CSE599i: Online and Adaptive Machine Learning **Winter 2018**

Lecture 4: Stochastic Multi-Armed Bandits, Pure Exploration

Lecturer: Kevin Jamieson *Scribes: William Agnew, Tomer Kaftan*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

Introduction

Previously, we looked at Regret Minimization for the multi-armed bandit problem, where the goal is to minimize the expected regret of a player. In contrast, we now consider the pure exploration, best-arm identification problem where the goal is to identify the best arm in as few samples as possible. We will focus on the fixed-confidence setting where the goal is to identify the best arm with high probability $(1 - \delta)$ in as few samples as possible. Algorithms for Pure Exploration in the fixed-confidence setting have the following format: A player is given n arms. At each timestep t, the player selects an arm to pull (I_t), and they observe some reward $(X_{I_t,t})$ for that sample. At any timestep, once the player is confident that they have identified the best arm, they may decide to stop.

Algorithm 1 Pure Exploration

```
procedure PURE EXPLORATION($\{1, 2, \ldots, n\}$)          ▷ Arms 1 through $n$
    for $1 \leq t \leq \infty$ do
        Choose $I_t \in \{1, 2, \ldots, n\}$               ▷ Player selects an arm
        Observe $X_{I_t,t}$       ▷ Player observes a reward for arm $I_t$ (w/ $\mathbb{E} = \mu_{I_t}$)
        Player may decide to STOP       ▷ Player stops if they have identified the best arm
    end for
    Return best arm
end procedure
```

Goal: *w.p.* $\geq 1 - \delta$, the player

1. Identifies $\arg\max_i \mu_i$
2. STOPs before M samples, where M is as small as possible

Notations:

- I_t: The arm chosen at round t
- $X_{i,t}$: reward observed for arm i at round t. For simplicity, we assume $X_{i,t}$ is bounded within $[0, 1]$
- μ_i: The expected reward for arm i
- $\mu^* = \max\limits_j \mu_j$
- $\Delta_i = \mu^* - \mu_i$

Successive Elimination

We now describe an algorithm for the fixed-confidence Pure Exploration setting known as Successive Elimination. This algorithm was first proposed by Even-dar et al. (2006) [1]. The Successive Elimination algorithm proceeds as follows: The player maintains a set of active arms S. At every round, the player first samples from the rewards of every arm in the active set. The player then removes all arms in the active set with estimated rewards that are outside an **any-time confidence interval** around the biggest estimated reward in the active set. When the active set has only one arm, the player identifies this arm with high probability as the best arm.

Algorithm 2 Successive Elimination

```
procedure SUCCESSIVE ELIMINATION($\{1, 2, \dots, n\}, \delta$)        ▷ $n$ arms, parameter $\delta$ in (0, 1)
    $S \leftarrow \{1, \cdots, n\}$                                  ▷ Initialize the set of active arms
    for $1 \leq t \leq \infty$ do
        Pull all arms in S
        $S \leftarrow S - \{i \in S : \exists j \in S : \hat{\mu}_{j,t} - U(t, \delta/n) \geq \hat{\mu}_{i,t} + U(t, \delta/n)\}$        ▷ Drop bad arms
        STOP when $|S| = 1$
    end for
    return $S$
end procedure
```

Additional Notation:

- S: The active set of arms
- $\hat{\mu}_{i,t} = \frac{1}{t}\sum_{j=1}^{t} X_{i,j}$: Estimated mean reward for Arm i after t pulls
- $U(t, \delta)$ may be any function that satisfies the **any-time confidence bound**, such that for any arm i:

$$\mathbb{P}\left(\bigcup_{t=1}^{\infty}\{|\hat{\mu}_{i,t} - \mu_i| > U(t, \delta)\}\right) \leq \delta$$

Importantly, with high probability these bounds hold for all time, rather than independently holding with high probability at each timestep individually.

Law of Iterated Logarithms and Any-time Confidence Bounds

When proving the correctness of best-arm identification algorithms, we rely on confidence intervals around estimated means that must hold with high probability **for all time**. The Law of Iterated Logaritms [2] and any-time confidence bounds give us those confidence intervals. More specifically, the Law of Iterated Logarithms bounds how much the observed average of random variable deviates from its expected value with high probability as the number of observations goes to infinity.

