修士論文

確率的マルチアームドバンディット問題に対する最適アーム探索アルゴリズムの提案

A Best-arm Identification Algorithm for Stochastic Multi-armed Bandit

2015 年 2 月 5 日

指導教員　若原 恭 教授

東京大学大学院
工学系研究科 電気系工学専攻

37-136481　福勢 晋

概要

マルチアームドバンディット問題 (Multi-Armed Bandit Problem) は、プルすると確率的に報酬が得られるスロットマシン (アーム) が複数台ある状況で、どのようにスロットマシンを選んでいけば累積報酬を最大化できるかという問題であり、広告表示最適化や臨床試験に応用できることに加えて、探索 (Exploration) と活用 (Exploitation) のトレードオフを内在している問題の中でも比較的単純なので、理論的な研究の価値も高い。従来、マルチアームドバンディットではリグレットと呼ばれる、常に最適アームをプルした場合に得られる累積報酬と実際に得られた累積報酬の差の期待値に注目して、これの最小化 (Regret Minimization) を目標に、様々なアルゴリズムが提案されてきたが、近年、報酬期待値が最も高いアームをいかに少ないサンプル数で見つけるかという、最適アーム探索 (Best-arm Identification) 問題が注目されている。臨床試験や心理学実験に応用した場合、被験者数を少なくしたり実験期間を短くする効果が期待されるので、応用上も重要な問題である。最適アーム探索は、sequential test(逐次検定) と繋がりが深く、有意差を検出するために必要なサンプル数が少ない sequential test を用いることで、最適アーム探索の必要サンプル数を下げることが出来る。そこで本研究では、従来よりも少ないサンプル数で sequential test を行う方法を 2 つ提案する。その中の 1 つは、LIL(Law of the Iterated Logarithm) から考えて理論的に最適であることを示す。提案した 2 つの sequential test を用いて、従来から知られている最適アーム探索アルゴリズムである successive elimination の改良を行う。また、sequential test を用いた新しい最適アーム探索アルゴリズムを提案する。そしてシミュレーションによって、これらのアルゴリズムが従来よりも少ないサンプル数で最適アームを見つけられることを示す。

Abstract

Recently best-arm identification in multi-armed bandit problem(MAB) has become intensively investigated. At each time, a player pulls an arm and gets a reward. The arm that has the largest mean reward is called optimal. The goal of this problem is to identify the optimal arm with the smallest number of pulls. This problem has various applications: clinical trials, psychological experiments, and so on. In previous works, it was revealed that sequential hypothesis testing is closely related with this problem and some best-arm identification algorithms that make use of sequential tests were proposed. However, sequential tests used by the algorithms are far from optimal. We propose two sequential tests of the mean of subgaussian random variables and show that one of the tests is optimal in the aspect of LIL(Law of the Iterated Logarithm). We improve the successive elimination algorithms using these tests and propose a new best-arm identification algorithm. We demonstrate that these algorithms have smaller sample-complexity than conventional best-arm identification algorithms by computer simulation.

目次

図目次

表目次

第1章
序論

1.1 研究背景

マルチアームドバンディット問題 (Multi-Armed Bandit Problem) は、プルすると確率的に報酬が得られるスロットマシン (アーム) が複数台ある状況で、どのようにスロットマシンを選んでいけば累積報酬を最大化できるかという問題であり、広告表示最適化や臨床試験に応用できることに加えて、探索 (Exploration) と活用 (Exploitation) のトレードオフを内在している問題の中でも比較的単純なので、理論的な研究の価値も高い。アームから得られる報酬のモデルとしては、アームごとに関連付けられた確率分布から i.i,d(independent and identically distributed) にサンプルされることを仮定した確率的マルチアームドバンディッド (stochastic multi-armed bandit) と、$[0, 1]$ の範囲内で任意の報酬が許される adversarial マルチアームドバンディッドという、大きく分けて 2 つのモデルがある。本研究では、確率的マルチアームドバンディッドを扱う。

従来、マルチアームドバンディットではリグレットと呼ばれる、常に最適アームをプルした場合に得られる累積報酬と実際に得られた累積報酬の差の期待値に注目して、これの最小化 (Regret Minimization) を目標に、様々なアルゴリズムが提案されてきたが、近年、報酬期待値が最も高いアームをいかに早く見つけるかという、最適アーム探索 (Best-arm Identification) 問題が研究されるようになった。古くは、文献 [1] のようにアームの報酬分布として正規分布を仮定した研究があるが、近年では報酬分布に subgaussian 分布や有界な分布を仮定した研究が急速に進んでいる [2], [3], [4]。subgaussian 分布や有界な分布は、報酬分布の形が未知で正規分布を仮定出来ない場合でも適用できる可能性があるので、応用が広い。最適アーム探索では、最適アームを見つけるために必要なサンプル数を sample-complexity と呼び、これを小さくすることが目標とされている。その中でも、fixed confidence という適当なエラー率 $\delta \in (0, 1]$ に対して $1 - \delta$ 以上の確率で最適アームを見つけるために必要なサンプル数を最小化する目標設定と、fixed horizen という適当な最大サンプル数 T に対して、T 回のプル後に最適アームを見つけている確率を最大化するという目標設定がある。本研究ではエラー率上限を保証出来るということが、臨床試験や心理学実験などの応用で重要であることを踏まえて fixed confidence を扱う。

fixed confidence に対応した最適アーム探索アルゴリズムは、successive elimination[2], ExpGap[3], lil'UCB[4] がある。これら 3 つの手法は元々最適アーム探索を目的として作られた手法だが、文献 [4] によって提案された LS(LIL Stopping) というものを使うと、任意のアルゴリズムを最適アーム探索に対応させることが出来る。文献 [4] によると、LUCB[5] と LS を組み合わせたときに実験的に小さい sample-complexity を達成できることが分かっている。LS は、subgaussian 確率変数の平均に関する sequential test(逐次検定) を元にして作られていて、sequential test の sample-complexity がアルゴリズムの性能に大きく影響する。調べた結果、文献 [4] で提案されている sequential test は、与えた第一種エラー率に対して実際のエラー率が 100 倍以上小さくなっ

ていることが分かった。実際のエラー率が高いほど sample-complexity は小さく出来るので、実際のエラー率は与えたエラー率になるべく近いほうが良い。実際のエラー率が与えたエラー率に近くなるような (タイトな)sequential test を行う研究は文献 [6] でされているが、LIL(Law of the Iterated Logairthm) から考えて最適では無いことが分かっている。

1.2　研究目的

従来よりも与えた第一種エラー率に対して実際のエラー率がより近くなるような sequential test を行う方法を 2 つ提案する。この sequential test が LIL の観点から考えて最適であることを示し、またシミュレーションによって実際の性能も従来より高いことを示す。2 つの sequential test を用いて successive elimination を改良して、LS1 elimination, LS2 elimination という 2 つのアルゴリズムを提案する。また、sequential test を用いた新しい最適アーム探索アルゴリズムを提案する。これらのアルゴリズムの sample-complexity はオーダー的に最適では無いが、実際の sample-complexity が従来よりも小さくなるということをシミュレーションによって示す。

第2章
マルチアームドバンディット問題とその背景

2.1 確率的マルチアームドバンディット問題の定式化

マルチアームドバンディット問題の基本的な説明を行う。複数台のスロットマシン (アームと呼ぶ) が並べられている状況を考える。プレイヤーは毎時刻、どれか一つのアームを選んでプル (サンプル) する。アームをプルすると、プレイヤーはアームからランダムに出力される報酬を得ることが出来る。このような状況で、プレイヤーはどのような基準でアームを選択していけば高い報酬を得ることが出来るのか、という問題がマルチアームドバンディット問題である (図 2.1)。

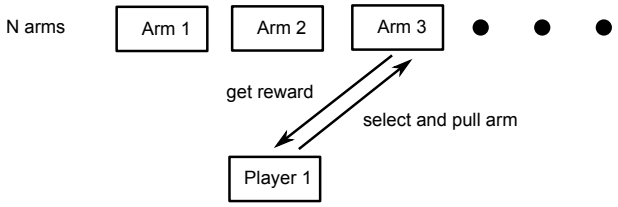

図 2.1: Multi-armed bandit problem.

