \( b_t \) can only depend on information before \( t \).

\(^9\)Log-return is standard in stock market since the returns are often assumed compounded.
CHAPTER 3. ADVERSARIAL LINEAR BANDITS

rounds. The learning rate is 1.

Denote the cumulative gain under Universal Portfolio by \( \hat{S}_n \). Using Jensen’s inequality twice, one could check that Universal Portfolio exceeds value line index, which is defined as the geometric mean of the single portfolios:

\[
\hat{S}_n \geq \left( \prod_{i \in [k]} S_n(e_i) \right)^{1/k},
\]

where \( e_i \)'s are the canonical basis in \( \mathbb{R}^k \).

Define the sensitivity matrix \( J_n(b) \) as

\[
(J_n(b))_{ij} = \int \frac{(\langle x, e_i \rangle - \langle x, e_m \rangle)(\langle x, e_j \rangle - \langle x, e_m \rangle)}{(b^T x)^2} d F_n(x) \quad i, j \in [m - 1]
\]

where \( F_n \) is the empirical distribution of \( x_1, \ldots, x_n \). Let \( J^* = J_n(b^*) \).

The main theorem in [Cov11] is the following:

**Theorem 3.4.1.** Suppose that \( x_1, x_2, \ldots, \in [a, c]^m \) for \( 0 < a \leq c < \infty \). Assume that

\[
W_n(b) := \frac{1}{n} \sum_{i \in [n]} \log(b^T x_i), \quad W(b) \quad \forall b \in P([k])
\]

\[
b^*_n \rightarrow b^*,
\]

\[
J^*_n \rightarrow J^*,
\]

where \( W(b) \) is strictly concave, the third partial derivatives of \( W \) are bounded on \( P([k]) \), and \( W(b) \) achieves its maximum at \( b^* \) in the interior of \( P([k]) \). Then,

\[
\frac{\hat{S}_n}{S^*_n} \sim \left( \sqrt{\frac{2\pi}{n}} \right)^{m-1} \frac{(m-1)!}{|J^*|^{1/2}}. \tag{3.4.4}
\]

Note that if taking logarithm on both sides of (3.4.4), we get the asymptotic regret bound:

\[
R_n = \log S^*_n - \log \hat{S}_n \sim O \left( m \log n \right).
\]

This is better than the finite-time bound on various algorithms for adversarial linear bandits. This is not surprising since the log-return of stocks is observable at the end of each round so no estimation is needed.

We end this section by applying Universal Portfolio to a real dataset. Since our result is of asymptotic flavor, we need to use stocks which have sufficient historical data. Particularly, we will use the stock data\(^\text{10}\) of the four largest banks in the US (in terms of their assets): JPMorgan Chase (JPM), Bank of America (BAC), Citigroup (C) and Wells Fargo & Co. (WFC). The whole time span is set from June 1990 to June 2020, and portfolio rebalancing is conducted on a monthly basis. The stock market vector is the closing price at the end of each month divided by the opening price at the beginning of the same month. We apply the Universal Portfolio\(^\text{11}\) to the dataset and the simulation results are given in Figure 3.1.

\(^{10}\) The dataset can be downloaded from https://finance.yahoo.com/.

\(^{11}\) The integrals used in weighting are calculated using Monte-Carlo approximation with 10000 samples.
The cumulative gain under Universal Portfolio is $\hat{S}_n = 7.52$. The cumulative gains for solely investing in JPM, BAC, C and WFC are 7.87, 2.91, 1.20 and 10.32, respectively. The value line index in this case is 4.11, which is surpassed by $\hat{S}_n$. We also calculate the best constant portfolio in hindsight. The approximate solution is given by $b_n^* = (0.4, 0, 0, 0.6)^T$ (0.4 for JPM and 0.6 for WFC), and the corresponding cumulative gain is around 12.60. This is close to investing only in WFC for 30 years. However, putting all the money in a single basket is extremely risky, so be cautious!
Chapter 4

Bandits with Cost

4.1 Best-Arm Identification

The best arm identification problem has a different goal from the classical bandits. Precisely, the objective considered in the former is a simple regret measured only by the decision at the end of the game. Consider drug testing as an example. When new drugs are tested on humans, the test and cure processes coincide. Yet in other situations where drugs are first experimented on mice, it is often the case that cost is negligible; people are mainly concerned with the potential effects on humans later on. As such, learning becomes a pure exploration process. In this section, we will introduce two different formulations of the problem based on the works [AB10; BMS09] and [GK16], respectively.

4.1.1 Best-Arm Identification with a Budget

Consider a (1-sub-gaussian) stochastic bandit with fixed horizon $n$, which can also be interpreted as the budget constraint for exploration. The goal is to minimize the simple regret at step $n+1$:

$$r_{n+1} = E_{\psi_n} [\Delta],$$

where $\psi_n$ is the decision function (a distribution on the indices) at $t = n + 1$ and $\Delta$ is the sub-optimality gap associated with the arm $\psi_n$. Common choices of $\psi_n$ are the empirical best arm (EBA), the most played arm in the exploration (MPA), or the empirical distribution of plays (EDP).

Pure exploration is closely related to the classical stochastic bandits. A key difference, however, is that the former may keep pulling arms which have been identified sub-optimal\(^1\). The amount of effort, as will be shown, is inversely proportional to the square of the sub-optimality gap. Algorithms to be introduced towards this direction involve the Uniform Allocation Strategy (UAS), UCB-E, and Sequential Halving. Similar algorithms also appear in the study of classical stochastic bandits, see Algorithm

\(^1\)This is similar to what we have seen in Theorem 2.5.1.
4.1. **BEST-ARM IDENTIFICATION**

1, 2, 4. The next two theorems assist us understand what one might expect by directly taking an algorithm from the classical stochastic bandit to the pure exploration context.

**Theorem 4.1.1.** Let $\epsilon(n)$ be a function of $n$. For any stochastic bandit, if the cumulative regret $R_n$ of a policy satisfies $R_n \leq \epsilon(n)$, then the simple regret of the same policy at $n + 1$ with $\psi_n$ chosen as the EDP satisfies

$$r_{n+1} \leq \frac{\epsilon(n)}{n}.$$

**Proof.** It follows from the definition of $r_{n+1}$ and EDP:

$$r_{n+1} = \mathbb{E} \left[ \sum_{i \in [k]} \frac{T_i(n)}{n} \Delta_i \right] \leq \frac{R_n}{n} \leq \frac{\epsilon(n)}{n}.$$

\[\square\]

On the other hand, a good policy for $R_n$ may be inferior for $r_{n+1}$, due to the lack of exploration:

**Theorem 4.1.2.** Let $\mathcal{E} = [0, 1]^k$ be the Bernoulli bandits. Let $\mu = (\mu_1, \cdots, \mu_k) \in \mathcal{E}$ be a bandit such that $\mu_k < \cdots < \mu_1 < 1$. Suppose that the worst-case cumulative regret of a policy satisfies

$$\max_{(\nu_i)_{i \in [k]} \in \mathcal{E}} R_{\nu_i} \leq \epsilon(n). \quad (4.1.1)$$

Then, for $k \geq 3$,

$$\max_{\sigma \in S_k} r_{\sigma(\mu), n+1} \geq \frac{\mu_2 - \mu_1}{2} (1 - \mu_1)^{\frac{2\nu(n)}{n^2}},$$

where $r_{\sigma(\mu), n+1}$ is the simple regret of the same policy in the environment $\sigma(\mu) := (\mu_{\sigma^{-1}(1)}, \cdots, \mu_{\sigma^{-1}(k)})$ and $S_k$ is the set of permutations on $[k]$.

The permutation step is taken to avoid pure luck of a trivial policy. For example, always pulling a fixed arm can be optimal when that arm has the largest mean. Most reasonable policies, however, are invariant under permutation of indices.

**Proof.** Let $\mu' = (0, \mu_2, \cdots, \mu_k)$ and $\mu'' = (0, \mu_2, \cdots, \mu_{k-1}, 0)$. For $i \in [k]$, let $G_i$ be the event that the cumulative gain from arm $i$ is 0. Fix $\sigma \in S_k$. It is easy to check that

$$r_{\sigma(\mu), n+1} \geq (\mu_2 - \mu_1) \mathbb{P}_{\sigma(\mu)} \left( \psi_{\sigma(\mu), n} \neq \sigma(1) \right)$$

$$\geq (\mu_2 - \mu_1) \mathbb{P}_{\sigma(\mu)} \left( \psi_{\sigma(\mu), n} \neq \sigma(1) \big| G_{\sigma(1)} \right) \mathbb{P}_{\sigma(\mu)} \left( G_{\sigma(1)} \right)$$

$$\geq (\mu_2 - \mu_1) \mathbb{P}_{\sigma(\mu')} \left( \psi_{\sigma(\mu), n} \neq \sigma(1) \big| G_{\sigma(1)} \right) \mathbb{P}_{\sigma(\mu)} \left( G_{\sigma(1)} \right),$$

conditioning events
where the last step follows by repeating the first two steps. To estimate the conditioning events, we appeal to the change of measure technique. Since the restriction of $\sigma(\mu)$ on $G_{\sigma(1)}$ is absolutely continuous with respect to $\sigma(\mu')$, and $\sigma(\mu')$ is supported on $G_{\sigma(1)}$,

$$P_{\sigma(\mu)}(G_{\sigma(1)}) = E_{\sigma(\mu')} \left[ \frac{d\sigma(\mu)}{d\sigma(\mu')} \right] = E_{\sigma(\mu')} \left[ (1 - \mu_1)^T_{\sigma(1)(n)} \right] \geq (1 - \mu_1)^{T_{\sigma(1)(n)}} \geq (1 - \mu_1) \frac{e(n)}{n^2},$$

where the last inequality follows from (4.1.1). Similarly,

$$P_{\sigma(\mu')}(G_{\sigma(k)}) \geq (1 - \mu_k) \frac{e(n)}{n^2}.$$

Let $\tilde{\sigma}$ be an element in $S_k$ which agrees with $\sigma$ but reverses the actions on 1 and $k$. Then $\tilde{\sigma}(\mu'') = \sigma(\mu'')$. Hence,

$$\max_{\sigma \in S_k} r_{\sigma(\mu),n+1} \geq \frac{1}{2} (r_{\sigma(\mu),n+1} + r_{\tilde{\sigma}(\mu),n+1}) \geq \frac{\mu_2 - \mu_1}{2} (1 - \mu_k) \frac{e(n)}{n^2} \left( P_{\sigma(\mu'')} \left( \psi_{\sigma(\mu),n} \neq \sigma(1) \right) + P_{\tilde{\sigma}(\mu'')} \left( \psi_{\tilde{\sigma}(\mu),n} \neq \tilde{\sigma}(1) \right) \right) \geq \frac{\mu_2 - \mu_1}{2} (1 - \mu_1) \frac{e(n)}{n^2} \left( 2 - P_{\sigma(\mu'')} \left( \psi_{\sigma(\mu),n} = \sigma(1) \right) - P_{\tilde{\sigma}(\mu'')} \left( \psi_{\tilde{\sigma}(\mu),n} = \tilde{\sigma}(1) \right) \right) \geq \frac{\mu_2 - \mu_1}{2} (1 - \mu_1) \frac{e(n)}{n^2}.$$

\[\square\]

**Remark 4.1.3.** For any Bernoulli bandit with distinct components which are not equal to 1, Theorem 4.1.2, Theorem 1.2.4 and the permutation invariance of the UCB imply that the simple regret incurred by applying Algorithm 2 has a polynomial lower bound $\Omega(n^\gamma)$, where $\gamma$ and the implicit constant depend on the bandit, i.e., the sub-optimality gap, the gap between 1 and the largest component, etc..