Theorem 1. *(Law of Iterated Logarithms) If $Z_t \sim \mathcal{N}(\mu, 1)$ for t=1,2,..., then*

$$\limsup_{t\to\infty} \frac{\sum_{s=1}^{t}(Z_s - \mu)}{\sqrt{2t\log\log t}} = 1 \textit{ with probability } 1$$

However, the Law of Iterated Logarithms as-is only holds in the limit as $t \to \infty$. Successive Elimination requires any-time confidence bounds that hold for finite numbers of samples. One such example interval that satisfies an any-time confidence bound is as follows:

Theorem 2. *(Anytime Confidence Interval) If X_t is within $[0, 1]$ and $0 < \delta < 1$, then*

$$P\left(\bigcup_{t=1}^{\infty}\left\{\left|\frac{1}{t}\sum_{s=1}^{t} X_s - \mathbb{E}[X_s]\right| \geq \sqrt{\frac{\log(4t^2/\delta)}{2t}}\right\}\right) \leq \delta$$

Proof.

$$P\left(\bigcup_{t=1}^{\infty}\left\{\left|\frac{1}{t}\sum_{s=1}^{t}X_s-\mathbb{E}[X_s]\right|\geq\sqrt{\frac{\log(4t^2/\delta)}{2t}}\right\}\right)\overset{\texttt{union bound}}{\leq}\sum_{t=1}^{\infty}P\left(\left|\frac{1}{t}\sum_{s=1}^{t}X_s-\mathbb{E}[X_s]\right|\geq\sqrt{\frac{\log(4t^2/\delta)}{2t}}\right)$$

Using Hoeffding's inequality (via the knowledge that rewards are bounded within $[0,1]$), we get:

$$\begin{aligned}P\left(\left|\frac{1}{t}\sum_{s=1}^{t}X_s-\mathbb{E}[X_s]\right|\geq\sqrt{\frac{\log(4t^2/\delta)}{2t}}\right)&\leq 2e^{-\frac{2\left(\sqrt{\frac{\log(4t^2/\delta)}{2t}}\right)^2t^2}{t}}\\&\leq 2e^{-\log(4t^2/\delta)}\\&\leq 2\cdot\frac{\delta}{4t^2}\\&\leq\frac{\delta}{2t^2}\end{aligned}$$

Plugging this back into the initial summation, we find:

$$\begin{aligned}\sum_{t=1}^{\infty}P\left(\left|\frac{1}{t}\sum_{s=1}^{t}X_s-\mathbb{E}[X_s]\right|\geq\sqrt{\frac{\log(4t^2/\delta)}{2t}}\right)&\leq\sum_{t=1}^{\infty}\frac{\delta}{2t^2}\\&\leq\frac{\delta}{2}\sum_{t=1}^{\infty}\frac{1}{t^2}\\&\leq\frac{\delta}{2}\cdot 2\\&\leq\delta\end{aligned}$$

□

This bound is a baseline that could be used in Successive Elimination by setting:

$$U(t,\delta)=\sqrt{\frac{\log(4t^2/\delta)}{2t}}$$

The current state of the art for any-time confidence bounds (Kaufmann et al. 2016 [3]) is:

$$P\left(\bigcup_{t=1}^{\infty}\left\{\left|\frac{1}{t}\sum_{s=1}^{t}X_s-\mathbb{E}[X_s]\right|\geq\sqrt{\frac{2\log(1/\delta)+6\log\log(1/\delta)+3\log\log(et)}{t}}\right\}\right)\leq\delta \qquad (\star)$$

These any-time confidence bounds are the correct way to avoid "p-hacking" when evaluating the statistical significance of results in settings where confidence interval tests are sequentially being applied to incoming data, and users are *continously monitoring* the results of the significance tests to decide when a result has been found. For example, many A/B testing frameworks can continously monitor p-values that specify if option A is better than B (by applying a statistical test on the data after every new user that interacts with the framework). Johari et al. (2015) [4] show that if an A/B testing framework does not use any-time bounds, continuous monitoring will eventually detect statistical significance by mistake after enough time has passed. By using statistical significance tests that rely on any-time confidence intervals that hold with high probability for all time, users of the A/B testing framework could safely monitor continuously without accidentally discovering erroneous results.