マルチアームドバンディットの中でも、アームの報酬が毎時刻、アームごとに関連付けられたある確率分布に従って確率的に生み出される場合を、確率的マルチアームドバンディット (Stochastic Multi-Armed Bandit) と呼ぶ。ここでは、その問題の定式化を行う。N 個のアームを期待値が高い順に整数で表して $1, 2, \cdots, N$ とする。それぞれのアームに関連付けられている確率分布 (報酬分布) を $\lambda_1, \lambda_2, \cdots, \lambda_N$ として、その期待値を $\mu_1 > \mu_2 \geq \cdots \geq \mu_N$ とする。ただし、最適アームが一意に決まるということを仮定している。つまり $\mu_1 > \mu_2$ としていることに注意する。時刻 t でプレイヤーがプルするアームを $I(t)$ として、時刻 $t-1$ までにアーム i をプルした回数を $T_i(t) = \sum_{s=1}^{t-1} \mathbb{I}\{I(s) = i\}$ とする。ただし、$\mathbb{I}\{e\}$ は事象 e が真のとき 1 で偽のとき 0 である関数とする。アーム i の $T_i(t)$ 回目のプルで得られる報酬を $X_{i,T_i(t)}$ とする。報酬は i.i.d(independent and identically distributed) を仮定してかつアーム間の独立性も仮定する。アーム i から時刻 $t-1$ までに得られた報酬の平均値を $\hat{\mu}_i(t) = \sum_{s=1}^{T_i(t)} X_{i,s}$ とする。アーム i の最適アームとの期待値の差を $\Delta_i = \mu_1 - \mu_i$ とする。本論文では特別の断りが無い限りは報酬分布は $(1/2)-$subgaussian であると仮定する。確率変数 X が $\sigma-$subgaussian であるとは式 (2.1) を満たすということであり、σ はスケールファクターと呼ばれる。例えば、$[-\sigma, \sigma]$ で有界な分布や分散 σ の正規分布は $\sigma-$subgaussian であり、ベルヌーイ分布は $(1/2)-$subgaussian である。$(1/2)-$subgaussian を仮定した理由は、マルチアームドバンディットの先行研究で $[0, 1]$ で有界な報酬分布が仮定されることが多く、また、$[0, 1]$ で有界な分布は $(1/2)-$subgaussian であるので、先行研究との比較を分かりやすくするためである。ただし、章でのみ定理の証明を簡潔化

するために、代わりに 1−subgaussian を仮定していることに注意。

$$E[\exp(sX)] \leq \exp\left(\frac{1}{2}\sigma^2 s^2\right) \quad (\forall s \in \mathbb{R}) \tag{2.1}$$

最後に、確率的マルチアームドバンディッド問題に対して、あるアルゴリズム $\mathcal{A}$ を使って実際にアームをプルしていく様子を Algorithm 2.1 に示す。

Algorithm 1 Stochastic Multi-armed Bandit

input: algorithm $\mathcal{A}$, reward distributions $\{\lambda_i\}_{i=1,2,\cdots,N}$

1: intialize $t \leftarrow 1$
2: **loop**
3: algorithm $\mathcal{A}$ choose $I(t)$
4: pull arm $I(t)$ and get reward $X_{I(t),T_{I(t)}(t)}$ from $\lambda_{I(t)}$
5: update $\hat{\mu}_{I(t)}(t) = \sum_{s=1}^{T_i(t)} X_{i,s}$
6: $t \leftarrow t + 1$
7: **end loop**

2.2　リグレット最小化

マルチアームドバンディッドの目標設定としては大きく分けて、報酬の累積値を最小化するリグレット最小化と、なるべく少ないプル回数で報酬が最大を見つける最適アーム探索の 2 つが考えられている。本研究のターゲットは最適アーム探索だが、リグレット最小化はマルチアームドバンディッドの中でも重要な問題なので簡単に説明する。リグレットとは式 (2.2) のように定義された、最も良いアームをプルし続けた場合の累積報酬と、アルゴリズムが得た累積報酬の差の期待値のことである。リグレット最小化の目的は、このリグレットを最小化するようなアーム選択をするということである。リグレットを最小化するということは累積報酬の期待値を最大化することでもある。この目的を達成するためには探索と活用のトレードオフを考える必要があり、現在分かっているそれなりに良いアームをプルするべきか (活用)、将来を見越してさらに良いアームを探すためにそれ以外のアームをプルするべきか (探索) をうまく選択する必要がある。これを実現するアルゴリズムは UCB1[7], KL-UCB[8], DMED[9], Thompson Sampling[10], Bayes-UCB[11] 等が知られている。また、式 (2.3) で表現されるようなリグレットの下限も知られている [12]。ただし、KL は Kullback-Leibler divergence である。UCB はオーダーだけ、KL-UCB, DMED, Thompson Sampling, Bayes-UCB は係数も含めて $t \to \infty$ で式 (2.3) の下限に一致する。この意味で、KL-UCB, DMED, Thompson Sampling, Bayes-UCB は最適なアルゴリズムと言われている。

$$R(t) \quad = \quad \sum_{i=1}^{N} \Delta_i T_i(t) \tag{2.2}$$

$$\liminf_{t\to\infty} \frac{R(t)}{\log(t)} \geq \sum_{\Delta_i > 0} \frac{\Delta_i}{\mathrm{KL}(\lambda_i, \lambda_1)} \tag{2.3}$$

2.3　最適アーム探索

本研究のターゲットとしている最適アーム探索という問題を説明する。目的は、平均値が最大のアームを少ない合計プル回数で探し出すことである。ただし、平均値に対して $\mu_1 > \mu_2 \geq \cdots \geq \mu_N$ という風に、最適なアーム $i = 1$ が一意に決まることを仮定する。各アームの平均値は確率的に推定することしかできないので、有限のサンプル数で 100% の確率で最適アームを見つけ出すようなアルゴリズムは不可能である。そのため、適切な問題設定をする必要があるが、現在のところ fixed confidence と fixed budget の 2 つの問題設定が考えられている。fixed confidence は第一種エラー率 δ が与えられた上で、$1-\delta$ 以上の確率で最適アームを見つけるために必要なサンプル数の最小化を目標にする。一方で、fixed budget はサンプル数上限 T が与えられた上で、T 回のサンプル後に最適アームを正しく特定できている確率の最大化を目標にする。従来この 2 つは別々に研究されていて、今のところそれぞれに対応したアルゴリズムを簡単に相互変換する方法も知られていない。fixed budget 設定のアルゴリズムである UCB-E[13] や UGapE(Unified Gap-based Exploration)[14] では問題の難易度が事前に分からないと最適アームを見つけたときにそれが間違えている確率に対して何の保証もできないのに対して、fixed confidence では問題の難易度が未知だとしても最適アームをもし発見できたとしたら、それが間違えている確率は δ 以下だということを保証することが出来る。問題の難易度が事前に分からないケースでも第一種エラー率を δ 以下に抑えた上で最適アーム探索が行えるということは、臨床試験や心理学実験などの応用で必要なことなので、本研究では fixed confidence をターゲットとする。ある最適アーム探索アルゴリズムが必要とするサンプル数のことを sample-complexity と呼ぶ。sample-complexity は総プル回数の期待値で測る考え方もあるが、現在最適アーム探索アルゴリズムの評価や分析として主流なのは、最適アームを見つけて停止した場合の総プル回数を sample-complexity とするものである。これは、アルゴリズムが停止しない可能性を許していることに注意する。sample-complexity をこのように定義した場合、fixed confidence 設定での最適アーム探索問題は式 (2.4) で表現できる。ただし、τ はアルゴリズムが最適アームを見つけて停止する停止時刻、δ は第一種エラー率、$\hat{b}(\tau)$ はアルゴリズムが停止したときに出力する最適アーム候補とする。

$$\begin{aligned} &\text{minimize} && E\left[\tau | \tau \text{ is finite}\right] \\ &\text{subject to} && P\left[\hat{b}(\tau) = 1, \tau \text{ is finite}\right] \geq 1-\delta \end{aligned} \tag{2.4}$$

文献 [15] によって式 (2.5) のような期待値の意味での sample-complexity の下限が示されている。$c > 0$ は正の定数である。文献 [13] で提案されている式 (2.6) で表現されるような H_1 を用いると、sample-complexity の下限は $\Omega(H_1 \log(1/\delta))$ と表現することが出来る。

$$E\left[\sum_{i=1}^{N} T_i(\tau)\right] \geq c\left(\sum_{i=2}^{N} \frac{1}{\Delta_i^2}\right) \log \frac{1}{8\delta} \tag{2.5}$$

$$H_1 = \sum_{i=2}^{N} \frac{1}{\Delta_i^2} \tag{2.6}$$

最適アーム探索と似た問題設定として、文献 [16] で提案された $(\epsilon, \delta)-$PAC(Probably Approximately Correct) というものがある (ただし、$\epsilon > 0$)。これは、最適アームとの平均の差が ϵ 以下であるアームを $\epsilon-$optimum アームと呼び、$1-\delta$ 以上の確率で $\epsilon-$optimum アームをどれか一つ