We now turn to discuss the algorithms. The first algorithm we have in mind is the Uniform Allocation Strategy, which is similar to Algorithm 1 except that there is no exploitation involved. The final decision $\psi_n$ is chosen as the EBA. Details of the algorithm are given in Algorithm 15.

**Algorithm 15:** The Uniform Allocation Algorithm

**Input:** $n$: horizon

**Output:** $\psi_n$

1: Play each arm $\left\lfloor \frac{n}{k} \right\rfloor$ times and compute their empirical mean: $\hat{\mu}_1, \ldots, \hat{\mu}_k$

2: $\psi_n = \arg \max_{j \in [k]} \hat{\mu}_j$

This is a finite measure but not a probability measure.
Theorem 4.1.4. For $n \geq k$, the simple regret $r_{n+1}$ in Algorithm 15 satisfies
\[
    r_{n+1} \leq \sum_{i \in [k]} \Delta_i e^{-\Delta_i^2 |\frac{t}{n}|} \leq \max_{i \in [k]} \Delta_i \sum_{i \in [k]} e^{-\Delta_i^2 |\frac{t}{n}|},
\]
(4.1.2)

where $\Delta_i$ is the sub-optimality gap of the arm $i$.

Remark 4.1.5. Let $\epsilon > \sqrt{1/2[\frac{\sigma^2}{k}]}$ be some parameter to be determined later. The bound in Theorem 4.1.4 can be made distribution-free:
\[
    r_{n+1} = \sum_{i \in [k]} \Delta_i \mathbb{P}(\psi_n = i) \leq \epsilon + \sum_{i : \Delta_i > \epsilon} \Delta_i \mathbb{P}(\psi_n = i) \leq \inf_{\epsilon} \left( \epsilon + (k-1)e^{-\epsilon^2 |\frac{t}{n}|} \right) \leq 2 \sqrt{\frac{k \log k}{n + k}},
\]

where the third inequality used that $x \to xe^{-cx^2}$ is decreasing when $x \geq \sqrt{\frac{1}{2s}}$, and the last inequality follows by taking $\epsilon = \sqrt{\log k/[\frac{n}{k}]}$.

The algorithm we are introducing next is called the Upper Confidence Bound Exploration (UCB-E), which is similar to the Algorithm 2 but with an additional parameter $a$ tuning the exploration rate. For convenience we let the decision function $\psi_n$ be the EBA.

Algorithm 16: The Upper Confidence Bound Exploration Algorithm

Input: $n$: horizon; $\delta$: confidence level; $a$: exploration parameter

Output: $\psi_n$

1: while $t \leq n$ do
2: \hspace{1cm} $\pi_t = \arg \max_{i \in [k]} \text{UCB}_i^{(a)}(t-1)$, where for $i \in [k],$
3: \hspace{1cm} $\text{UCB}_i^{(a)}(t-1) = \begin{cases} 
\infty & T_i(t-1) = 0 \\
\hat{\mu}_i(t-1) + \sqrt{\frac{a}{T_i(t-1)}} & T_i(t-1) > 0,
\end{cases}$
4: \hspace{1cm} and $\hat{\mu}_i(t-1) = \frac{1}{T_i(t-1)} \sum_{s \in [t-1]} x_{is} I_{\pi_s = i}$
5: end while
6: $\psi_n = \arg \max_{i \in [k]} \hat{\mu}_i(n)$

A result in [AB10] states the following:

Theorem 4.1.6. For $0 \leq a \leq \frac{25n}{72H_1}$, where
\[
    H_1 = \sum_{\Delta_i > 0} \frac{1}{\Delta_i^2},
\]

the simple regret \( r_{n+1} \) in Algorithm 16 satisfies
\[
r_{n+1} \leq \max_{i \in [k]} \Delta_i \mathbb{P} \left( \Delta_{w_i} > 0 \right) \leq 2 \max_{i \in [k]} \Delta_i n k e^{-\frac{a}{50}}.
\]
Particularly, when \( a = \frac{25n}{72} H_1 \), \( r_{n+1} \) decays exponentially fast as \( n \to \infty \).

Proof. Let \( i^* \in [k] \) denote the best arm and \( c < 1 \). Consider the event
\[
G = \left\{ \mu_i(t) \in \left[ \hat{\mu}_i(t) - c \sqrt{\frac{a}{T_i(t)}}, \hat{\mu}_i(t) + c \sqrt{\frac{a}{T_i(t)}} \right] \text{ for all } i \in [k] \text{ and } t \in [n] \right\}.
\]
\( G \) occurs with probability at least \( 1 - 2k e^{-c^2 a/2} \). In the rest of the proof everything is conditioned on \( G \). For any sub-optimal \( i \), it will not be played if its confidence edge fails to compensate for the gap. Indeed, for \( T_i(t-1) \geq (c + 1)^2 a \Delta_i^{-2} \),
\[
\mathbb{P} \left( \text{UCB}_i^{(a)}(t) < \text{UCB}_i^{(a)}(t) \right) \leq \mathbb{P} \left( \mu_{i^*} < \mu_i + (c + 1) \sqrt{\frac{a}{T_i(t-1)}} \right) = 0.
\]
Hence, any sub-optimal arm \( i \) can be played at most \( (c + 1)^2 a \Delta_i^{-2} \) times. Suppose that
\[
n > 2a(c + 1)^2 \sum_{i : \Delta_i > 0} \frac{1}{\Delta_i^2} = 2a(c + 1)^2 H_1.
\]
Then on \( G \), \( T_{i^*}(n) \geq T_i(n) \) so that
\[
T_i(n) \geq \frac{4c^2 a}{\Delta_i^2} \quad \forall i \in [k] \text{ and } \Delta_i > 0
\]
would guarantee that \( i^* \) is optimal. To see \( T_i(n) \) must be of order \( \Delta_i^{-2} \), note that in round \( t \), if \( i^* \) is to be played, then for any sub-optimal \( i \in [k] \),
\[
\mu_{i^*} + (1 + c) \sqrt{\frac{a}{T_i(t-1)}} \geq \mu_i + (1 - c) \sqrt{\frac{a}{T_i(t-1)}},
\]
bounding \( T_i(t-1) \) from below as
\[
T_i(t-1) \geq \frac{(1 - c)^2}{4} \min \left\{ \frac{a}{\Delta_i^2}, \frac{T_i(t-1)}{(1 + c)^2} \right\}. \quad (4.1.4)
\]
It is easy to see that the same bound holds when \( i^* \) is not played. Hence (4.1.4) holds for all \( t \leq n + 1 \). Taking \( t = n + 1 \) combined with the assumption on \( n \),
\[
T_i(n) \geq \frac{(1 - c)^2 a}{4 \Delta_i^2}.
\]
Setting \( c = 1/5 \) (solve for \( (1 - c)^2/4 = 4c^2 \)) yields the desired result. \( \square \)
Remark 4.1.7. Taking \( a = \Theta(\log n) \) gives the same exploration rate as in the original UCB algorithm. The resulting bound for \( r_{n+1} \) decays polynomially, which is consistent with the lower bound obtained in Theorem 4.1.2.

Remark 4.1.8. The proof given here also works when \( \psi_n \) is chosen as the MPA, which only requires (4.1.3) on \( G \). One can also obtain a distribution-free bound for \( r_{n+1} \). Particularly, for \( a = a \log t (\alpha > 1) \) and \( \psi_n \) taken as the MPA,

\[
r_{n+1} = \Theta\left(\sqrt{\frac{ak \log n}{n}}\right).
\]

Whereas for the EBA, the current available distribution-free bound is of order \( \Theta(\sqrt{\log n}) \). More details could be found in [AB10].

Choosing an optimal \( a \) in Algorithm 16 is often difficult in practice, since it needs an oracle access to the sub-optimality gaps. To address this issue, an alternative algorithm called Successive Rejects was proposed in [AB10]. The Successive Rejects algorithm divides the exploration process into \( k-1 \) phases, eliminating the worst performing arm at the end of each phase. Here we introduce a similar version of the Successive Rejects algorithm called Sequential Halving [LS18], which is easier to state.

The Sequential Halving algorithm can be thought as the multi-phase UAS. The exploration process in the algorithm consists of \( L = \lceil \log k \rceil \) (base 2) phases, with each phase allocated \( \lceil n/L \rceil \) rounds for equal exploration. At the end of each phase, arms explored in that phase will be sorted in decreasing order according to their empirical means. The top half of the arms will survive into the next round while the bottom half get eliminated. The details of the algorithm are given below:

Algorithm 17: The Sequential Halving Algorithm

**Input:** \( n \): horizon  
**Output:** \( \psi_n \)

1. **Initialization:** \( \mathcal{A}_1 = [k] \), \( L = \lceil \log k \rceil \)
2. for \( \ell = 1, 2, \ldots, L \) do
3. \( \) Play each arm in \( \mathcal{A}_\ell \) exactly \( \lfloor n/L \rfloor \) times
4. \( \) Compute the empirical means of the arms in \( \mathcal{A}_\ell \) using the data in round \( \ell \), and sort in decreasing order: \( \hat{\mu}_1^{(\ell)} \geq \cdots \geq \hat{\mu}_{\lceil |A_\ell|/2 \rceil}^{(\ell)} \)
5. Take \( \mathcal{A}_{\ell+1} \) as the the arms in \( \mathcal{A}_\ell \) corresponding to \( \hat{\mu}_1^{(\ell)}, \ldots, \hat{\mu}_{\lceil |A_\ell|/2 \rceil}^{(\ell)} \)
6. end for

The next theorem gives an upper bound for the simple regret in the Sequential Halving algorithm.

**Theorem 4.1.9.** For convenience of statement, assume the components of \( \mu \) are arranged in decreasing order: \( \mu_1 \geq \cdots \geq \mu_k \). Then, the simple regret \( r_{n+1} \) in the Sequential Halving algorithm satisfies

\[
r_{n+1} \leq 3 \max_{i \in [k]} \Delta_i \log k \cdot e^{-\frac{n}{16H_2(\mu) \log k}},
\]  

(4.1.5)
where

\[ H_2(\mu) = \max_{i : \Delta_i > 0} \frac{i}{\Delta_i^2}. \]

Before giving the proof, note that

\[ H_2(\mu) \leq H_1(\mu) \leq (1 + \log k) H_2(\mu), \]

which implies that (4.1.5) is comparable to the one in Theorem 4.1.6.