Proof of Successive Elimination

We will now prove that $w.p. \geq 1-\delta$ Successive Elimination identifies the best arm in $\mathcal{O}\left(\sum_{i \neq i^*}^{n} \Delta_i^{-2} \log\left(n \log(\Delta_i^{-2})\right)\right)$ samples. We can prove this by showing that $w.p \geq 1-\delta$:

1. The arm with the biggest expected reward μ^* will always remain in the active set S
2. All non-optimal arms i with reward $\mu_i \leq \mu^*$ will be dropped from S after $\mathcal{O}\left(\sum_{i \neq i^*}^{n} \Delta_i^{-2} \log\left(n \log(\Delta_i^{-2})\right)\right)$ samples

Lemma 1. *Let event $\mathcal{E}$ be the case that for any arm at any time t, the estimated reward $\hat{\mu}_{i,t}$ is outside of the confidence bound around the true mean μ_i.*

$$\mathcal{E} = \bigcup_{i=1}^{n} \bigcup_{t=1}^{\infty} \{|\hat{\mu}_{i,t} - \mu_i| > U(t, \delta/n)\}$$

The event holds with $\mathbb{P}(\mathcal{E}) \leq \delta$

Proof. By the Union bound,

$$\mathbb{P}(\mathcal{E}) \leq \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t=1}^{\infty} \{|\hat{\mu}_{i,t} - \mu_i| > U(t, \delta/n)\}\right)$$

And, by the any-time confidence bound:

$$\begin{aligned}\sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t=1}^{\infty} \{|\hat{\mu}_{i,t} - \mu_i| > U(t, \delta/n)\}\right) &\leq \sum_{i=1}^{n} \frac{\delta}{n} \\ &\leq n\frac{\delta}{n} \\ &\leq \delta\end{aligned}$$

□

Theorem 3. *With probability $\geq 1-\delta$, the best arm remains in the active set S until termination.*

Proof. Arm i will only be dropped from set S if $\exists j$ such that

$$\hat{\mu}_{j,t} - U(t, \delta/n) \geq \hat{\mu}_{i,t} + U(t, \delta/n) \tag{1}$$

Additionally, in the case where $\mathcal{E}^c$ holds, we know that the estimated rewards are always within a confidence bound around the true mean, and so

$$\mu_j + U(t, \delta/n) \geq \hat{\mu}_{j,t} \quad \text{and} \quad \hat{\mu}_{i,t} \geq \mu_i - U(t, \delta/n)$$

Plugging the inequalities above into Equation (1), we can see that if $\mathcal{E}^c$ holds, dropping an arm implies that:

$$\mu_j + U(t, \delta/n) - U(t, \delta/n) \geq \mu_i - U(t, \delta/n) + U(t, \delta/n)$$

Which in turn can only be true if

$$\mu_j \geq \mu_i$$

This will never occur for the best arm, as no other arm will have a higher expected reward. So, the best arm will always remain in the active set if $\mathcal{E}^c$ holds. Additionally, by Lemma 1, $\mathbb{P}(\mathcal{E}^c) \geq 1-\delta$. Therefore, with probability $\geq 1-\delta$, the arm with the highest expected reward μ^* is guaranteed to remain within the active set S until termination. □

Next we must provide an upper bound for how long Successive Elimination takes to terminate (i.e., the sampling complexity of Successive Elimination).

Theorem 4. *In the case where Successive Elimination successfully identifies the best arm, it will terminate after* $\mathcal{O}\left(\sum_{i\neq i^*}^{n} \Delta_i^{-2} \log\left(n \log(\Delta_i^{-2})\right)\right)$ *samples.*

Proof. By the removal rules of Successive Elimination, one of the events that will remove arm i from S is if:

$$\hat{\mu}_t^* - U(t, \delta/n) \geq \hat{\mu}_{i,t} + U(t, \delta/n) \tag{2}$$

(Where $\hat{\mu}_t^*$ is the estimated reward of the arm with the largest expected reward, μ^*)

Additionally, in the case where $\mathcal{E}^c$ holds, we know that the estimated rewards are always within a confidence bound around the true mean, and so

$$\hat{\mu}_t^* \geq \mu^* - U(t, \delta/n) \quad \text{and} \quad \hat{\mu}_{i,t} \leq \mu_{i,t} + U(t, \delta/n)$$