探し出すという問題である。ϵ よりも小さい差は意味が無いという場合に、複数ある $\epsilon-$optimum アームの中でどれを選んでも良い代わりに sample-complexity を減らすことが出来るという応用上の意味がある。これに対するアルゴリズムは、median elimination[16] があり、このアルゴリズムの sample-complexity は [15] によって示される下限と一致していて、$O(n\epsilon^{-2}\log(\delta^{-1}))$ である。最適アーム探索は $(\epsilon,\delta)-$PAC で言うと $\epsilon=0$ のケースなので、問題の難しさとしては $(\epsilon,\delta)-$PAC よりも最適アーム探索のほうが難しい。難しさの例として、median elimination は $(\epsilon,\delta)-$PAC では最適だが、$\epsilon=0$ ではアルゴリズムが成立しないので最適アーム探索には用いることができない。逆に、本研究での提案手法や関連研究として紹介する最適アーム探索アルゴリズムは全て、少ない変更で $(\epsilon,\delta)-$PAC に対応することが出来る。そのため、本研究でのターゲットは $(\epsilon,\delta)-$PAC では無く最適アーム探索としている。

2.4　関連研究

本研究のターゲットである fixed-confidence に対する最適アーム探索アルゴリズムは大きく 2 つに分けることが出来て、元から最適アーム探索を目標として作られたアルゴリズムと、元々は最適アーム探索アルゴリズムでは無いが、文献 [4] で提案されている LS と組み合わせることによって、最適アーム探索に対応させたアルゴリズムがある。従来提案されている最適アーム探索アルゴリズムを表 2.1 に示す。最適アーム探索に元々対応していないアルゴリズムは、algorithm の列で +LS と表記した。前述のように、sample-complexity は評価として主流なのは、最適アームを見つけて停止した場合の総プル回数を sample-complexity とするものであり、アルゴリズムが停止しない可能性を許している。そのため、アルゴリズムを比較するときは sample-complexity の他に、停止するかどうかにも注目する必要がある。$\mathbb{P}[\text{stop}]=1$ の列は、それぞれのアルゴリズムが有限時刻で必ず停止するかどうかを表している (停止する場合に○)。同じ、sample-complexity を持つアルゴリズムなら停止しないものより停止するものの方が望ましい。現時点で最も良い sample-complexity を持つ、Exp-Gap, lil'UCB は必ず停止することを保証できていない。一方で、LS2 HP(提案手法) は必ず停止するが sample-complexity の分析が出来ていない。

表 2.1: Best-arm identification algorithms in fixed-confidence setting.

algorithm	year	sample-complexity	$\mathbb{P}[\text{stop}]=1$
successive elimination[2]	2002	$O(\sum_{2\le i}^{N}\Delta_i^{-2}\log(N\delta^{-1}\Delta_i^{-2}))$	×
LUCB[5] + LS	2012	not analyzed	×
Exp-Gap[3]	2013	$O(\sum_{2\le i}^{N}\Delta_i^{-2}\log(\delta^{-1}\log(\Delta_i^{-2}))$	×
lil'UCB[4]	2014	$O(\sum_{2\le i}^{N}\Delta_i^{-2}\log(\delta^{-1}\log(\Delta_i^{-2}))$	×
LS1, LS2 elimination(proposed)	2015	$O(\sum_{2\le i}^{N}\Delta_i^{-2}\log(N\delta^{-1}\log(\Delta_i^{-2})))$	×
LS2 HP(proposed)	2015	not analyzed	○

2.4.1　LS(LIL Stopping)

文献 [4] で紹介されている LS(LIL Stopping) は、$1-\delta$ 以上の確率で最適アームを見つけることの出来る停止基準で、任意のアルゴリズムと組み合わせて使うことが出来る。これによって例

えば、リグレット最小化アルゴリズムである UCB1 や、(ϵ, δ)-PAC アルゴリズムである LUCB 等の、最適アーム探索に対応していないアルゴリズムを最適アーム探索に対応させることが出来る。原理は successive elimination とほぼ同じで、高確率で非最適アームであるアームの数を数えて、それが $N-1$ 個になれば残った一つが高確率で最適アームであるという原理である。具体的にはある時刻 t で以下の式が成り立ったときに停止する。$U(T_i)$ は文献 [4] で提案されている sequential test による信頼区間列であり式 (2.12) で表される。

$$\hat{\mu}_{m(t)}(t) - \hat{\mu}_i(t) \geq U(T_{m(t)}(t)) + U(T_i(t)) \quad (\forall i \neq m(t)) \tag{2.7}$$

$$U(T_i) \quad = \quad (1+\sqrt{\epsilon})\sqrt{\frac{2}{t}(1+\epsilon)\left(\frac{\log((1+\epsilon)t+2)}{\delta}\right)} \tag{2.8}$$

このルールに従った場合、停止したときに $m(t)$ が最適アームでは無い確率は $1-\frac{2+\epsilon}{\epsilon}\left(\frac{\delta}{\log(1+\epsilon)}\right)^{1+\epsilon}$ 以下である。エラー率がそのままの δ で出てこないので、実際に用いるときは元々のエラー率を δ' として、式 (2.9) を逆に解いて δ を求めてから LS を行う。

$$\delta' = \frac{2+\epsilon}{\epsilon}\left(\frac{\delta}{\log(1+\epsilon)}\right)^{1+\epsilon} \tag{2.9}$$

2.4.2 Successive Elimination

文献 [2] で提案されている successive elimination は、まず、最適かも知れないアームの集合 A を持っておき、高確率で最適では無いアームをその集合から排除していく。そして、最後に 1 つだけ残ったアームを最適アームとして出力するアルゴリズムである。ただし、$U(t,\delta) = 2\sqrt{\log(cNt^2/\delta)}$ とする。高確率で最適では無いという判断は LS と同じ原理に基づいていて、具体的には式 (2.10) が成り立つようなアーム i を排除していく。

$$\hat{\mu}_{m(t)}(t) - \hat{\mu}_i(t) \geq U(T_{m(t)}(t)) + U(T_i(t)) \tag{2.10}$$

4 章では、この $U(t,\delta)$ を提案する 2 つの sequential test で置き換えたアルゴリズム LS1 elimination, LS2 elimination を提案する。

2.4.3 lil'UCB

文献 [4] で提案されている lil'UCB は、現在知られているアルゴリズムの中で sample-complexity のオーダーが最も小さい最適アーム探索アルゴリズムである。sample-complexity は式 (2.11) で表されるが、文献 [4] によって $N=2$ のときにはこれが最適であることが示されている。

$$E[\tau|\tau \text{ is finite}] = O\left(\sum_{i=2}^{N}\frac{1}{\Delta_i^2}\log\left(\frac{\log(\Delta_i^{-2})}{\Delta_i}\right)\right) \tag{2.11}$$

lil'UCB はオリジナルの UCB1[7] や他の UCB 系アルゴリズムと同様、過去の報酬に基づいてアームごとに UCB(Upper Confidence Bound) を計算して、毎時刻 UCB が最大のアームをプルするというアルゴリズムである。他の UCB 系アルゴリズムの信頼区間は通常の検定に基づいて計算される。また、信頼区間の計算に使われるエラー率は、例えば $\delta = O(1/t^4)$ 等の時刻に対

Algorithm 2 successive elimination

input: $N, \delta > 0$, bound function $U(t, \delta)$

intialize $t \leftarrow N, A \leftarrow \{1, 2, \cdots, N\}$
for $i = 1$ to N **do**
　pull arm i
end for
while $|A| > 1$ **do**
　$m \leftarrow \arg\max_{i \in A} \{\hat{\mu}_i(t)\}$
　$B \leftarrow \left\{ i \in A \mid \hat{\mu}_m(t) - \hat{\mu}_i(t) > U\left(T_m(t), \frac{\delta}{N}\right) + U\left(T_i(t), \frac{\delta}{N}\right) \right\}$
　$A \leftarrow A \setminus B$
　$I(t) \leftarrow \arg\min_{i \in A} \{T_i(t)\}$
　pull $I(t)$ arm
　$t \leftarrow t + 1$
end while

output: an arm in A (algorithm promise $|A| = 1$)

して減少する関数で表現される。一方で、lil'UCB ではエラー率を固定した上で sequential test に基づいて UCB が計算されることが特徴で、さらに sample-complexity が最適なオーダーの sequential test を用いることによって sample-complexity を削減することに成功している。また、LS や successive elimination はアーム間の独立性を仮定しないでも成立するのに対して、lil'UCB はアーム間の独立性を仮定することで初めて成立する停止基準が提案されている。アルゴリズムを以下に示す。ただし、$U(T_i)$ は式 (**??**) で表される。β, λ は lil'UCB 独自の停止基準に関係するパラメータで、$\beta = 1, \lambda = (2+\beta)^2/\beta^2 = 9$ という値が文献 [4] で推奨されていて、そのように設定するとアルゴリズムのエラー率が δ' 以下に収まる。ただし、LS と同様に δ は元々のエラー率 δ' に対して式 (2.9) を解くことによって求める。

$$U(T_i) = (1+\beta)(1+\sqrt{\epsilon})\sqrt{\frac{1}{2t}(1+\epsilon)\left(\frac{\log((1+\epsilon)t+2)}{\delta}\right)} \tag{2.12}$$