**Proof.** Assume \( k = 2^L \) and ignore the floor/ceiling effect for convenience. In the \( \ell \)-th round, each arm will be played \( T_{\ell} = 2^{\ell-1} n / k L \) times. For any sub-optimal arm \( i \), using the sub-gaussian concentration inequality,

\[ \mathbb{P}(\hat{\mu}_{\ell,i} \geq \hat{\mu}_{\ell,1} | i \in A_\ell) \leq e^{-\frac{T_{\ell} \Delta_i^2}{4}}. \]

Let \( S(A, m) \) be the first \( m \) arms in \( A_m \), where \( A_m \) is the same as \( A \) but with arms arranged in increasing order. Suppose that \( 1 \in A_\ell \) and \( 1 \notin S(A_{\ell+1}) \). Then at least half of the arms in \( A_\ell \) are estimated above the empirical mean of arm 1, of which at least another half have their indices greater than \( |A_\ell|/4 \). That is,

\[ \sum_{i \in A_\ell \setminus S(A_\ell, |A_\ell|/4)} \mathbb{I}(\hat{\mu}_{\ell,i} \geq \hat{\mu}_{\ell,1}) \geq \frac{|A_\ell|}{4}, \]

and

\[ \min_{i \in A_\ell \setminus S(A_\ell, |A_\ell|/4)} i \geq \frac{|A_\ell|}{4}. \]

Therefore, using Markov’s inequality,

\[ \mathbb{P}(1 \notin A_{\ell+1} | 1 \in A_\ell) \leq \mathbb{P}\left( \sum_{i \in A_\ell \setminus S(A_\ell, |A_\ell|/4)} \mathbb{I}(\hat{\mu}_{\ell,i} \geq \hat{\mu}_{\ell,1}) \geq \frac{|A_\ell|}{4} \right) \leq \frac{4}{|A_\ell|} \cdot \frac{3}{4} |A_\ell| \cdot e^{-\frac{T_{\ell} \Delta_i^2}{4}} \leq 3e^{-\frac{n^2}{4|A_\ell|/4}} \leq 3e^{-\frac{n^2}{16|A_\ell|/4}}. \]

Summing \( \ell \) from 1 to \( L \) gives

\[ \mathbb{P}(1 \notin A_{L+1}) \leq 3L e^{-\frac{n^2}{16H_2(\mu)}} = 3 \log k e^{-\frac{n^2}{16 \log k H_2(\mu)}}. \]

The proof is finished by noting that \( r_{n+1} \leq \max_{i \in [k]} \Delta_i \mathbb{P}(1 \notin A_{L+1}) \).

**Remark 4.1.10.** Compared to UAS, Sequential Halving takes into account the varying scales of the sub-optimality gaps, thus has a better performance when the sub-optimality gaps are drastically different. For instance, let \( \Delta_2 = \Delta << 1 \) and \( \Delta_i = 1 \) for \( i > 2 \). In this case, the exponent in (4.1.2) is of order \( \Theta(n \Delta^2/k) \) while in (4.1.5) it is of order \( \Theta(n \Delta^2 / \log k) \).
4.1 BEST-ARM IDENTIFICATION

4.1.2 Best-Arm Identification with a Fixed Confidence

In the previous part we considered the bandits where the budget is limited by a fixed horizon, and the goal is to minimize the probability of \( \psi_n \) being sub-optimal\(^3\). We now reverse the roles of the constraint and the objective. Particularly, in the fixed confidence set-up, we are given a confidence level \( \delta > 0 \). The goal is find a good triple \((\pi, \tau, \psi)\): a policy \( \pi \), a stopping criterion \( \tau \) (stopping time) and a decision function \( \psi \in [k] \) (which is measurable to the sigma-algebra determined by \( \tau \)) such that for any admissible environment \( \mu \),

\[
\mathbb{P}_\mu \left( \tau < \infty, \Delta_\psi > 0 \right) \leq \delta. \tag{4.1.6}
\]

Existence of such a \( \tau \) is obvious by taking \( \tau = \infty \). Of course, this is not to be encouraged. From an economical point view, one wishes to find \( \tau \) such that \( \mathbb{E}[\tau] \) is as small as possible. It is not yet clear whether such an optimal \( \tau \) exists. In the following we will first describe an asymptotic lower bound for \( \mathbb{E}[\tau] \) satisfying (4.1.6), and then provide an algorithm that matches the lower bound as \( \delta \to 0 \). The complete analysis of the algorithm will not be given, but can be found in the original work [GK16]. A more specific and readable sketch could be found in Chapter 33 in [LS18]. Our focus will be on the main ideas only.

Let \( \mathcal{E} \) denote the set of admissible \( k \)-armed bandit environments. For any \( \mu \in \mathcal{E} \), let \( C_\mu \) be the set of bandits in \( \mathcal{E} \) whose optimal arms are different from the ones of \( \mu \). Then we have the following result.

**Theorem 4.1.11.** For any triple \((\pi, \tau, \psi)\) satisfying (4.1.6),

\[
\mathbb{E}_\mu[\tau] \geq c^*(\mu) \log \left( \frac{1}{4\delta} \right),
\]

where

\[
c^*(\mu) = \left( \sup_{a \in \mathcal{P}([k])} \inf_{\mu' \in C_\mu} \left( \sum_{i \in [k]} a_i D_{KL}(\mu, \mu'_i) \right) \right)^{-1}.
\]

\( D_{KL}(\cdot, \cdot) \) is the KL-divergence and \( \mathcal{P}([k]) \) is the set of distributions on \([k]\).

**Proof.** Assume that \( \tau \) is finite a.s. Let \( \Delta_\psi(\mu) \) be the sub-optimality gap of the decision \( \psi \) in the environment \( \mu \). For any \( \mu' \in C_\mu \), let \( G \) be the event that \( \tau < \infty \) and \( \Delta_\psi(\mu) > 0 \). It is easy to check that the complement of \( G \), up to a null set, is contained in event that \( \tau < \infty \) and \( \Delta_\psi(\mu') > 0 \). Hence, using the Bretagnolle-Huber inequality,

\[
2\delta \geq \mathbb{P}_\mu(G) + \mathbb{P}_{\mu'}(G^c) \geq \frac{1}{2} e^{-D_{KL}(\mathbb{P}_\mu, \mathbb{P}_{\mu'})}. \tag{4.1.7}
\]

\(^3\)The true objective is the expected sub-optimality gap at step \( n + 1 \); however, the way we analyze it is through upper bounding the probability of making an sub-optimal decision.
As opposed to the fixed horizon case, \( \mathbb{P}_\mu \) and \( \mathbb{P}_\mu' \) are probability measures on the sigma-algebra generated by \( \tau \). Therefore,

\[
D_{KL}(\mathbb{P}_\mu, \mathbb{P}_\mu') = \sum_{n=1}^{\infty} \mathbb{P}(\tau = n) \mathbb{E}_{\mu} \left[ \sum_{i=1}^{n} \log \frac{p_{\mu,s_i}}{p_{\mu',s_i}} \right] \tag{4.1.7}
\]

is the conditional relative entropy. Therefore,

\[
\leq \sum_{n=1}^{\infty} \mathbb{P}(\tau = n) \mathbb{E}_{\mu} \left[ \sum_{i=1}^{n} \log \frac{p_{\mu,s_i}} {p_{\mu',s_i}} \right] 
= \sum_{n=1}^{\infty} \mathbb{P}(\tau = n) \mathbb{E}_{\mu} \left[ \sum_{j \in [k]} \mathbb{E}_{\mu}[T_j(n)]D_{KL}(\mu_j, \mu'_j) \right] 
= \sum_{j \in [k]} \mathbb{E}_{\mu}[T_j(\tau)]D_{KL}(\mu_j, \mu'_j),
\]

where for the first inequality we used the fact that conditioning does not increase relative entropy, see Exercise 5.13 in [RS15]. Plugging the lower bound back in (4.1.7) yields

\[
\sum_{j \in [k]} \mathbb{E}_{\mu}[T_j(\tau)]D_{KL}(\mu_j, \mu'_j) \geq \log \left( \frac{1}{4\delta} \right).
\]

Dividing both sides by \( \mathbb{E}_{\mu}[\tau] \) and taking infimum over \( \mu' \) on both sides,

\[
\inf_{\mu' \in C_\mu} \sum_{j \in [k]} \frac{\mathbb{E}_{\mu}[T_j(\tau)]}{\mathbb{E}_{\mu}[\tau]} D_{KL}(\mu_j, \mu'_j) \geq \frac{\log \left( \frac{1}{4\delta} \right)}{\mathbb{E}_{\mu}[\tau]}, \tag{4.1.8}
\]

Noting that \( \sum_{j \in [k]} \mathbb{E}_{\mu}[T_j(\tau)]/\mathbb{E}_{\mu}[\tau] = 1 \) finishes the proof. \( \square \)

Theorem 4.1.11 inspires the construction of an asymptotically optimal stopping time. Since \( \mu \) is unknown, one needs to spend some time estimating \( \mu_i \) for \( i \in [k] \). This is necessary for any reasonable policy satisfying (4.1.6). Then one can use the estimated \( \mu_i \), together with (4.1.8) to determine if the expected accuracy is attained, therefore when to stop. The expectation in (4.1.8) can be replaced by the random time itself. Indeed, if at time \( t \), for some \( \eta \in C_{\hat{\mu}} \),

\[
Z_t := \inf_{\mu' \in C_\mu} \sum_{j \in [k]} T_j(t)D_{KL}(\hat{\mu}_j, \mu'_j) = \sum_{j \in [k]} T_j(t)D_{KL}(\hat{\mu}_j, \eta_j) \leq \beta_j(\delta) = \mathcal{O} \left( \log \left( \frac{1}{\delta} \right) \right),
\]

where \( \beta_j(\delta) \) is some function of order \( \log(1/\delta) \) and to be determined later, then it is unlikely that the expected accuracy is achieved by stopping immediately. Therefore it is advisable to keep moving towards the direction where accuracy can be improved most. For this, one calculates the difference of the time distribution between stopping immediately and the optimal one under the estimate, namely, \( \alpha_i(t) \), which is the \( i \)-th
component of the optimal convex combination coefficients achieving $c^*(\hat{\mu})$, and chooses the one that gives the biggest jump:

$$
\pi_{t+1} = \arg \max_{i \in [k]} \left( \alpha_i(t) - \frac{T_i(t)}{t} \right).
$$

The rationale behind the above greedy strategy is that $\hat{\mu}$ is close enough to $\mu$. Therefore, one should keep exploring the arms which are relatively underplayed. The decision function $\psi$ is taken as the EBA. The algorithm described here is called the Track-and-Stop algorithm.