Therefore, in the case of event $\mathcal{E}^c$, the event from Equation 2 is guaranteed to occur (and arm i will be dropped) as long as:

$$\mu^* - 2U(t, \delta/n) \geq \mu_i + 2U(t, \delta/n)$$

Which we can rearrange as

$$\Delta_i \geq 4U(t, \delta/n)$$

Where $\Delta_i = \mu^* - \mu_i$

By solving for the minimum value of t for which $\Delta_i \geq 4U(t, \delta/n)$, we can upper bound how long it takes for arm i to be removed from S. It can be shown that the miniminum T_i such that $\Delta_i \geq 4U(T_i, \delta/n)$ satisfies:

$$T_i \leq c\Delta_i^{-2} \log\left(\frac{n \log(\Delta_i^{-2})}{\delta}\right)$$

(Using the confidence bound from ($\star$), where c is a constant)

Each non-optimal arm i will have been removed from S after it has been sampled at most T_i times. So, summing up the upper bounds on the samples of each arm, we find that all non-optimal arms will have been removed from S within

$$\mathcal{O}\left(\sum_{i\neq i^*}^{n} \Delta_i^{-2} \log\left(n \log(\Delta_i^{-2})\right)\right)$$

samples. At this point Successive Elimination will terminate. □

In 2004, Mannor et al. [5] showed that the lower bound on the sampling complexity of pure exploration algorithms is $\mathcal{O}(\sum_{i\neq i^*} \Delta_i^{-2})$. The upper bound on the sample complexity of Successive Elimination misses this lower bound by a factor of $\log\left(n \log(\Delta_i^{-2})\right)$. Improvements on Successive Elimination have better sampling complexity, and more closely approach the known lower bound.

Improvements on Successive Elimination

We now describe three algorithms which improve on the sample complexity of successive elimination. All have the same high level idea of repeatedly sampling some arms, estimating an upper bound on their reward, and eliminating arms that cannot possibly be the best.

Exponential-Gap Elimination [6]

Exponential-Gap Elimination eliminates low reward arms over multiple rounds, only keeping arms close to an arm that is approximately the best. As more rounds are played, arms must be exponentially closer to the estimated best arm to be kept for the next round.

Algorithm 3 Exponential-Gap Elimination

```
 1: procedure Exponential-Gap Elimination($S, \delta$)        ▷ $S$, set of $n$ arms, parameter $\delta$ in (0, 1)
 2:     $S_1 \leftarrow [n], k \leftarrow 1$
 3:     while $|S_k| > 1$ do
 4:         $\epsilon_k \leftarrow 2^{-k}/4$, $\delta_k \leftarrow \delta/(50k^3)$
 5:         Sample each arm $i \in S_k$ $t_k \leftarrow (2/\epsilon_k^2)\log(2/\delta_k)$ times
 6:         Let $\hat{p}_i^k$ be the average reward of arm $i$ in round $k$
 7:         $i_k \leftarrow$Median Elimination($S_k, \epsilon_k/2, \delta_k$) and $\hat{p}_*^k \leftarrow \hat{p}_*^{i_k}$
 8:         $S_{k+1} \leftarrow S_k \setminus \left\{i \in S_k : \hat{p}_i^k < \hat{p}_*^k - \epsilon_k\right\}$
 9:         $k \leftarrow k + 1$
10:     end while
11: return $S_k$                                                ▷ $S_k$ has only one arm
12: end procedure
```

Given a set S of n bandit arms and $\epsilon, \delta > 0$, the median elimination algorithm [1] is a (ϵ, δ)-PAC algorithm outputs an arm within ϵ of optimal with probability at least $1 - \delta$ with $\mathcal{O}((n \ \epsilon^2)\log(1 \ \delta))$ samples. Exponential-Gap Elimination uses median elimination as a sub-routine to efficiently provide a rough estimate of the maximum expected reward over the arms exponential-gap elimination has not yet eliminate. This rough estimate is then used to eliminate even more arms which cannot possibly be the best in step 5. While the maximum expected reward could be estimated from the past pulls of arms, using the median elimination sub-routine gives the same result in fewer additional samples, making the algorithm more efficient.