Algorithm 3 lil'UCB Algorithm

input: $N, \delta, \lambda > 0$, bound function $U(t, \delta)$

intialize sample each arm once,　$T_i(t) \leftarrow 1, t \leftarrow N$
while $T_i(t) < 1 + \lambda \sum_{j \neq i} T_j$ for all i **do**
　$I(t) \leftarrow \arg\max_{i \in \{1,2,\cdots,N\}} \{\hat{\mu}_i(t) + U(T_i(t))\}$
　sample arm $I(t)$
　$t \leftarrow t + 1$
end while

output: arm that has highest $T_i(t)$

2.4.4 Exp-Gap(Exponential-Gap Elimination Algorithm)

文献 [3] で提案されている Exp-Gap(Exponential-Gap Elimination Algorithm) は、lil'UCB と同様の sample-complexity を持つアルゴリズムで、サブルーチンとして median elimination アルゴリズム [16] を用いていることが特徴である。median elimination は lil'UCB と同様、アーム間の独立性に依存したアルゴリズムである。従って、Exp-Gap もアーム間の独立性を仮定しないと成立しない。また、現在知られているアルゴリズムの中で最も sample-complexity が低いという点でも lil'UCB と同様の性質を持つが、シミュレーションでの実際のサンプル数が lil'UCB よりも 100〜1000 倍多くなることが文献 [4] によって示されているので、今回は実験の比較対象から外す。

2.4.5 LUCB

文献 [17] によって提案された $(\epsilon, \delta)-$PAC の拡張問題で、$\epsilon-$optimum アーム一つだけではなく上位 m 個のアームを見つけるという、m 最適アーム探索という問題がある。具体的には、あるアーム i に対して $\mu_i \leq \mu_m - \epsilon$ が成り立つときに、アーム i は $(\epsilon, m)-$optimum アームであるという定義を用意して、$(\epsilon, m)-$optimum アームを m 個見つけるという問題である。これに対して文献 [5] で紹介されているアルゴリズム LUCB は、$m = 1$ つまり通常の $(\epsilon, \delta)-$PAC に対しても有効であると言われているが、$epsilon = 0$ つまり最適アーム探索の場合には成立しない。しかし、前述の LS を組み合わせて最適アーム探索に対応させた場合、実験的に小さい sample-complexity を達成できるということが、文献 [4] によって示されている。そのため、本研究では LUCB 本来の停止基準は用いずに、代わりに停止基準として LS を用いたものを比較対象として用いる。文献 [5] で紹介されている LUCB をより具体化したアルゴリズム LUCB1 と、LS を組み合わせた場合のアルゴリズムを以下に示す ($m = 1$ の場合)。ただし、アルゴリズム中の $\beta_1(u, t)$ は式 (2.13) で表される。アルゴリズムの仕組みとしては、各ラウンドで平均値が最大のアーム h_* と、UCB が最大のアーム l_* の 2 つのアームをプルするというものである。そのため、各ラウンドで時刻は 2 進む。

$$\beta_1(u, t) = \sqrt{\frac{1}{2u} \log\left(\frac{5nt^4}{4\delta}\right)} \tag{2.13}$$

Algorithm 4 LUCB($m = 1$) + LS

input: confidence $\delta > 0$, sequential test bound function $U(t, \delta)$

1: **intialize** sample each arm once　$T_i(t) \leftarrow 1, t \leftarrow N$
2: **while** LS criteria not holds **do**
3:　$h_* \leftarrow \underset{1 \leq i \leq N}{\arg\max} \{\hat{\mu}_i(t)\}$
4:　$l_* \leftarrow \underset{1 \leq i \leq N, i \neq h_*}{\arg\max} \{\hat{\mu}_i(t) + \beta_1(T_i(t), t)\}$
5:　sample arm h_* and l_*
6:　$t \leftarrow t + 2$
7: **end while**

output: arm that has highest mean

第3章
Sequential Testの改良

3.1 Sequential Test

sequential testに関して簡単な説明を行う。証明の簡略化のために、この章でのみ報酬分布として(1/2)−subgaussianの代わりに1−subgaussianを仮定していることに注意。t-検定や二項検定などの基本的な統計学的仮説検定では、まず、サンプル数Tを適当に決めてT個のサンプルを集める。そして、サンプル元の分布の平均が0に等しいという帰無仮説を設定して、T個のサンプルがその帰無仮説に従うかどうかを検定する。第一種エラー率と第二種エラー率というものがあり、第一種エラー率は帰無仮説が正しいのに間違っていると結論付けてしまう確率のことで、第二種エラー率は帰無仮説が間違っているのに正しいと結論付けてしまう確率のことである。通常は、第一種エラー率の上限を適当に5%とかで与えた上で、第二種エラー率を最小化するような検定を行う。ここで、T個のサンプルを集めている途中で検定を行うことが出来ないということに注意しないといけない。もし、T個のサンプルを集めている途中で何度も検定を行うと、全体としての第一種エラー率が最初に与えた上限を超えてしまうからである。

一方で、sequential testというのは、T個のサンプルを集めている途中で検定を行うような方式である。Tは有限の値に設定することもあれば、本稿のように∞を想定する場合もある。この利点は、サンプル数がNに到達する前に帰無仮説が間違っているということが分かれば、検定に必要なサンプル数を低く抑えることが出来るところにある。普通の検定では、第一種エラー率が同じ二つの検定の良し悪しを比べる場合、第二種エラー率が低い方が良い検定とみなされるが、∞個のサンプルを想定する場合、帰無仮説が間違っていると言えない場合、検定を終了することが無いので第二種エラーが発生するということは無い。その代わりに、sequential testでは検定が終了するまでのサンプル数が少ない方が良い検定だとみなされる。実際には、sequential testは毎時刻である範囲を計算して、平均がそこからはみ出たら検定を終了する。この毎時刻計算される範囲を本稿では信頼区間列と呼ぶことにする。信頼区間列が狭いほうが検定が終了するのが早いので良い検定だと言える。説明のために、最も簡単なsequential testの例を示す。問題として、1枚のコインがあったとしてそれを投げたときに表が出る確率と裏が出る確率が等しいかどうかを検定してみる。実際には、表が出たら1、裏が出たら−1としてその平均値が0かどうかを検定することとする。sequential testでは無い通常の検定の場合、サンプル数T回、第一種エラー率δ、T個のサンプルの平均値を$\hat{\mu}_T$として、Hoeffding Inequalityを用いると式(3.1)が成り立つ。

$$P\left[\hat{\mu}_T \geq \sqrt{\frac{2\log(1/\delta)}{T}}\right] \leq \delta \tag{3.1}$$

つまり、サンプル数T回、第一種エラー率δの片側信頼区間は$\sqrt{2\log(1/\delta)/T}$と言うことが出来る。時刻tでの第一種エラー率$\delta(t)$というものを用意して、毎時刻エラー率$\delta(t)$で通常の検

定を行うことで sequential test を行うことを考える。具体的には $\delta(t)$ を以下の式のように設定する。$\delta(t)$ は $\sum_{t=1}^{\infty}\delta(t) \leq \delta$ になるような関数なら何でも良いが、今回は信頼区間列のオーダーが小さくなるようなものを選んだ。

$$\delta(t) = \frac{2\log 2}{3(1+t)\log^2(1+t)}\delta \tag{3.2}$$

式 (3.2) を式 (3.1) に代入して、以下のように信頼区間列の狭まり方を確認してみる。ただし、$C = \log(3/(2\log 2))$ である。

$$P\left[\exists t,\ \ \hat{\mu}_t \geq \sqrt{\frac{2}{N}\left(\log(1+t) + \log(\log^2(1+t)) + \log(1/\delta) + C\right)}\right] \leq \delta \tag{3.3}$$

信頼区間列の t に関するオーダーに注目すると $O(\sqrt{\log(t)/t})$ である。後述するように、最適なオーダーは $O(\sqrt{\log\log(t)/t})$ なので、この方法では最適なオーダーは達成出来ない。しかし、通常の検定さえ行うことが出来ればそれをそのまま sequential test に拡張することが可能なので、sequential test がまだ開発されていないような検定を拡張するのには有効である。

3.2　LIL(Law of the Iterated logarithm)

sequential test と関係の深い Law of the Iterated Logarithm という法則を紹介する。まず、ランダムウォークを考える。このランダムウォークが 0 より大きい確率で全ての t で収まるような領域はどんな領域かを示すのが、law of the iterated logarithm である。具体的には、平均が 0 で分散が 1 の i.i.d(independent and identically distributed) な確率変数列 $X_1, X_2, \cdots, X_t$ を考える。その和を $S_t = X_1 + X_2 + \cdots + X_t$ とする。このとき以下のような式が成り立つ。ただし、a.s. は almost surely である。

$$\limsup_{t\to+\infty} \frac{S_t}{\sqrt{t\log\log(t)}} = \sqrt{2} \quad a.s. \tag{3.4}$$