**Algorithm 18:** The Track-and-Stop Algorithm

**Input:** $\delta$: confidence level  
**Output:** $(\pi, \tau, \psi)$

1. Choose each arm once and set $t = k$
2. while $Z_t \leq \beta(\delta)$ do  
3. if $\arg \min_{i \in [k]} T_i(t) \leq \sqrt{t}$ then  
4. $\pi_{t+1} = \arg \min_{i \in [k]} T_i(t)$ // forced exploration  
5. else  
6. $\pi_{t+1} = \arg \max_{i \in [k]} \left( \alpha_i(t) - \frac{T_i(t)}{t} \right)$ // constraint detection  
7. end if  
8. Update the $\pi_{t+1}$-th component of $\hat{\mu}$ and set $t = t + 1$  
9. end while  
10. $\tau = t$, $\psi = \arg \max_{i \in [k]} \hat{\mu}_i$

When the environment is Gaussian, the next theorem shows the asymptotic optimality of the regret given by the Track-and-Stop Algorithm.

**Theorem 4.1.12.** Let $\mu$ be a $k$-armed Gaussian bandit with a unique optimal arm. Take $\beta(\delta) = k \log(t^2 + t) + f^{-1}(\delta)$, where

$$
f(x) = e^{k-x} \left( \frac{x}{k} \right)^k \quad x \geq k.
$$

Then, the triple $(\pi, \tau, \psi)$ satisfies (4.1.6) and

$$
\lim_{\delta \to 0} \frac{E_{\mu}[\tau]}{\log \left( \frac{1}{\delta} \right)} = c^*(\mu).
$$

**Remark 4.1.13.** The choice of $\beta$ follows from the choice of concentration bounds used in the proof. One thing to note is that as $\delta \to \infty$, $\beta(\delta)$ is of order $\log(1/\delta)$.

### 4.2 Budget-Limited Bandits

Best arm identification problems have separate exploration and exploitation processes, with the former constrained by a budget. We now consider an alternative model
called the budget-limited bandits, where exploration and exploitation interact in a similar way as in the stochastic bandits, and pulling an arm for either purpose comes with a cost. Given a budget, the goal is to find a policy under which the cumulative regret is small when the budget is running out.

Budget-limited bandits were first introduced in [Tra+10], where each arm is associated with a fixed pulling cost. An algorithm called the $\epsilon$-First Policy was given to solve the corresponding optimization problem, which is similar to Algorithm 1 in chapter 1. Due to the difficulty of finding the trade-off between exploration and exploitation, more practical algorithms were proposed in a follow-up paper [Tra+12] by the same authors. Further generalization on relaxing the time-invariant cost can be found in [Din+13]. A comprehensive summary of the results in related fields can be found in [ZT18]. In the rest of this section, we will first introduce the set-up of the problem and then give two practical algorithms for it. We will closely follow the discussion in [Tra+12].

Consider a $k$-armed stochastic bandit. Let $B > 0$ be a fixed budget. For $i \in [k]$, $c_i > 0$ is the cost incurred by pulling $i$, and $\mu_i$ is the reward. $c_i$ are constants and known to the players. For simplicity, we assume that centered reward distributions are 1-subgaussian. Let $\pi$ be a policy under the budget constraint $B$. Let $T_{i,B}$ denote the total number of times played on $i$. The budget constraint enforces that

$$\sum_{i \in [k]} c_i T_{i,B} \leq B. \quad \text{a.s.} \quad (4.2.1)$$

Any policy satisfying (4.2.1) is called feasible. Similar to (1.1.2), the regret in this case is defined as

$$R_B = \max_{\pi'} \sum_{i \in [k]} \beta_i \mu_i - \sum_{i \in [k]} \mathbb{E}[T_{i,B}] \mu_i,$$

where $\beta_i$ is the total number of times played on $i$ under a deterministic feasible policy $\pi'$. To understand for what policy $R_B$ is small, we first need to know the best we can do when $\mu_i$’s are known. Assuming so, the above budgeted learning problem reduces to an unbounded knapsack problem\footnote{An unbounded knapsack problem is an optimization problem of the form:}

$$\max_{x} \sum_{i \in [k]} x_i v_i$$

subject to

$$\sum_{i \in [k]} x_i w_i \leq C, \quad x_i \in \mathbb{N} \quad \forall i \in [k].$$

This is known to be NP-hard. However, it can be approximately solved via a density-ordered greedy algorithm, see [KKM04]. Particularly, in each round, one chooses the arm giving the largest reward-to-cost ratio among all arms that could be played using the remaining budget, and plays it as many times as possible. The process terminates when no arm is affordable. If the solutions are allowed to be fractional, then the optimal solution can be easily calculated as $\beta_i = \left\lfloor \frac{B}{c_i} \right\rfloor$, where $i^* = \arg\max_{i \in [k]} \mu_i / c_i$. When $\mu_i$’s are unknown, this near-optimal algorithm can still be carried out but with an additional step of estimating $\mu_i$, for which we will use UCB on offer optimism.

The following algorithms, termed knapsack-based upper confidence bound exploration and exploitation (KUBE) and fractional knapsack-based upper confidence bound
4.2. BUDGET-LIMITED BANDITS

exploration and exploitation (fractional KUBE), respectively, combine the ideas from unbounded knapsack problems and UCB to provide a practical strategy for budgeted bandit learning.

Algorithm 19: The KUBE/fractional KUBE Algorithm

**Input:** \( \delta \): confidence level; \( B \): budget  
**Output:** \( \pi = (\pi_t)_{t \in [n]} \)

1. **Initialization:** \( t = 1 \) and \( B_1 = B \)
2. **while** pulling is feasible **do**
3. \( \mathcal{A}_t = \{i \in [k] : c_i \leq B_t\} \)
4. For \( i \in \mathcal{A}_t \), compute its \( \delta \)-UCB: \( UCB_i \)
5. Use \( UCB_i \) as the mean and approximately solve the corresponding knapsack problem via the density-ordered greedy algorithm. Denote the solution as \( m_{i,t} \) for each \( i \in \mathcal{A}_t \) // This step is for KUBE only
6. \( \pi_t \sim p_t \), where \( p_t(i) = \frac{m_{i,t}}{\sum_{i \in \mathcal{A}_t} m_{i,t}} \) (KUBE)  
\( \pi_t = \arg \max_{i \in \mathcal{A}_t} \frac{UCB_i}{c_i} \) (fractional KUBE)
7. \( B_{t+1} = B_t - c_{\pi_t} \)
8. \( t = t + 1 \)
9. **end while**

In [Tra+12], \( \delta \) is selected adaptively as in [ACF02], with \( \delta_t = t^{-1} \). Here, we set \( \delta = B^{-2} \) and prove an optimal instance bound for the regret in fractional KUBE. Similar results can be proved for KUBE but with more involved calculations.

**Theorem 4.2.1.** Let \( \delta = B^{-2} \). For \( i \in [k] \), define the reward-to-cost ratio as \( r_i = \frac{\mu_i}{c_i} \), and let \( i^* = \arg \max_{i \in [k]} r_i \). Let \( \Delta_i = r_i^* - r_i \) be the reward-to-cost ratio gap. Let \( c_{\min} = \min_{i \in [k]} c_i \). Suppose \( B \geq c_{\min} k \) (which is not essential but makes the statement simpler). The regret in the fractional KUBE algorithm is bounded by

\[
R_B \leq \frac{2r_i^*}{c_{\min}} + 16 \log B \sum_{i : \Delta_i > 0} \frac{1}{c_i \Delta_i}.
\]

**Proof.** Since a knapsack problem has a smaller admissible region than its fractional counterpart,

\[
R_B \leq r_i^* B - \sum_{i \in [k]} \mathbb{E}[T_{i,B}] \mu_i.
\]

Let \( n = \min\{t : B_{t+1} \leq c_{\min}\} \). It is clear that \( n \leq \frac{B}{c_{\min}} \) a.s..
CHAPTER 4. BANDITS WITH COST

Therefore, with probability at least \(1 - c^{-1}\min B \delta\),
\[
    r_i \leq \text{UCB}_i(t, \delta) \quad \forall t \in [n].
\]  
(4.2.2)

Conditional on (4.2.2), for each sub-optimal \(i\), supposing \(T_i(t-1) \geq 8c^{-2}_i\Delta_i^{-2}\log(1/\delta)\),
\[
    \mathbb{P} \left( \frac{\text{UCB}_i(t-1, \delta)}{c_i} \geq r_i \right) \leq \delta.
\]

Hence, with probability at least \(1 - (c^{-1}_i \min B + k - 1)\delta\), or \(1 - 2c^{-1}_i \min B \delta\), any sub-optimal \(i\) is played at most \(8c^{-2}_i\Delta_i^{-2}\log(1/\delta)\) times. Therefore,
\[
    R_B \leq \frac{2r_i \cdot B^2 \delta}{c_{\min}} + 8 \log \left( \frac{1}{\delta} \right) \sum_{i: \Delta_i > 0} \frac{1}{c_i \Delta_i}.
\]

Taking \(\delta = B^{-2}\) yields the desired result. 

**Remark 4.2.2.** When \(c_i = 1\) for all \(i \in [k]\), the budgeted bandit problem is equivalent to the classical stochastic bandit. In this case, both KUBE and fractional KUBE reduce to the UCB, and Theorem 4.2.1 recovers Theorem 1.2.4. Particularly, by the results in chapter 1, the instance bound given by both KUBE and fractional KUBE are optimal in terms of the leading order asymptotics.

**Remark 4.2.3.** The regret bound for KUBE derived in [Tra+12] is less optimal than the one for fractional KUBE, in terms of the constants. However, numerical experiments in [Tra+12] showed that KUBE outperformed fractional KUBE by a 30-40% margin. It is not yet clear why this occurs. On the other hand, at each step, fractional KUBE has smaller computational cost compared to KUEB, which requires a complete sort of the estimated reward-to-ratio values for each arm\(^5\).

**Remark 4.2.4.** Fixed cost assumption could be restrictive in some circumstances. A possible generalization is to assume that costs generated by the same arm are i.i.d. random variables. In this case, an extra step is needed to estimate the mean cost for each arm per round thus the upper confidence bonus requires some modification. This leads to the so-called UCB with Budget constraint and Variable cost algorithm (UCB-BV), see [Din+13]. Of course, one may also use the idea of Thompson sampling as described in Algorithm 5, sampling the reward and cost parameters from the two independent posterior distributions, respectively, and choosing the arm giving the best reward-to-cost ratio. This is the Budgeted Thompson Sampling (BTS). For more information, see [Xia+15]. Analysis of more general cost assumptions can be found in [CES20].