Algorithm 4 Median Elimination

```
 1: procedure Median Elimination($S, \epsilon, \delta$)   ▷ $S$, set of $n$ arms, error $\epsilon$ in (0, 1), failure probability $\delta$ in
    (0, 1)
 2:     $\epsilon_1 \leftarrow \epsilon/4$, $\delta_1 \leftarrow \delta$, $k \leftarrow 1$
 3:     repeat
 4:         Sample each arm $i \in S_k$ $1/(\epsilon_k/2)^2\log(3/\delta_k)$ times
 5:         Let $\hat{p}_i^k$ be the average reward of arm $i$ in round $k$
 6:         Let $m_k$ be the median $\hat{p}_i^k$ over all $i \in S_k$
 7:         $S_{k+1} \leftarrow S_k \setminus \{i : \hat{p}_i^r < m_k\}$
 8:         $\epsilon_{k+1} \leftarrow \frac{3}{4}\epsilon_k$, $\delta_{k+1} \leftarrow \delta_k/2$, $k \leftarrow k + 1$
 9:     until $|S_k| = 1$
10: return $S_k$
11: end procedure
```

Exponential-Gap Elimination has a sample complexity $\mathcal{O}(\sum_{i=2}^{n}\Delta_i^{-2}\log(\frac{1}{\delta}\log\frac{1}{\Delta_i}))$, assuming that i_1 is the best arm without loss of generality.

lil'UCB [7]

lil'UCB is the current state of the art multiarmed bandit algorithm both in theory and practice.

Let

$$\mathrm{H}_{\mathbf{1}} = \sum_{i \neq i^*} \Delta_i^{-2} \text{and } \mathrm{H}_{\mathbf{3}} = \sum_{i \neq i^*} \frac{\log\log_+(1/\Delta_i^2)}{\Delta_i^2}$$

Algorithm 5 lil'UCB

```
procedure LIL'UCB($S, \delta, \epsilon, \lambda, \beta$)    ▷ $S$, set of $n$ arms, parameter $\delta$ in (0, 1), parameters $\epsilon, \lambda, \beta > 0$
    $T_i(r) \leftarrow 1$ for each arm $i$
    Sample each of the $n$ arms once
    while $T_i(r) < 1 + \lambda \sum_{i \neq j} T_j(r)$ for all $i$ do
        Sample arm
            $I_r = \underset{i \in \{1,\ldots,n\}}{\arg\max} \left\{ \hat{\mu}_{i,T_i(r)} + (1+\beta)U(t,\delta) \right\}$
        $T_i(r+1) \leftarrow T_i(r) + 1$ if $I_r = i$, else $T_i(r+1) \leftarrow T_i(r)$
    end while
return $\arg\max_{i \in \{\mathbf{1},\ldots,n\}} T_i(r)$
end procedure
```

Let $\beta \in (0, 3]$. Then there exists an absolute constant $\lambda > 0$ such that with probability at least $1 - \delta$ lil'UCB terminates after at most $c_1 \mathbf{H}_1 \log(1/\delta) + c_3 \mathbf{H}_3$ samples and outputs the best arm. Lil'UCB has many similarities to UCB1. UCB1 is an algorithm for optimal play, not best arm selection, but lil'UCB easily converts this into outputting the best arm by outputting the most played arm, which will be the best arm afer an appropriate number of plays. In addition, lil'UCB introduces and optimizes several parameters to the UCB to achieve a tigher bound and better performance. Finally, lil'UCB uses an anytime confidence bound that holds for all future samples w.hp., while UCB1 uses a normal confidence bound that does not hold for all future samples w.h.p.

LUCB++ [8]

LUCB++ finds the top-K best arms, rather than just the best arm. Intuitively, finding the top-k arms requires finding the k-th and $k+1$-th best arms; that is, one must be confident that the worst arm in the top-k set is better than the best arm in the set of remaining arms. LUCB++ follows this intuition: it procedes by creating and refining estimates of the expected rewards of the k-th best and $k+1$-th best arms, terminating once it has estimated that the worst arm in the top k set is better than the best arm in the set of remaining arms with the desired confidence.