この式によるとランダムウォーク S_t が全ての t で収まるような領域は、$|S_t| < O(\sqrt{t\log\log(t)})$ であることが示唆される。逆に、これより小さいオーダーの領域では t が大きくなるにつれていつかは S_t が領域をはみ出してしまう。

3.3　最適な 1 変数 Sequential Test の提案

3.3.1　理論的背景

LIL の結果を考えると、sequential test の信頼区間列は $t \to +\infty$ で $\sqrt{2\log\log(t)/t}$ に近づけば最適と言える。このような性質を満たす sequential test を最適と呼ぶことにする。sequential test の信頼区間列を作る試みは過去に行われているが、[4] では任意の $\epsilon > 0$ に対して $\sqrt{(2+\epsilon)t\log\log(t)}$ であり、[6] では $\sqrt{3t\log\log(t)}$ というように漸近最適では無い。また、[18] によると数値積分をすることで最適な sequential test を行うことが出来るが、通常の検定と比べて計算量が多くなってしまう。そこで、最適な信頼区間列を現実的な計算量で計算する方法を提案する。確率空間 $(\Omega, \mathcal{F}, P)$ と、離散時間フィルトレーション $\{\mathcal{F}_t\}_{t=0,1,\cdots}$ を用意する。

$\{\Delta_t\}_{t=1,2,\cdots}$ を各 Δ_t が 1-subgaussian に従う平均 0 の確率変数とする。このとき $X_t = \sum_{k=1}^{t} \Delta_k$ を考えると martingale になっている。ただし、$X_0 = 0$ とする。マルチアームドバンディットでは、$\{\Delta_t\}_{t=1,2,\cdots}$ が毎時刻得られる報酬に、$X_t = \sum_{k=1}^{t} \Delta_k$ がその合計に対応する。まず、以下のような関数 $h(x)$ を定義する。プロットを図 3.1 に示す。

定義 1. 関数 $h(x)$ を以下の式で定義する。ただし、$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x exp(-t^2)$ を誤差関数とする。

$$h(x) = \frac{\sqrt{x}}{\sqrt{\pi}\ \mathrm{erf}(\sqrt{x})} \exp(-x) \tag{3.5}$$

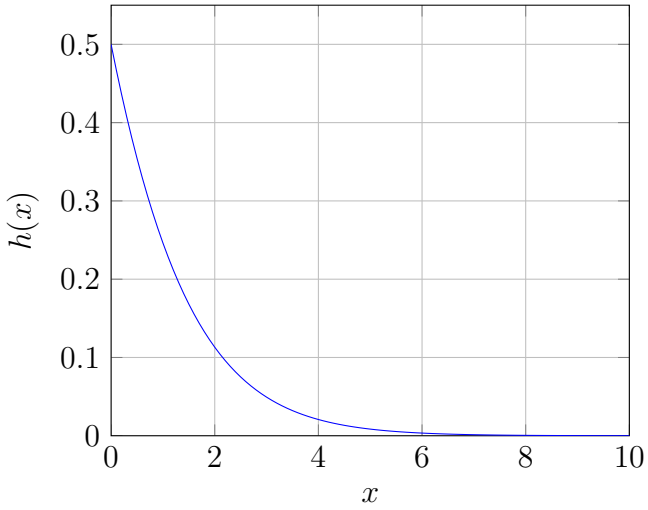

図 3.1: $h(x)$ plot.

次の定理 1 が subgaussian 確率変数列の平均に対する sequential test の定理である。帰無仮説は全ての Δ_t が 1-subgaussian でありかつ平均 0 であることとする。$\{\delta_t\}_{t=1,2,\cdots}$ というものを用意して、$\delta_t\ u_t\ p(u_t) \geq h\,(t\ u_t^2/2)$ という条件が満たされたら帰無仮説が棄却されたとして停止する。ただし、各 δ_t は時刻 $t-1$ までの情報によって決まるとする。このとき第一種エラー率、つまり停止した場合に帰無仮説が正しい確率は $\sup_{0<t}\{\delta_t\}$ 以下になる。

定理 1. $u_t = \frac{1}{t} X_t$ として、$p(x) > 0$ を $0 < x$ の範囲で定義された $\int_0^\infty p(x)dx = 1$ である凸関数とする。また、$\{\delta\}_{t=1,2,\cdots}$ を各 $\delta_t > 0$ が $\mathcal{F}_{t-1}$ 可測である確率変数列とする。このとき以下の式が成り立つ。

$$\mathbb{P}\left[\exists t > 0, \quad \delta_t\ u_t\ p(u_t) \geq h\left(\frac{1}{2} t\ u_t^2\right)\right] \leq \sup_{0<t}\{\delta_t\} \tag{3.6}$$

証明. 付録 A.3 に掲載。

これを使って実際に sequential test を行うためには $p(x)$ を具体的に与えないといけない。$p(x)$ に対してどういう関数を設定するかだが、本稿では定義 2 のように $q(x)$ を用意してその積分が 1 になるように正規化したものを $p(x)$ として設定する。この式は、$p(x)$ が凸になるということと、後述するように信頼区間列が漸近最適になるということの両方を満たすように、試行錯誤の上で選んだ。ただし、$p(x)$ が凸になることは数値計算的にしか確認していないので、この $p(x)$ を使った sequential test が正しいという数学的な保証は出来ていない。$p(x)$ のプロッ

トと、$p(x)$ が凸であることの数値的な確認のために $p''(x)$ のプロットを図 3.2 に示す。$p(x)$ を計算するためには $Q := \int_0^\infty q(x)dx$ を求めないといけないが、これは数値積分で求めた結果、$\int_0^\infty f(x)dx \approx 2.519$ であった。詳細は付録 A.1 に示す。

定義 2. $p(x), q(x), Q$ を以下のように定義する。$p(x)$ は $q(x)$ を積分が 1 になるように正規化したものである。

$$\begin{aligned} p(x) &= \frac{q(x)}{Q} \\ q(x) &= \frac{1}{x(2.085x + \log(1+1/x)\log^2(1+\log(1+1/x)))} \\ Q &= \int_0^\infty q(x)dx \end{aligned}$$

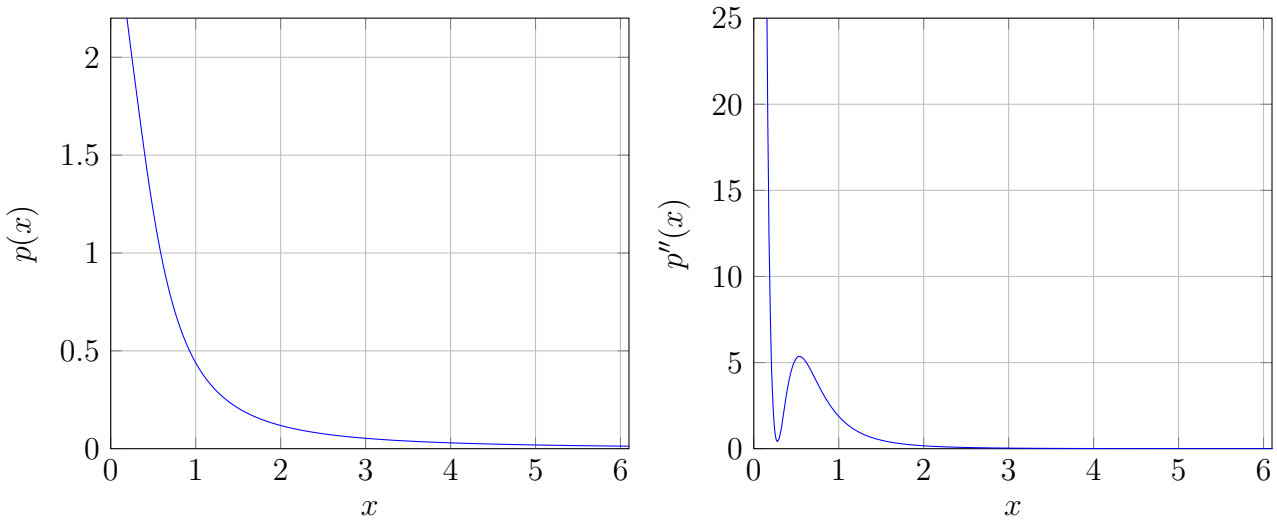

図 3.2: $p(x)$ and $p''(x)$ plot.