### 4.3 Multi-Fidelity Bandits

We consider an alternative budget bandit model called multi-fidelity bandits [Kan+16]. Similar to the budget-limited bandits in the previous section, multi-fidelity bandits also

\(^5\)Note that KUBE updates the reward-to-ratio value of one arm at a time, so it might be negligible.
4.3. **MULTI-FIDELITY BANDITS**

Involves a budget constraint that should not be surpassed. However, the cost assumption in a multi-fidelity bandit is quite different. Instead of attaching a pulling cost to the $k$ arms respectively, it assumes that the reward of each arm can be observed at $M$ different fidelities, with each fidelity $m \in [M]$ costing $\lambda(m)$ for an observation. Often, a fidelity represents an approximation to the original problem from the computational side. A high-fidelity observation has better accuracy and is therefore associated with a larger cost. In the following we assume that

- Cost is a strict increasing function with fidelity. Particularly, $0 < \lambda(1) < \lambda(2) \leq \cdots < \lambda(M)$;
- Denote by $\{\xi^{(m)}\}_{m \in [M]}$ the approximation errors which is a decreasing sequence in $m$ with $\xi^{(M)} = 0$. For every $i \in [k]$ and $m \in [M]$, the $m$-fidelity approximation (mean) for $\mu_i$ is $\mu^{(m)}_i$, and satisfies $|\mu_i - \mu^{(m)}_i| \leq \xi^{(m)}$;
- For each arm, the observation at fidelity $m$ is a $1$-sub-gaussian random variable$^6$

Let $\Lambda > 0$ be the budget. Let $\pi_t$ be a policy. Here $\pi_t = (\pi_{t,1}, \pi_{t,2})$, where the first component is the arm to play and the second component is the chosen fidelity. Let

$$n = \max \left\{ t : \sum_{j=1}^{t} \lambda^{(\pi_{j,2})} \leq \Lambda \right\}.$$  

To motivate the definition of the regret, note that our goal is the learn the $M$-fidelity mean reward at each arm, and playing the same arm using different fidelities lead to different absolute cumulative rewards. This is obviously not we have intended to get. The problem here is the interpretation of $\mu$ as the average reward per time, which does not scale with the budget under the multi-fidelity setting. In this case, it would be more natural to consider budget-normalized expected rewards. For $i \in [k]$, $\mu_i$ is interpreted as the average reward by utilizing a unit of the budget. This sets a stark distinction between multi-fidelity bandits and budget-limited bandits. Treat a unit of budget as an attempt in exploration. An $m$-fidelity estimate for $\mu_i$ then requires exploring $i \lambda^{(m)}$ times. This somehow reminds us of some algorithms we have seen before, such as the ETC and the Elimination algorithm. The difference, however, is that a low-fidelity (poor exploration) affects the bias of the estimator instead of its variance. Based on these observations, we define the regret for multi-fidelity bandits as

$$R_\Lambda = \Lambda \mu_{i^*} - E \left[ \sum_{t \in [n]} \lambda^{(\pi_{t,2})} \mu_{\pi_{t,1}} \right],$$  

where $i^*$ is the optimal arm. It is easy to check that up to a constant, $R_\Lambda$ can be written as

$$E \left[ \sum_{t \in [n]} \lambda^{(\pi_{t,2})} \Delta_{\pi_{t,1}} \right] \leq R_\Lambda = E \left[ \sum_{t \in [n]} \lambda^{(\pi_{t,2})} \Delta_{\pi_{t,1}} \right] + \text{constant term},$$  

$^6$Sub-gaussian assumption can be generalized as long as appropriate concentration inequalities exists.
where $\Delta_i$ is the sub-optimality gap between $i$ and $i^*$. We will now leave the constant term out since it will not affect the order of the bound for $R_{\Lambda}$. To get a decomposition of $R_{\Lambda}$ as in (1.1.2), we need a few more notations. Let $T_{i}^{(m)}(t)$ be the number of times of playing $i$ at fidelity $m$ before time $t$. Then,

$$R_{\Lambda} = \sum_{i \in [k]} \Delta_i \sum_{m \in [M]} \hat{\lambda}^{(m)} \mathbb{E} \left[ T_{i}^{(m)}(m) \right].$$

We now describe an algorithm called the Multi-fidelity UCB (MF-UCB) for learning in multi-fidelity bandits. MF-UCB was first introduced in [Kan+16]; it is algorithmic structure resembles the Elimination algorithm in Algorithm 4. For each $i \in [k]$, MF-UCB starts with the low-fidelity estimate for $\mu_i$. An upper confidence criterion is then used to determine which arm to play in the next round. The fidelity will be chosen as the lowest fidelity on which the arm needs further exploration. The details of the algorithm are given as follows:

**Algorithm 20: The MF-UCB Algorithm**

**Input:** $\lambda^{(m)}$: fidelity update parameters; $\Lambda$: budget; $\hat{\lambda}^{(m)}$: fidelity approximation errors; $\rho$: confidence level parameter  
**Output:** $\pi_t$: action sequence

1. **Initialization:** Play each arm once and observe using the lowest fidelity; $t = k + 1$ and $\Lambda_t = \Lambda - k \hat{\lambda}^{(1)}$
2. **while** pulling is feasible **do**
3. **for** $i = 1, \ldots, k$ **do**
4. **for** $m = 1, \ldots, M$ **do**
5. $B_{i}^{(m)}(t) = \hat{\mu}_{i}^{(m)}(t - 1) + \sqrt{\frac{\rho \log t}{T_{i}^{(m)}(t - 1)}} + \hat{\lambda}^{(m)}$, // optimistic estimate using UCB and the fidelity approximation error where $\hat{\mu}_{i}^{(m)}(t - 1)$ is the sample mean of $\mu_{i}^{(m)}$ estimated before $t$.
6. **end for**
7. $B_i(t) = \min_{m \in [M]} B_{i}^{(m)}(t)$ // Be conservative while being optimistic
8. **end for**
9. $\pi_{t,1} = \arg \max_{i \in [k]} B_i(t)$
10. $\pi_{t,2} = \min \left\{ m \leq M : \sqrt{\rho \log t / T_{\pi_{t,1}}^{(m)}(t - 1)} \geq \lambda^{(m)} \right\}$ // determine the lowest under-explored fidelity
11. $\Lambda_{t+1} = \Lambda_t - \lambda^{(\pi_{t,2})}$
12. $t = t + 1$
13. **end while**

In the definition of $B_{i}^{(m)}(t)$, the second term comes from the confidence level under the sub-gaussian assumption, and the third term is the error of the $m$-fidelity approximation, which will not be affected as $T_{i}^{(m)}(t - 1)$ increases. Therefore, when $T_{i}^{(m)}(t - 1)$ is so large that the upper confidence bonus falls within the same level as $\hat{\lambda}^{(m)}$, one probably should go to the next fidelity for further exploration. This observation suggests
choosing $\gamma^{(m)} = O(\xi^{(m)})$. Indeed, it was shown in [Kan+16] that taking

$$\gamma^{(m)} = \sqrt{\frac{\lambda^{(m)}}{\lambda^{(m+1)}}} \xi^{(m)} \leq \xi^{(m)}$$

gives a good trade-off.

In order to see how Algorithm 20 works, note that $\min_t \mathbb{E}_{i^*} \Delta_t \leq \Delta_{i^*}$ with high probability. We now condition on this event. For each sub-optimal $i \in [k]$, it will be played regularly until it enters the $m$-fidelity approximation where

$$\mu_i - (\mu_i^{(m)} + \xi^{(m)}) \geq (1+c)\gamma^{(m)}$$

for some $c > 0$. We may choose $c = 1$ for convenience. Let

$$\mathcal{K}^{(m)} = \{ i \in [k] : (4.3.1) \text{ holds for } m \text{ and does not hold for } m' < m \}.$$

We expect that first $m$ fidelities are able to identify the sub-optimality of the arms in $\mathcal{K}^{(m)}$ with high probability. This is made precise by the following theorem:

**Theorem 4.3.1.** Let $\rho > 4$. There exists some $\Lambda_0 > 0$ such that for all $\Lambda > \Lambda_0$,

$$R_\Lambda \leq \log \left( \frac{\Lambda}{\lambda^{(M)}} \right) \sum_{m \in [M]} \sum_{i \in \mathcal{K}^{(m)}} \Delta_i \left( \frac{\lambda^{(m)}}{\mu_i - (\mu_i^{(m)} + \xi^{(m)})} \right)^2.$$

The details of the proof can be found in [Kan+16]. Here we give two remarks on the immediate consequences.

**Remark 4.3.2.** It is clear from the analysis above that the arms with large sub-optimality gaps can be identified using low-fidelity approximations, i.e., they belong to $\mathcal{K}^{(m)}$ where $m$ is small. Those with small sub-optimality gaps will finally end up in the $M$-fidelity phase. This implies that MF-UCB works no differently from the original UCB for the hard-to-distinguish arms, but makes it more economical to find the easy-to-distinguish ones provided that a multi-fidelity approximation scheme is available.

**Remark 4.3.3.** MF-UCB is at most a constant times worse than UCB. To see this, note that every arm in the UCB is observed using the highest-fidelity. For $m \in [M]$,

$$\Delta_i \leq \mu_i - \mu_i^{(m)} + \xi^{(m)} = \left( \mu_i^{(m)} - (\mu_i^{(m)} + \xi^{(m)}) \right) + 2\xi^{(m)}$$

$$\leq \left( 1 + \sqrt{\frac{\lambda^{(m+1)}}{\lambda^{(m)}}} \right) \left( \mu_i^{(m)} - (\mu_i^{(m)} + \xi^{(m)}) \right).$$

Therefore,

$$\frac{\lambda^{(m)}}{(\mu_i^{(M)} - (\mu_i^{(M)} + \xi^{(M)}))^2} \leq 4.$$
On the other hand, one would expect that MF-UCB demonstrates a superior performance over UCB if for $i \in \mathcal{K}_m$, $\mu_i - (\mu_i^{(m)} + \varepsilon^{(m)})$ is close to $\Delta_i$, and $\lambda^{(M)} \gg \lambda^{(m)}$. A possible formulation of the former is the following fast decay condition:

$$
\sum_{\ell \in [m]} \frac{1}{(\zeta^{(\ell)})^2} \leq \frac{1}{(\zeta^{(m+1)})^2}.
$$
Chapter 5

Gittins Index

5.1 Markov Decision Processes

In this section, we give an informal introduction to the Markov Decision Processes. Compared to bandit learning, the new framework allows one to make decisions with future planning. We will not be snared with the existence and measurability issues. Instead, we shall be focused on the ideas of optimization and control. Our discussion will closely follow [Sze10] and [LS18].

Let $\mathcal{S}$ and $\mathcal{A}$ be the state and action set, respectively. Here we assume $\max\{|\mathcal{S}|, |\mathcal{A}|\} < \infty$. Let $\{S_t\}_{t \in \mathbb{N}}$ be an $\mathcal{S}$-valued random process. A reward function $r_a(i)$ is defined from $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}$. For $a \in \mathcal{A}$, let $P_a(\cdot, \cdot) : \mathcal{S} \times \mathcal{S} \to [0, 1]$ be the transition kernel associated with $a$. If an action $a$ is taken at $t$, then $S_t$ evolves according to $P_a$, i.e.,

$$
\mathbb{P}(S_{t+1} | \text{history}) = P_a(S_t, S_{t+1}),
$$

where the history contains information of the past states and associated actions. These ingredients together are called a Markov Decision Process (MDP).