Algorithm 6 LUCB++

```
procedure LUCB++($S, \delta, t$)    ▷ $S$, set of $n$ arms, parameter $\delta$ in (0, 1), top-$k$ arms to find
    Sample all arms once
    for rounds $t = n+1, n+2, \ldots$ do
        TOP$_k \leftarrow \arg\max_{S \subset [n]:|S|=t} \sum_{i \in S} \hat{\mu}_{a,T_a(k)}$
        if $\min_{a \in TOP_k} \hat{\mu}_{a,T_a(k)} - U(N_a(k), \frac{\delta}{2(n-k)}) > \max_{a \in [n]-TOP_k} \hat{\mu}_{a,T_a(k)} - U(N_a(k), \frac{\delta}{2k})$ then return TOP$_k$
        else
            $h_t = \hat{\mu}_{a,T_a(t)} - U(N_a(t), \frac{\delta}{2(n-k)})$
            $l_t = \max_{a \in [n] \setminus TOP_t} \hat{\mu}_{a,T_a(t)} + U(N_a(t), \frac{\delta}{2k})$
            Sample $h_t$ and $l_t$
        end if
    end for
end procedure
```

This algorithm has a sample complexity that is $\mathcal{O}(\sum_{i=1}^{k} \Delta_i^{-2} \log(\frac{(n-k)\log(\Delta_i^{-2})}{\delta} + \sum_{j=k+1}^{n} \Delta_j^{-2} \log(\frac{k \log(\Delta_{})}{}))$.

Fixed Budget Setting

All of the previous algorithms attempt to find the best arm with confidence δ and the least samples. We now consider an alternative setting where we try to find the best arm with the highest confidence given a fixed budget of T samples. This setting is applicable in many real-world scenarios where an agent must make descisions in real time, and therefore can only conduct a limited number of simulations or experiments, or where the number of samples that may be taken is otherwise limited by factors like cost. In this setting, for any bandit algorithm there is some problem such that the bandit algorithm will fail to output the best arm with probability at least

$$\exp\left(-\frac{T}{\log(n)\sum\limits_{i\neq i^*}\Delta_i^{-2}}\right)$$

where T is the sample budget [9]. Or, put in terms of fixed confidence, for any bandit algorithm there is some problem that requires at least

$$T = \log(\frac{n}{\delta})\sum_{i\neq i^*}\Delta_i^{-2}$$

samples.

Sequential Halving [6]

Sequential Halving, as its name suggests, samples arms over multiple rounds, eliminating the worst half of arms each round. In each round, every remaining arm is sampled the same number of times. When only a few rounds have been played, arms have been sampled little, and thus the estimates of their values have higher varience. However, only half of the arms are eliminated each round, so failure in the first few rounds requires many arm value estimates to be incorrectly higher than the estimate of the best arm, which is unlikely. As more rounds are played, the top arms are sampled more, giving more confident estimates of their value.

Algorithm 7 Sequential Halving

```
procedure SEQUENTIAL HALVING($S, T$)                ▷ $S$, set of $n$ arms, $T$, sample budget
    for $k = 0$ to $\lceil \log_2 n \rceil$ do
        Sample each arm $i \in S_k$ $t_k = \left\lfloor \frac{T}{|S_k| \lceil \log_2 n \rceil} \right\rfloor$ times
        Let $\hat{p}_i^k$ be the average reward for arm $i$
        Let $S_{k+1}$ be the set of $\lceil S_k/2 \rceil$ arms in $S_k$ with the highest observed average reward
    end forreturn $S_{\lceil \log_2 n \rceil}$
end procedure
```

This algorithm needs $\mathcal{O}(\max_{i\neq 1} \frac{i}{\Delta_i^2} \log n \log \frac{\log n}{\delta})$ samples to identify the best arm with probability at least $1-\delta$, where arm one is assumed to be the best without loss of generality. Note that $\max_i \frac{i}{\Delta_i^2} \leq \sum_i \Delta_i^{-2}$ since the sequence $(\Delta_1^{-2}, \Delta_2^{-2}, \ldots, \Delta_n^{-2})$ is decreasing, so for any Δ_i^{-2}, there are at least i other elements in the sequence at least as large as Δ_i^{-2}.

References

[1] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

[2] Philip Hartman and Aurel Wintner. On the law of the iterated logarithm. *American Journal of Mathematics*, 63(1):169–176, 1941.

[3] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

[4] Ramesh Johari, Leo Pekelis, and David J Walsh. Always valid inference: Bringing sequential analysis to a/b testing. *arXiv preprint arXiv:1512.04922*, 2015.

[5] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

[6] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1238–1246, 2013.

[7] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

[8] Max Simchowitz, Kevin Jamieson, and Benjamin Recht. The simulator: Understanding adaptive sampling in the moderate-confidence regime. *arXiv preprint arXiv:1702.05186*, 2017.

[9] Alexandra Carpentier and Andrea Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Conference on Learning Theory*, pages 590–604, 2016.