次に信頼区間列が LIL から考えて最適であることを確認する。そのためにまずは、定理 1 に式 (3.7) を代入して計算される信頼区間列を、陽に計算出来る関数 $u(t)$ で評価した上で、それを LIL から考えられる最適な $\sqrt{2\log\log(t)/t}$ と比較する。それが次の定理 2 である。

定理 2. 任意の $\delta \in (0, 1]$ に対して、以下のような関数 $u(t)$ を考える。

$$u(t) = \sqrt{\frac{2}{t}} \sqrt{-\frac{1}{2} W_{-1}\left(-\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2}\right)} \tag{3.7}$$

このとき $T = \frac{25a^2Q^2}{2\delta^2 e^2 \pi}$ として、全ての整数 $t > T$ で以下の式が成り立つ。

$$\delta u(t) p(u(t)) \geq h\left(\frac{1}{2}tu^2(t)\right) \tag{3.8}$$

また、$t \to \infty$ で次のような性質を持つ。

$$\limsup_{t\to\infty} \frac{u(t)}{\sqrt{t\log\log(t)}} = \sqrt{2} \tag{3.9}$$

証明. 付録??に掲載。

定理 1 に式 (3.7) を代入して行われる sequential test の信頼区間列 $U_1(t,\delta)$ を次のように定義する。

定義 3. 任意の $t \in 1,2,\cdots, \delta \in (0,1]$ に対して、信頼区間列 $U_1(t,\delta)$ を次のように定義する。

$$U_1(t,\delta) = \inf\left\{u|\delta\ u\ p(u) \geq h\left(\frac{1}{2}t\ u^2\right)\right\} \tag{3.10}$$

3.3.2　数値計算による有効性の確認

sequential test の有効性を確認するために、信頼区間列のプロットを従来手法と比較してみる。比較対象は、前述の第一種エラー率を配分することで通常の検定を拡張する方法 (ナイーブと呼ぶ) と、[6] の Theorem 2 で提案されているものと、lil'UCB[4] の Lemma 3 で提案されているものとする。各手法の信頼区間列の式を以下に示す。過去の結果は両側信頼区間に対する結果も含まれているので、公平な比較のため、全て両側信頼区間に揃えている。そのため、提案手法やナイーブな手法は δ の代わりに $\delta/2$ を用いている。Theorem 2 は Initial Time というものがあり、$t \leq 173\log\left(\frac{4}{\delta}\right)$ では検定を行わない。lil'UCB は右辺の確率が δ では無いので、与えた第一種エラー率上限が右辺に等しいとして δ を逆に解いて求める。ϵ は適当な定数で、[4] の中では $\epsilon = 0.01$ が推奨されているのでそれを用いる。小さい t についてのプロットを図 3.3 に、大きい t についてのプロットを図 3.4 に示す。大きい t についてはオーダーに注目するために、信頼区間列を $\sqrt{\log\log(t)/t}$ で割ったものをプロットした。asymp は LIL から考えられる最適な $\sqrt{2\log\log(t)/t}$ に相当するラインである。

$$\mathbb{P}\left[\exists t>0,\quad \frac{\delta}{2}\,|\hat{\mu}_t|\,p(|\hat{\mu}_t|) \geq h\left(\frac{1}{2}t\,\hat{\mu}_t^2\right)\right] \leq \delta \quad \text{(proposed)}$$

$$\mathbb{P}\left[\exists t>0,\ \ |\hat{\mu}_t| \geq \sqrt{\frac{2}{t}\left(\log(1+t)+\log\left(\frac{\log^2(1+t)}{\log(2)}\right)+\log\left(\frac{2}{\delta}\right)\right)}\right] \leq \delta \quad \text{(naive)}$$

$$\mathbb{P}\left[\exists t>173\log\left(\frac{4}{\delta}\right),\ \ |\hat{\mu}_t| \geq \sqrt{\frac{3}{t}\left(2\log\log\left(\frac{5}{2}t\right)+\log\left(\frac{2}{\delta}\right)\right)}\right] \leq \delta \quad \text{(theorem2[6])}$$

$$\mathbb{P}\left[\exists t>0,\ \ |\hat{\mu}_t| \geq (1+\sqrt{\epsilon})\sqrt{\frac{2}{t}(1+\epsilon)\left(\frac{\log((1+\epsilon)t+2)}{\delta}\right)}\right] \leq \frac{2(2+\epsilon)}{\epsilon}\left(\frac{\delta}{\log(1+\epsilon)}\right)^{1+\epsilon} \quad \text{(lil'UCB[4])}$$

さらに優劣を分かりやすくするために、平均が 0 では無かったときにその差を検出するために必要なサンプル数に注目してみる。提案手法を基準 (= 1) として、ある平均のずれ Δ を検出するためにその他の手法が提案手法の何倍のサンプル数が必要かを図 3.5 にプロットした。実用上は小さい t での性能が問題になるので、小さい t でのみプロットした。

次に、与えた第一種エラー率の上限 δ に対して、実際の第一種エラー率がどのくらいになっているかを見てみる。実際の第一種エラー率が与えた δ に近いほど、タイトに評価できていると言える。第一種エラー率はモンテカルロシミュレーションで計算するが、無限サンプルの sequential test なので、本当の第一種エラー率は知ることが出来ない。有限の $t \leq T$ に対して調べることにする。実験条件は、モンテカルロの試行回数を 10 万回。$T = 100000$ とした。シ

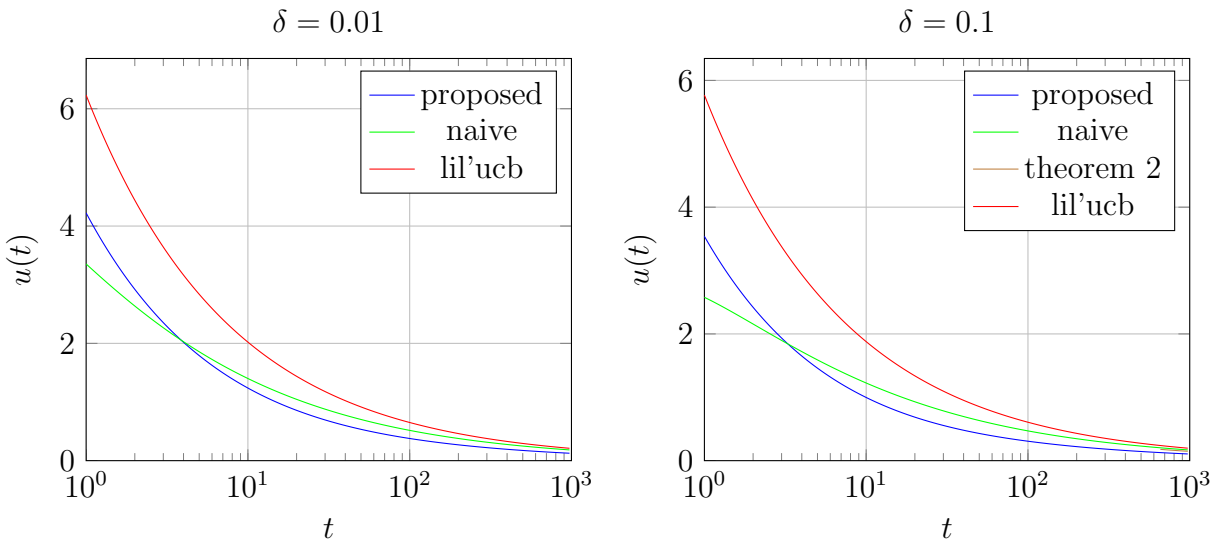

図 3.3: Sequential test bounds $u(t)$ in small t. $(\delta = 0.01, 0.1)$

ミュレーションによって得た第一種エラー率の信頼区間は、ベータ分布の累積密度関数を用いた二項検定で信頼水準 95% で計算した。結果を図 3.6、表 3.1 に示す。表の中で 0 は第一種エラー率が低すぎて信頼区間を計算出来なかった部分である。lil'UCB は全ての δ で第一種エラー率が低すぎるので図 3.6 にプロットしていない。

表 3.1: Reference type I error rate δ vs actual error rate. $(1.2\text{E-}3 = 1.2 \times 10^{-3})$

δ	proposed	naive	theorem 2	lil'UCB($\epsilon = 0.01$)
0.5	2.31E-1 [2.28E-1 ~ 2.33E-1]	7.48E-2 [7.32E-2 ~ 7.64E-2]	1.45E-3 [1.22E-3 ~ 1.70E-3]	0
0.25	1.18E-1 [1.16E-1 ~ 1.20E-1]	3.48E-2 [3.37E-2 ~ 3.60E-2]	4.60E-4 [3.37E-4 ~ 6.02E-4]	0
0.125	6.14E-2 [5.99E-2 ~ 6.28E-2]	1.66E-2 [1.58E-2 ~ 1.73E-2]	2.00E-4 [1.22E-4 ~ 2.97E-4]	0
0.0625	3.09E-2 [2.98E-2 ~ 3.20E-2]	7.45E-3 [6.93E-3 ~ 8.00E-3]	6.00E-5 [2.20E-5 ~ 1.17E-4]	0
0.03125	1.54E-2 [1.46E-2 ~ 1.62E-2]	3.76E-3 [3.39E-3 ~ 4.15E-3]	0	0
0.015625	7.54E-3 [7.01E-3 ~ 8.09E-3]	1.86E-3 [1.60E-3 ~ 2.14E-3]	0	0
0.0078125	3.98E-3 [3.60E-3 ~ 4.38E-3]	8.90E-4 [7.15E-4 ~ 1.08E-3]	0	0