A specification of actions at each time is given by a policy $\pi = \{\pi_t\}_{t \in \mathbb{N}} \subset \mathcal{A}^\mathbb{N}$, which is defined as a sequence adapted to the history. The action sets here are assumed state-independent. $\pi$ is called Markov if $\pi_t$ depends only on $S_t$ for $t \in \mathbb{N}$. A Markov policy is called stationary if it has no time-dependence. $\pi$ is called deterministic if $\pi_t$ is equal to its conditional expectation on the history. The coupling of an MDP and a Markov policy is called a Markov Reward Process (MRP).

In a MRP, a quantity of immense interest is the cumulative reward $R_\pi$:

$$
R_\pi = \mathbb{E}\left[ \sum_{t \in \mathbb{N}} r_{\pi_t}(S_t) \right].
$$

\footnote{Generally, for $(i, a) \in \mathcal{S} \times \mathcal{A}$, the reward $r$ is a random variable with some distribution $\nu_a(i)$. The reward function $r_a(i)$ is taken as expected reward $\mathbb{E}_{\nu_a}[r]$.}

\footnote{There are two reasons for considering Markov policies. First of all, a Markov policy preserves the Markovian structure (which may be time-inhomogeneous) of $\{S_t\}_{t \in \mathbb{N}}$ under coupling. Secondly, when $\mathcal{A}$ is finite, it can be shown that every policy is dominated by a Markov policy, see [Put14].}
However, (5.1.1) is infinite in general. To fix this, we consider two modifications of $R_\pi$ which are frequently used in practice: $\alpha$-discounted cumulative reward and long-run average reward:

$$R_\pi = \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} r_{\pi_t}(S_t) \right], \quad 0 < \alpha < 1 \quad (\alpha\text{-}discounted \ cumulative \ reward) \quad (5.1.2)$$

$$R_\pi = \limsup_{n \to \infty} \frac{1}{n} \mathbb{E} \left[ \sum_{t \in [n]} r_{\pi_t}(S_t) \right] \quad (long\text{-}run \ average \ reward) \quad (5.1.3)$$

For any initial distribution $\mu$ on $S$, our goal is to find a policy $\pi^*$ such that

$$\mathbb{E}[R_{\pi^*}] = \sup_{\pi} \mathbb{E}[R_\pi],$$

where $R_\pi$ is defined in (5.1.2) or (5.1.3). At first sight, it is not clear at all if such a $\pi^*$ exists. In the rest of the section, we will introduce a systematic approach called Bellman’s optimality criterion to explicitly construct $\pi^*$. In both cases, we will show that $\pi^*$ can be chosen as a stationary deterministic policy and obtained via dynamic programming. We will discuss $\alpha$-discounted cumulative reward and long-run average reward separately.

### 5.1.1 $\alpha$-Discounted Cumulative Reward

Let $\Pi$ and $\Pi_{sd}$ be the set of Markov policies and stationary deterministic policies, respectively. When $R_\pi$ is the discounted cumulative reward, the celebrated Blackwell’s Theorem [Bla65] tells us that if $S$ is finite and action sets are the same for all states, then

$$\sup_{\pi \in \Pi} R_\pi = \sup_{\pi \in \Pi_{sd}} R_\pi.$$ 

Therefore it is enough to consider policies in $\Pi_{sd}$ only. Notice that $\pi \in \Pi_{sd}$ is uniquely determined by identifying $\pi(i)$ for every $i \in S$, and each $\pi(i)$ can take any element in $A$. Hence, $|\Pi_{sd}| = |A|^{|S|}$. Also, for $\pi \in \Pi_{sd}$,

$$R_\pi = \sum_{i \in S} \mu(i)v_\pi(i) \leq \sum_{i \in S} \mu(i)v^{*}(i),$$

where $v_\pi(i)$ is the $\alpha$-discounted cumulative reward started at $i$\footnote{This is also called the value function with respect to $\pi$.}, and $v^{*}(i) = \sup_{\pi \in \Pi_{sd}} v_\pi(i)$. We will show that there exists a stationary deterministic policy that achieves $v^{*}(i)$ for all $i \in S$. The main idea lies in the one-step iteration formula (Bellman’s equations): For $\pi \in \Pi_{sd}$ and $i \in S$, using the Markov property,

$$v_\pi(i) = r_{\pi(i)}(i) + \alpha \sum_{j \in S} P_{\pi(i)}(i,j)v_\pi(j). \quad (5.1.4)$$
5.1. MARKOV DECISION PROCESSES

Written in a compact form,

\[ v_\pi = r_\pi + \alpha P_\pi v_\pi, \]  

(5.1.5)

where \( v_\pi = (v_\pi(i))_{i \in S} \), \( r_\pi = (r_\pi(i))_{i \in S} \) and \( P_\pi = ((P_\pi(i, j))_{i, j \in S} \) and \( P_\pi^T \). If define \( T_\pi = r_\pi + \alpha P_\pi : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|} \), then \( v_\pi \) is a fixed point under \( T_\pi \). It is easy to check that \( T_\pi \) is an \( \alpha \)-contraction under the \( \ell_\infty \) norm. Hence, \( v_\pi \) is unique and can be computed using the fact that \( v_\pi = \lim_{n \rightarrow \infty} T_\pi^n u \) for any \( u \in \mathbb{R}^{|S|} \). Of course, since everything is linear, one can simply solve \( v_\pi \) via formula

\[ v_\pi = (I - \alpha P_\pi)^{-1} r_\pi. \]

The idea of iteration extends to the situations where the operator is not affine-linear. For \( i \in S \), let \( \pi^\ast(i) \) be a policy such that \( v_{\pi^\ast(i)}(i) = v^\ast(i) \) (achievable since \( |S|, |A| < \infty \)). Then,

\[ v^\ast(i) = r_{\pi^\ast(i)}(i) + \alpha \sum_{j \in S} P_{\pi^\ast(i)}(i, j) v_{\pi^\ast(i)}(j) \leq r_{\pi^\ast(i)}(i) + \alpha \sum_{j \in S} P_{\pi(i)}(i, j) v^\ast(j) \]

\[ \leq \sup_{\pi \in \Pi_{ad}} \left( r_{\pi(i)} + \alpha \sum_{j \in S} P_{\pi(i)}(i, j) v^\ast(j) \right). \]

Note that for \( v^\ast(i) \) the supremum in the bound only uses the information of \( \pi \) at \( i \). Therefore,

\[ v^\ast \leq \sup_{\pi \in \Pi_{ad}} (r_\pi + \alpha P_\pi v^\ast) \quad (5.1.6) \]

where \( v^\ast = (v^\ast(i))_{i \in S} \), and the inequality holds component-wise. Similar to (5.1.5), if define \( T^\ast = \sup_{\pi \in \Pi_{ad}} (r_\pi + \alpha P_\pi) : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|} \), then \( v^\ast \) satisfies

\[ v^\ast \leq T^\ast v^\ast. \]  

(5.1.7)

It is easy to check that \( T^\ast \) is also an \( \alpha \)-contraction under the \( \ell_\infty \) norm, thus has a unique fixed point \( v^\ast \), which can be calculated as \( v^\ast = \lim_{n \rightarrow \infty} (T^\ast)^n v^\ast \). We claim that \( v^\ast \) is the value function corresponding to some \( \pi \in \Pi_{ad} \). This follows by observing that \( v^\ast \) satisfies (5.1.5) with \( \pi \) chosen as the policy \( \pi^\ast \) achieving the supremum in \( v^\ast = T^\ast v^\ast \). Therefore, \( v^\ast \leq v^\ast \). On the other hand, \( (T^\ast)^n v^\ast \) is a non-decreasing sequence thanks to (5.1.7), which implies \( v^\ast \geq v^\ast \). Hence, \( v^\ast = v^\ast \), and \( \pi^\ast \) is a stationary deterministic policy that is optimal for all initial starts.

The above analysis also provides a way to find the optimal policy. For example, one can compute \( T^\ast v, (T^\ast)^2 v, \ldots, (T^\ast)^n v \) until some convergence criteria is met. The terminal solution, say \( (T^\ast)^n v \), may not be a valid value function in general; however, it does give rise to a genuine value function no worse than itself. Suppose that \( v \) is a value function associated with \( \pi \) and \( T^\ast v = T^\ast v > v \) for some \( \pi^\prime \). Then

\[ v_{\pi^\prime} = \lim_{n \rightarrow \infty} T^\ast v \geq T^\ast v = T^\ast v, \]

where for the first inequality we used that \( T^\ast v \geq T^\ast u \) if \( v \geq u \). This approach is called value iteration. An alternative approach is the policy iteration. It starts from an

\footnote{For every \( \pi \in \Pi_{ad} \), one can check that \( v_\pi \) also satisfies (5.1.7).}
arbitrary policy and computes the value function under the policy (This step is called policy evaluation and involves solving (5.1.5)). The value function is then fed back to update the policy (This step is called policy improvement and involves applying $T^*$).

The computational cost of policy iteration is much higher than that of value iteration due to the policy evaluation step.

### 5.1.2 Long-Run Average Reward

Now consider $R_\pi$ as the long-run average reward defined in (5.1.3). Similar to the discounted case, we will restrict to the stationary deterministic policies in the discussion and find a best policy among them. We will then show that this policy is actually optimal among all Markov policies.

Fix $\pi$ as a stationary policy. In the MDP set-up, \( \{S_t\}_{t \in \mathbb{N}} \) is a time-homogeneous Markov chain with its transition kernel given by

\[
P_{\pi}(i, j) = P_{\pi(i)}(i, j) \quad i, j \in S.
\]

For the moment we assume that \( \{S_t\}_{t \in \mathbb{N}} \) is ergodic, from which one can deduce that \( \{S_t\}_{t \in \mathbb{N}} \) admits a unique stationary distribution \( \nu_\pi \) such that

\[
\lim_{n \to \infty} P^n_{\pi} = \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} P^{t-1}_{\pi} \to V_\pi := (\nu_\pi, \cdots, \nu_\pi)^T.
\]

Therefore,

\[
R_\pi = \limsup_{n \to \infty} \frac{1}{n} \mathbb{E} \left[ \sum_{t \in [n]} r_\pi(S_t) \right] = \mathbb{E}_{\nu_\pi}[r_\pi(x)] = \lim_{n \to \infty} \mathbb{E}[r_\pi(S_n)],
\]

where \( r_\pi \) is the same as defined in section 5.1.1. For any initial distribution \( \mu \), interpolating from 0 to \( \infty \) yields

\[
\mu^T (V_\pi - I) r_\pi = \mu^T \sum_{n=0}^{\infty} (P_\pi - I) P^n_\pi r_\pi
\]
\[
= \mu^T \sum_{n=0}^{\infty} \mathbb{E} \left[ (P_\pi - I)(P^n_\pi - V_\pi)r_\pi \right]
\]
\[
= \mu^T (P_\pi - I) \mathbb{E} \left[ \sum_{n=0}^{\infty} (P^n_\pi - V_\pi) r_\pi \right] \Phi_{\pi}
\]

Assuming $\Phi_{\pi}$ is well-defined, then $\Phi_{\pi}(i)$ represents the cumulative advantage of starting at $i$ over starting at $\nu_\pi$. Since the above equality holds for all initial distributions,

\[
(V_\pi - I) r_\pi = (P_\pi - I) \Phi_{\pi} \Rightarrow V_\pi r_\pi + \Phi_{\pi} = r_\pi + P_\pi \Phi_{\pi}.
\]

(5.1.8)
5.1. MARKOV DECISION PROCESSES

One can prove that (5.1.8) actually holds without assuming ergodicity and that \( \Phi \) exists (in which case the \( \Phi \) is replaced by its Cesàro sum). Note that \( V_{r \pi} \) can be written as \( R_1 \), where 1 is the all-one vector in \([\mathbb{R}]^S\). For fixed \( \pi \in \Pi_{sd} \),

\[
R_1 + \Phi \leq \sup_{\pi' \in \Pi_{id}} \left( r_{\pi'} + P_{\pi'}\Phi \right).
\]  

(5.1.9)

It can be shown that there exists a pair \((\rho, \Phi)\) such that

\[
\rho 1 + \Phi = \sup_{\pi' \in \Pi_{id}} \left( r_{\pi'} + P_{\pi'}\Phi \right).
\]  

(5.1.10)

It is worth mentioning that \( \Phi \) is not unique as its translation also satisfies (5.1.10). Let \( \pi^* \in \Pi_{sd} \) be a policy achieving the supremum in (5.1.10). We claim that \( \pi^* \) is optimal among all Markov policies and \( R_{\pi^*} = \rho \). For the latter, note that

\[
R_{\pi^*} 1 + \Phi_{\pi^*} = \left(r_{\pi^*} + P_{\pi^*}\Phi_{\pi^*}\right)
\]

\[
\rho 1 + \Phi = \left(r_{\pi^*} + P_{\pi^*}\Phi\right).
\]

Taking the difference of two equations yields

\[
(R_{\pi^*} - \rho) 1 = (P_{\pi^*} - I)(\Phi_{\pi^*} - \Phi),
\]

which combined with \( P_{\pi^*} \) being a stochastic matrix implies \( R_{\pi^*} = \rho \). For the former, we fix a deterministic Markov policy\(^5\) \( \pi \), and define its policy function at time \( t \) by \( \pi^t : S \to A \), i.e., \( \pi^t_i = \pi^t(X_t) \). The \( t \)-step transition kernel is \( P^{(t)}_\pi = P_\pi \cdots P_{\pi^t-1} \). Note that we cannot directly raise the power to \( t \) due to the potential non-homogeneity. Therefore,

\[
R_{\pi^*} = \mu^T \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} P^{(t)}_\pi r_{\pi^t}.
\]

(5.1.11)

For each \( t \),

\[
P^{(t)}_\pi r_{\pi} = P^{(t)}_\pi (r_{\pi^t} + P_{\pi^t}\Phi - P_{\pi^t}\Phi)
\]

\[
\leq P^{(t)}_\pi (r_{\pi^t} + P_{\pi^t}\Phi - P_{\pi^t}\Phi)
\]

\[
= P^{(t)}_\pi (\rho 1 + \Phi - P_{\pi^t}\Phi)
\]

\[
= \rho 1 + P^{(t)}_\pi - P^{(t+1)}_\pi.
\]

Substituting the estimate into (5.1.11) yields \( R_{\pi} \leq \rho = R_{\pi^*} \).

Markov Decision (Reward) Processes are one of the central topics in Reinforcement Learning (RL), where the transition kernels need to be estimated together with the policies. This is beyond the scope of the notes and we refer to the famous book [SB18] for further information. On the other hand, Bellman’s optimality criterion can be used to derive Bayesian optimal policies for a class of Bayesian bandit models. The rest of the chapter will be devoted to understanding how this can be done. We will need another tool called optimal stopping which will be introduced next.

\(^5\)Non-deterministic Markov policies can be shown similarly but with more complicated notations.
CHAPTER 5. GITTINS INDEX

5.2 Optimal Stopping

In this section, we introduce some basic results in optimal stopping. Our exposition will be based on [Sze10] and [Fer02]. We start with a finite-time optimal stopping problem. Let \( \{X_t\}_{t \in \mathbb{N}} \) be a random process adapted to a filtration \( \{\mathcal{F}_t\}_{t \in \mathbb{N}} \). One may think that \( X_t \) represents the reward at \( t \). Let \( \mathcal{H}_n \) be the set of stopping times \( \tau \) which are bounded by \( n \) a.s., where \( n \) is a fixed integer. The goal of optimal stopping with horizon \( n \), roughly speaking, is to seek a \( \tau^* \in \mathcal{H}_n \) to maximize the expected reward collected at \( \tau \). This can be equivalently stated as finding a \( \tau^* \in \mathcal{H}_n \) such that

\[
E[X_{\tau^*}] = \sup_{\tau \in \mathcal{H}_n} E[X_{\tau}]. \tag{5.2.1}
\]

Of course, it is far from obvious that such a \( \tau^* \) even exists. In the following, we will describe an explicit construction of \( \tau^* \) which satisfies (5.2.1) using the idea of backwards induction.

Suppose that we are at time \( t \) and observe \( X_t \). For the next step, we can either move to \( t + 1 \) or stop immediately. Since we are not allowed to peek into the future, the best we can hope for taking a step forward at present is the conditional expectation of the maximum expected gain at \( t + 1 \). Intuitively, if this conditional expectation is better than the current observation, then move forward, otherwise stop. As we will see, the resulting stopping time defined in this way is optimal in the sense of (5.2.1).

For construction, we need to define the maximum expected gain \( Y_t \) at time \( t \). As there are two possible actions, \( Y_t \) is therefore given by the maximum of their respective gain. If stopping immediately, the gain is just \( X_t \), otherwise the expected gain would be

\[
E[Y_{t+1} \mid \mathcal{F}_t].
\]

This leads to the so-called Snell’s envelope:

\[
Y_t = \max \{ X_t, E[Y_{t+1} \mid \mathcal{F}_t] \},
\]

with \( Y_n = X_n \). It follows from definition that \( \{Y_t\}_{t \in [n]} \) is a supermartingale, i.e., the smallest supermartingale dominating \( \{X_t\}_{t \in [n]} \). We next prove that

\[
\tau^* = \min \{ t : X_t = Y_t \} \tag{5.2.2}
\]

is optimal.

**Theorem 5.2.1.** Let \( \tau^* \) be defined as in (5.2.2). Assuming \( X_t \) is integrable for all \( t \in [n] \), then \( E[X_{\tau^*}] = \sup_{\tau \in \mathcal{H}_n} E[X_{\tau}] \).

**Proof.** Let \( \tau \in \mathcal{H}_n \). Since \( Y_t \) is a supermartingale, and \( Y_t \geq X_t \) for all \( t \in [n] \), by optional stopping theorem, \( \{Y_{t \wedge \tau}\}_{t \in [n]} \) is a supermartingale, and

\[
E[X_t] \leq E[Y_t] \leq E[Y_1]. \tag{5.2.3}
\]

On the other hand, note that \( \{Y_{t \wedge \tau}\}_{t \in [n]} \) is a martingale. Indeed, it follows from direct computation that for \( A \in \mathcal{F}_t \),

\[
E \left[ Y_{t \wedge \tau} ; A \right] = E \left[ Y_{t+1 \wedge \tau} ; A \cap \{ \tau^* \leq t \} \right] + E \left[ Y_{t+1 \wedge \tau} ; A \cap \{ \tau^* \geq t + 1 \} \right]
\]

\[
= E \left[ Y_{t \wedge \tau} ; A \cap \{ \tau^* \leq t \} \right] + E \left[ Y_{t+1} \wedge \mathcal{F}_t ; A \cap \{ \tau^* \geq t + 1 \} \right]
\]

\[
= E \left[ Y_{t \wedge \tau} ; A \cap \{ \tau^* \leq t \} \right] + E \left[ Y_t ; A \cap \{ \tau^* \geq t + 1 \} \right]
\]

\[
= E \left[ Y_{t \wedge \tau} ; A \right].
\]
Applying optional stopping theorem again yields
\[ E[X_\ast] = E[Y_\ast] = E[Y_1]. \] (5.2.4)

Combining (5.2.3) and (5.2.4) and taking supremum over \( \tau \) in \( H_\ast \) finish the proof. \( \square \)

We now present a similar result when \( \{X_t\}_{t \in \mathbb{N}} \) is a sequence defined on a Markov chain. In this case, \( \tau \) can be any stopping time that is finite a.s., and we will show that there exists a simple characterization for \( \ast \) which resembles Bellman’s optimality criterion (5.1.7). To begin with, consider a time-homogeneous Markov chain \( \{S_t\}_{t \in \mathbb{N}} \) with a finite state space \( S \) and transition kernel \( P : r : S \rightarrow \mathbb{R} \) is the reward function. Let \( 0 < \alpha < 1 \) be the discount factor. Suppose that a fixed cost \( \gamma \) needs to be paid in order to receive the reward and enter the next round. Therefore, the total discounted (net) reward in the first \( t \) rounds with continuation is
\[ X_t = \sum_{s \in \{t-1\}} \alpha^{s-1}(r(S_s) - \gamma). \]

Our goal is to find a stopping time \( \tau \) such that \( E[X_\tau] \) is maximized. To get some intuition, note that if \( \gamma \leq \min_{i \in S} r(i) \), one should keep going forever since the expected reward in the future is always positive. If \( \gamma > \max_{i \in S} r(i) \), then it would be optimal not to start at all. One might expect that any \( \gamma \) in between gives a trade-off between moving and stopping.

To derive an explicit formula for the optimal \( \ast \), notice that the model considered here can be treated as a Markov Decision Process. A player at state \( i \in S \) can take actions in \( A = \{0, 1\} \), where 1 means moving forward following \( P \) and the reward is \( r(i) - \gamma \), while 0 means stopping (absorbing state at \( i \)) and the reward is null. One should be aware of a subtle difference between what is described here and the optimal stopping problem on Markov chains. Stopping times only give policies (denoted by \( \Pi' \)) that take 0 once and for all, which is a subset of \( \Pi \). However, it can be shown that the suprema of cumulative reward with respect to both \( \Pi' \) and \( \Pi \) are the same. Based on this observation, we have that for \( \{S_t\}_{t \in \mathbb{N}} \) started at \( i \), \( \sup_{\tau} E[X_\tau] = v'(i) \), where \( v'(i) \) is the optimal value function defined in section 5.1.1. The optimal action at \( i \) achieving \( v' \) in (5.1.7) is given by
\[ \zeta'(i) = \begin{cases} 0 & \text{if } r(i) - \gamma + \alpha \sum_{j \in S} P(i, j)v'(i) \leq 0 \\ 1 & \text{otherwise} \end{cases} \] (5.2.5)

Particularly, \( v'(i) > 0 \) if (and only if) \( \zeta'(i) = 1 \). The optimal policy is therefore given by \( \pi^*_i = \zeta'(S_i) \). It is easy to check that \( \pi^*_i = 0 \) implies that \( \pi^*_{i'} = 0 \) for \( i' > i \). Hence, \( \pi^* \) obtained from Bellman’s optimality criterion is indeed a stopping time which is optimal. As will be seen, this result plays a crucial role in defining the Gittins index policy in the next section.