また、第一種エラーがどのくらいの t で多く起きているかを調べてみる。時刻 t 以前で第一種エラーが発生する確率を $\alpha(t)$ と置く。alpha spending function と呼ばれるこの関数を、モンテカルロシミュレーションで計算してみる。これは絶対的な優劣を付けられるものでは無く、代わりに検定力の配分を見ることが出来る。例えば、alpha spending function が小さい t で大きくなっていれば、小さい t により強い検定力を配分している。逆に、小さい t では小さくて t が大きくなるにつれて大きくなっていれば、大きい t に強い検定力を配分していると言える。lil'UCB は実際の第一種エラー率が低すぎて実用的では無いので、それを除いた手法について結果を以下に示す。実験条件は、$\delta = 0.25$ としてその他は前述の第一種エラー率を計算した実験と同じである。

実験結果をまとめてみる。単純な naive がより複雑な theorem 2 や lil'UCB よりも小さい t で良

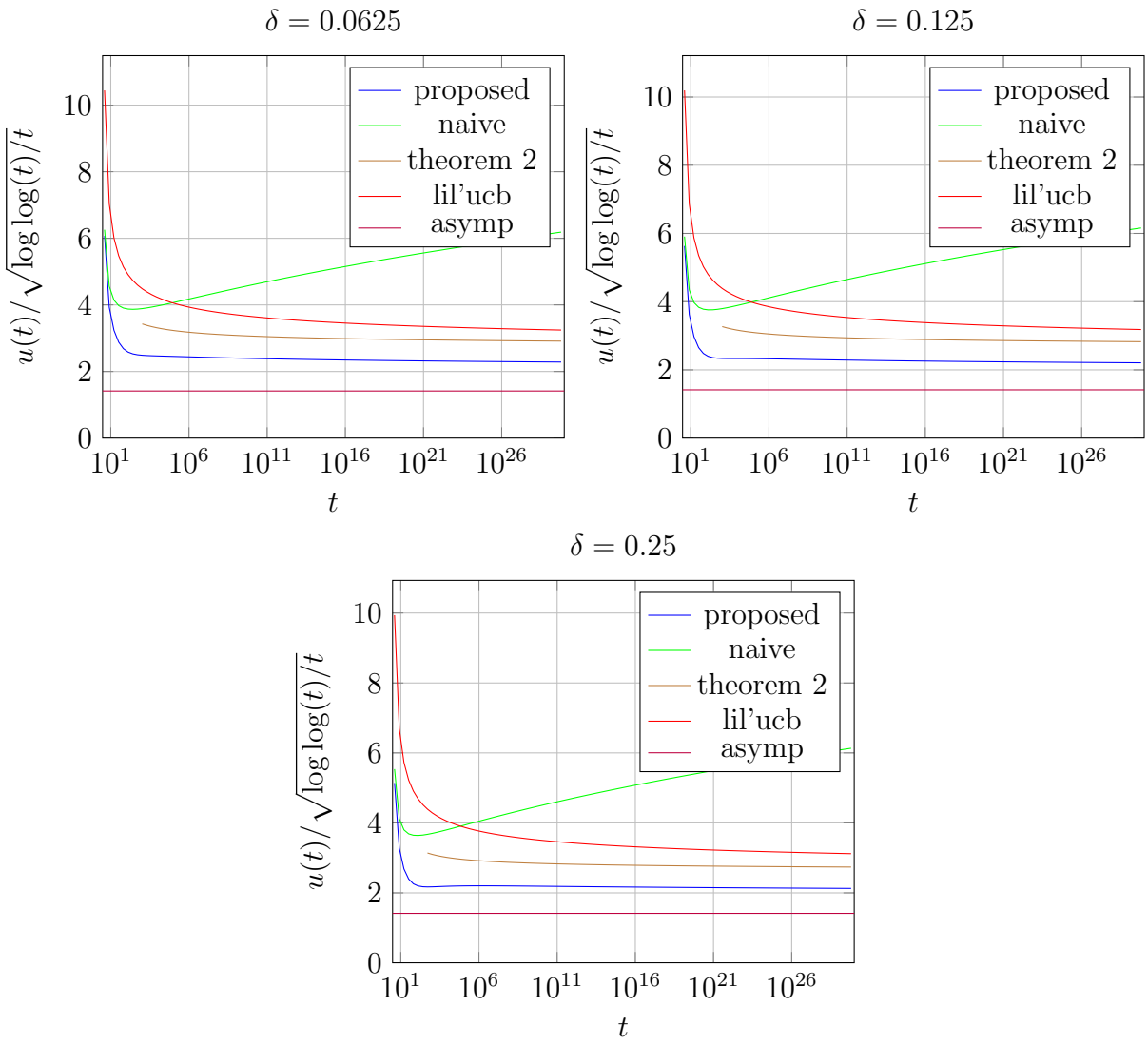

図 3.4: Sequential test bounds $u(t)$ in large t. $(\delta = 0.0625, 0.125, 0.25)$

い結果を出しているので実用的だが、theorem 2 や lil'UCB はオーダーが最適な $O(\sqrt{\log\log(t)/t})$ なのに対して、naive は $O(\sqrt{\log(t)/t})$ で最適では無い。一方で提案手法は小さい t での実際の性能が他のどの手法よりも高く、十分大きい t でオーダーだけでは無く係数まで含めて最適な $\sqrt{2\log\log(t)/t}$ に近づく。ただし信頼区間列を陽に計算出来ないので計算量とのトレードオフがある。与えた δ に対して実際の第一種エラー率がどう変わるかだが、提案手法、naive、 theorem 2、 lil'UCB の順でタイトに評価できていることが分かる。alpha spending function の形は、naive は小さい t に検定力が偏っていて、theorem 2 は大きい t に検定力が偏っていることが分かる。提案手法の alpha spending function は両者の中間になっている。トータルで見て、提案手法は従来手法と比べて実用的であると言える。

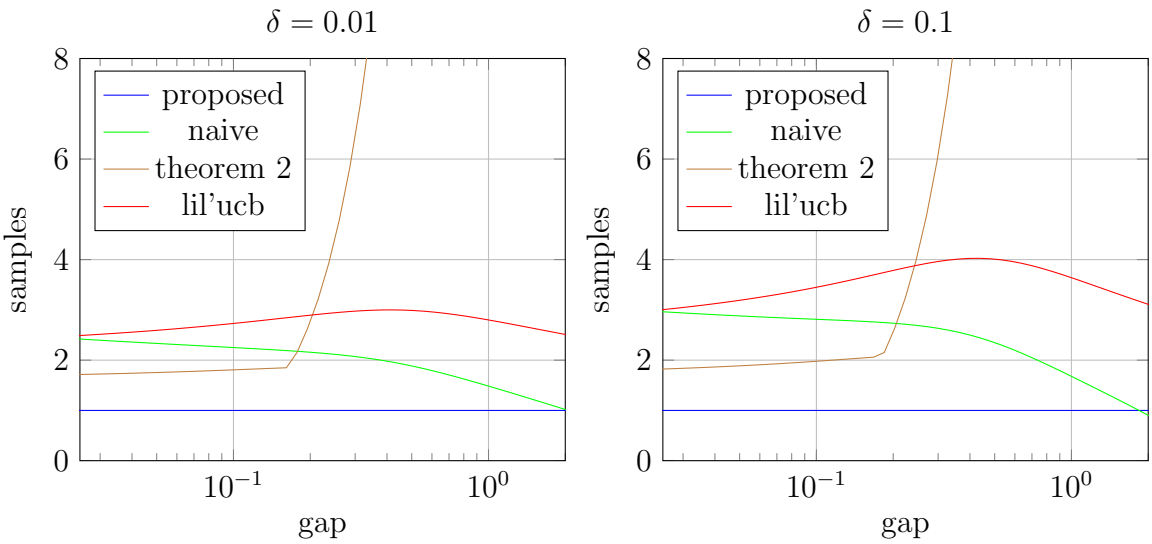

図 3.5: Gap vs required samples. ($\delta = 0.01, 0.1$)