We end this section by giving an interesting application of Theorem 5.2.1.

\(^{6}\)I did not prove this, but it seems similar to the property of the retirement policies for the one-armed bandit, see [LS18].
Example 5.2.2 (Secretary Problem). Suppose that there are \( n \) applicants looking for the same job and they come in a random order for an interview. Denote the true score of the applicant coming at time \( t \) by \( Z_t \). For simplicity, we assume that \( Z_1, \ldots, Z_n \) are random variables independently drawn from the uniform distribution on \([0, 1]\), so that for any permutation \( \sigma \) on \([n]\), \( \mathbb{P}(Z_{\sigma(1)} > \cdots > Z_{\sigma(n)}) = 1/n! \). Denote the true rank of the \( t \)-th applicant as \( r(Z_t) \). The interviewer wants to maximize the chance of finding the best applicant for the job. However, in each round of interview, he/she needs to decide whether to accept the applicant or not. The applicants who have been turned away cannot be recalled (This may not be true in practice since there is always a waitlist procedure). The only information that the interviewer has in round \( t \) is the relative rank of the current applicant among those who have been interviewed. That is, \( Q_t = \sum_{s \in [t]} I_{Z_s \geq Z_t} \). Denote the natural filtration by \( \mathbb{F}_t = (Q_1, \ldots, Q_t) \). The goal of the Secretary Problem is to find a stopping time \( \tau^* \leq n \) (adapted to \( \{\mathbb{F}_t\}_{t \in [n]} \)) such that

\[
\mathbb{P}(r(Z_{\tau^*}) = 1) = \mathbb{E} I_{r(Z_{\tau^*}) = 1} = \sup_{\tau} \mathbb{E} I_{r(Z_{\tau}) = 1}.
\]

Since \( I_{r(Z_{\tau}) = 1} \) is not \( \mathbb{F}_t \)-measurable, we therefore consider its conditional expectation on \( \mathbb{F}_t \) so we define \( X_t = \mathbb{E}[I_{r(Z_{\tau}) = 1} | \mathbb{F}_t] \). Under our assumption, it is easy to verify that

\[
X_t = \frac{t}{n} \cdot I_{Q_t = 1}.
\]

So the Snell’s envelop \( \{Y_t\}_{t \in \mathbb{N}} \) is given by

\[
Y_t = \max \left\{ \frac{t}{n} \cdot I_{Q_t = 1}, \mathbb{E}[Y_{t+1} | \mathbb{F}_t] \right\},
\]

with \( Y_n = X_n = I_{Q_n = 1} \). By an induction argument,

\[
Y_t = \begin{cases} 
\max \left\{ \frac{t}{n} \cdot \frac{1}{n} \sum_{s=0}^{n-t-1} \frac{1}{t+s} \right\} & \text{if } Q_t = 1 \\
\frac{1}{n} \sum_{s=0}^{n-t-1} \frac{1}{t+s} & \text{otherwise.}
\end{cases}
\]

So there is a critical point \( \tau^* \) such that \( \frac{\tau^*}{n} \geq \frac{1}{n} \sum_{t=0}^{n-\tau^*-1} \frac{1}{t+\tau^*} \), which has an asymptotic solution \( \tau^* \geq \lfloor n/e \rfloor \). Therefore, the optimal policy is to directly reject the first \( 1/e \) proportion of the applicants, and from then on select the first applicant that is relatively best until the last round.

5.3 Gittins Index and Bayesian Optimality

Let us return to the \( k \)-armed Bayesian bandits. For \( i \in [k] \), denote by \( Q_i \) the prior distribution of \( \mu_i \). We assume that \( Q_1, \ldots, Q_k \) are independent so that \( Q(\mu_1, \ldots, \mu_k) = \prod_{i \in [k]} Q_i(\mu_i) \). In this section, using the tools developed before, we provide a complete characterization of Bayesian optimal policies when the discounted cumulative reward is considered. This result is known as Gittins Index Theorem and probably one of the
most important discoveries in bandit learning. To lay out the discussion, it is helpful to look at same problem from a Markovian perspective.

Most policies in Bayesian bandits are based on calculating the posterior distribution of the parameters. Under our assumption, this is equivalent to computing the posterior for each arm separately. Whenever an arm is pulled, the new data is fed to update its posterior using Bayes’ formula. Posteriors of other arms remain the same. Assume that the class of posterior distributions form a parametric family. For example, if we use beta prior $B(a, \beta)$ for each arm in a Bernoulli bandit, the posteriors of the arms are still beta distributions (conjugacy) with parameters determined by the respective sufficient statistics. Note that in this particular case, the sufficient statistics are given by the sum of rewards for each arm, which evolve in a Markovian way. This allows us to view each arm as a Markov chain with state space given by the posterior distributions (or their associated parameters). The action of the player in each round determines which Markov chain will move to the next state. The reward collected at each step depends on the current state of the Markov chain only. Therefore, the Bayesian bandit problem can be reformulated as a bandit whose arms are given by independent Markov Reward Processes.

Let us give an abstract set-up of what we described above. For simplicity consider $k$ Markov chains $S_t^{(1)}, \ldots, S_t^{(k)}$ with the same state space $S$ and transition kernel $P$. Let $r : S \to \mathbb{R}$ be a reward function. At time $t$, the player picks an arm using a policy $\pi$, based on which arms evolve as

$$S_{t+1}^{(i)} \sim \begin{cases} P(S_{t}^{(i)}, \cdot) & \text{if } \pi_t = i \\ \delta_{S_t^{(i)}}(\cdot) & \text{otherwise,} \end{cases}$$

where $\delta_s(\cdot)$ is the Dirac mass at $s$. The resulting reward is given by $r(S_t^{(\pi_t)})$. Since $r$ is completely known, one could hope to achieve zero regret. Thus we set the objective function as the gain by the player. Particularly, for a given policy $\pi$, we consider the infinite-horizon $\alpha$-discounted cumulative reward:

$$R_\pi = \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} r(S_t^{(\pi_t)}) \right].$$

Our goal is seek a policy $\pi^*$ such that

$$R_{\pi^*} = \sup_{\pi} R_\pi. \quad (5.3.1)$$

Again, it is not clear by the question itself if such a $\pi^*$ exists. Surprisingly, John C. Gittins showed in his seminal papers [Git74] and [Git79] that such $\pi^*$ does exist and is given by an index policy. One can simply take index policies as first calculating some statistics of each arm independently and comparing the results to make a decision. Many of the algorithms such as UCB fall within this realm. However, we should point out that not all Bayesian optimal policies are index policies, as they depend heavily on the structure of the bandit as well as the objective function.
For \( i \in S \), define the Gittins index of \( i \) as
\[
g(i) = \inf \left\{ \gamma : \nu^*(i) = 0 \right\},
\]
where \( \nu^*(i) \) is the value function used in section 5.2 when discussing optimal stopping on Markov chains. The Gittins index policy is to choose the arm with the highest Gittins index in each round:
\[
\pi^*_t = \arg\max_{i \in [k]} g(X^{(i)}_t).
\] (5.3.2)

The next theorem tells us that \( \pi^* \) is optimal in the sense of (5.3.1).

**Theorem 5.3.1** (Gittins Index Theorem). Let \( \pi^* \) the Gittins Index policy defined in (5.3.2). Under the previous assumptions on the Markov structure on the arms, \( \pi^* \) satisfies (5.3.1).

We give an intuitive proof for the theorem based on [Web92].

**Proof.** For the moment let \( i \) be the only arm that the player can choose in each round. We come up with an explicit way to evaluate the best discounted cumulative gain using the knowledge in section 5.2. Assume a fixed fee \( \gamma \) needs to be paid before the next pull, and we call \( \gamma \) the prevailing charge. In this case, the problem reduces to the Markov optimal stopping problem in section 5.2, with the optimal policy given by (5.2.5). It is easy to check that (5.2.5) is the same as the stopping time \( \tau = \{ t : g(X_t) < \gamma \} \).

However, prevailing charge is an artificially introduced quantity. To make the game run in infinite time, we use the following trick. Whenever \( \tau \) is arriving, we lower \( \gamma \) to the Gittins index of the current state so the game keeps going until the next stop, for which the same procedure is applied. This modified process remains piecewise optimal with zero gain and will last forever.

Now consider \( k \) arms together. Suppose that each arm starts with the prevailing charge of their initial states. The prevailing charge for \( i \) will be updated as in the single-armed case if \( \pi_t = i \) and \( g(X_t^{(i)}) \) is less than the current prevailing charge used for \( i \). Such modification ensures that the best policy on an arm between two neighboring prevailing charges results in a 0 expected reward. Therefore,
\[
R_{\pi} \leq \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} g(X_t^{(\pi_t)}) \right],
\]
where equality holds if the \( \pi \) keeps playing the same arm until its prevailing charge changes. It is easy to check that the Gittins index policy \( \pi^* \) satisfies this, so
\[
R_{\pi^*} = \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} g(X_t^{(\pi^*_t)}) \right].
\]

The proof is finished by using the Hardy-Littlewood inequality (identical ordering of two non-negative sequences maximizes their inner product):
\[
\sup_{\pi} \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} g(X_t^{(\pi_t)}) \right] = \mathbb{E} \left[ \sum_{t \in \mathbb{N}} \alpha^{t-1} g(X_t^{(\pi^*_t)}) \right].
\]
\( \square \)
5.3. **GITTINS INDEX AND BAYESIAN OPTIMALITY**

**Remark 5.3.2.** Even though Gittins index policy is optimal, it is very sensitive to the assumptions made for its derivation. For example, the discounting needs to be geometric.

**Remark 5.3.3.** Note that for $i \in S$, its Gittins index $g(i)$ satisfies $G(g(i)) = 0$, where $G(\gamma) = r(i) - \gamma + \alpha \sum_{j \in S} P(i,j) v^*(j)$. It can be checked that $G$ is a decreasing function. So one may use the bisection method to find the intervals containing $g(i)$, see [BF90].

An alternative computational algorithm is due to Pravin Varaiya, see [VWB85]. The idea lies in that observing that if for $i \in S$, define its associated stopping time $T_i = \min\{t : g(X_t) < g(i)\}$, then $g(i)$ satisfies

$$g(i) = \frac{\mathbb{E}\left[\sum_{t=1}^{T_i-1} \alpha^{t-1} r(X_t)\right]}{\mathbb{E}\left[\sum_{t=1}^{T_i-1} \alpha^{t-1}\right]}.$$

Treating the stopping time as killing, one can modify the transition matrix to rewrite the formula in a closed form. Since $i^{(1)} := \arg\max r(i) = \arg\max g(i)$ and $T_{i^{(1)}} \equiv 2$, one can use downward iteration to find the Gittins indices in a decreasing manner.
Bibliography