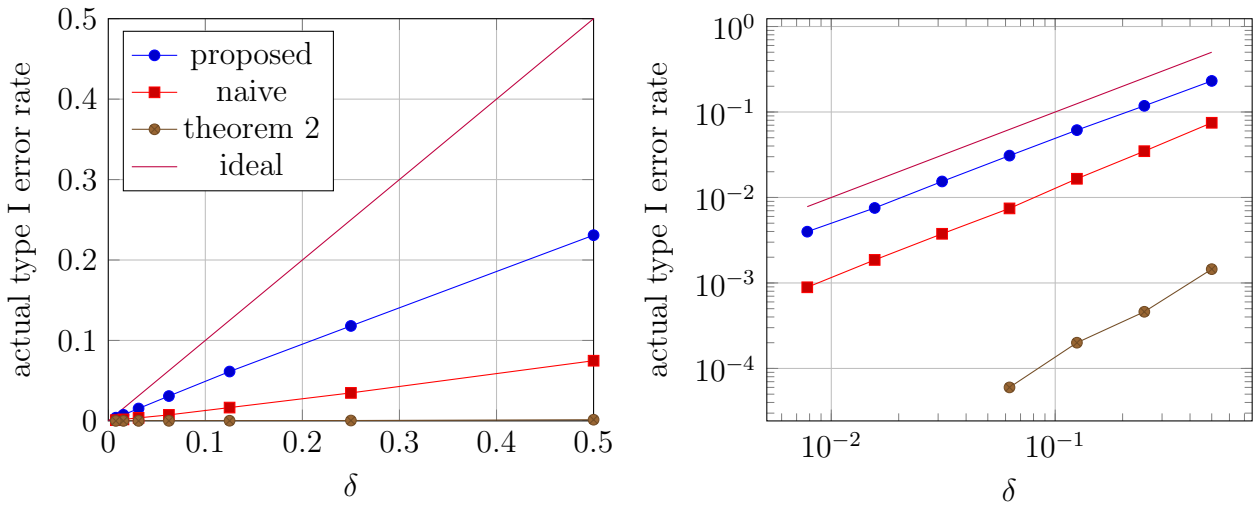

図 3.6: δ vs actual type I error rate plot. (linear and logarithmic)

3.4　2 変数 Sequential Test の提案

3.4.1　理論的背景

最適アーム探索では、最終的に 2 つのアームの平均の差がゼロより大きいということを言いたい。これは、それぞれのアームに対して sequential test から信頼区間列を計算して、その区間に被りが無いということで確認することが出来るが、実は 2 つのアームのプル回数が同じくらいの場合には、2 つのアームの平均の差に対する特別の sequential test を用意することによって、最大で必要プル回数を半分程度にすることが出来る。

定理 3. 平均が 0 でスケールファクター 1 の 2 つの sub-gaussian 確率変数列 $X_1, X_2, \cdots, X_t$ と $Y_1, Y_2, \cdots, Y_t$ を考える。$u_t = \frac{1}{n(t)} \sum_{t_2=1}^{n}(t) X_{t_2}, v_t = \frac{1}{m(t)} \sum_{t_2=1}^{m}(t) Y_{t_2}$ として、$p(x) > 0$ を $0 \leq x$ の範囲で凸で $\int_0^\infty p(x)dx = 1$ である関数とする。このとき任意の $\delta \geq 0$ に対して以下の式が成

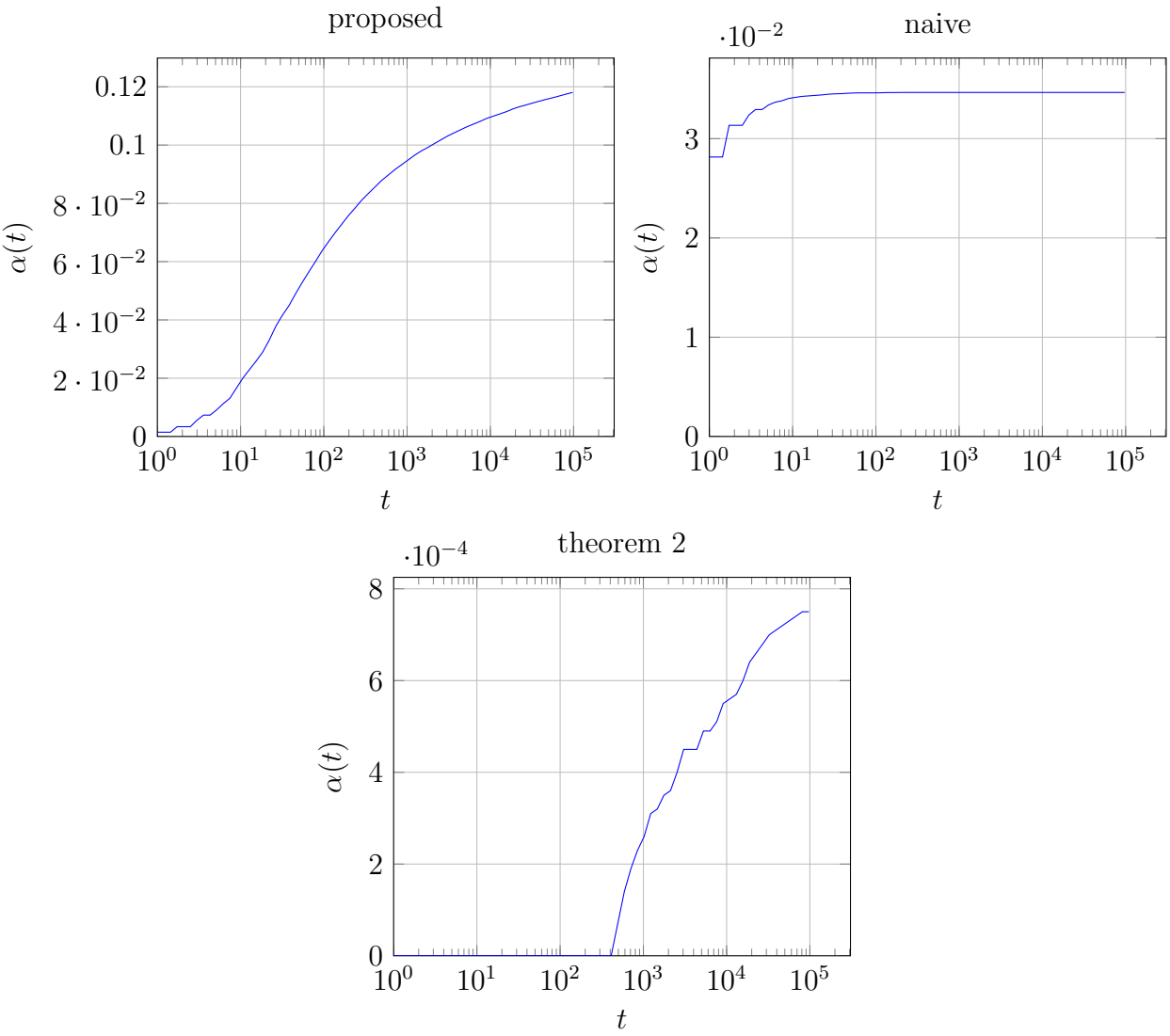

図 3.7: Alpha spending function. ($\delta = 0.25$)

り立つ。

$$\mathbb{P}\left[\exists t > 0, \quad \delta\ u_t\ p(u_t)\ v_t\ p(v_t) \geq h\left(\frac{1}{2}n(t)\ u_t^2\right) h\left(\frac{1}{2}m(t)\ v_t^2\right)\right] \leq \delta \tag{3.11}$$

証明. 付録に掲載。

また、以下のように $\delta = \delta_1 \cdot \delta_2$ と分解すれば、1 変数の場合の sequential test を組み合わせて 2 変数の sequential test が行えることがわかる。

$$\begin{cases} \delta_1\ u_t\ p(u_t) & \geq h\left(\frac{1}{2}n(t)\ u_t^2\right) \\ \delta_2\ v_t\ p(v_t) & \geq h\left(\frac{1}{2}m(t)\ v_t^2\right) \end{cases}$$
$$\Rightarrow \delta\ u_t\ p(u_t)\ v_t\ p(v_t) \geq h\left(\frac{1}{2}n(t)\ u_t^2\right) h\left(\frac{1}{2}m(t)\ v_t^2\right) \tag{3.12}$$

これを用いて実際に 2 つのアーム間の平均の差を検定する方法を説明する。今、ある時刻 t で 2 つのアームの平均の差が Δ だったとする。u_t, v_t はそれぞれ 2 つのアームの真の平均から

のずれを意味する。帰無仮説は 2 つのアームの平均が等しいこととする。帰無仮説が正しいとすると、$u_t + v_t = \Delta$ が成り立つ。u_t, v_t は未知なので直接上の定理の条件を満たしているかは確認できないが、ありうる全ての $u_t + v_t = \Delta$ で上の定理の条件を満たすということが確認できれば、帰無仮説は正しくないということが結論付けられる。問題は、ありうる全ての (u_t, v_t) で定理の条件を満たしているかを確認することだが、$x < 0$ で $p(x) = 0$ なので、$u_t < 0$ または $v_t < 0$ で明らかに定理の条件を満たさない。つまり、この検定単独では検定が行えない。そこで、図 3.8 のように 1 変数の sequential test を 2 つ組み合わせて、、全部で 3 つの検定を組み合わせて検定を行うことを考える。第一種エラー率は均等に $\delta/3$ で割り振る。2 変数 sequential test では埋めきれない隙間を 1 変数 sequential test が補っていることが図からわかる。3 つの検定

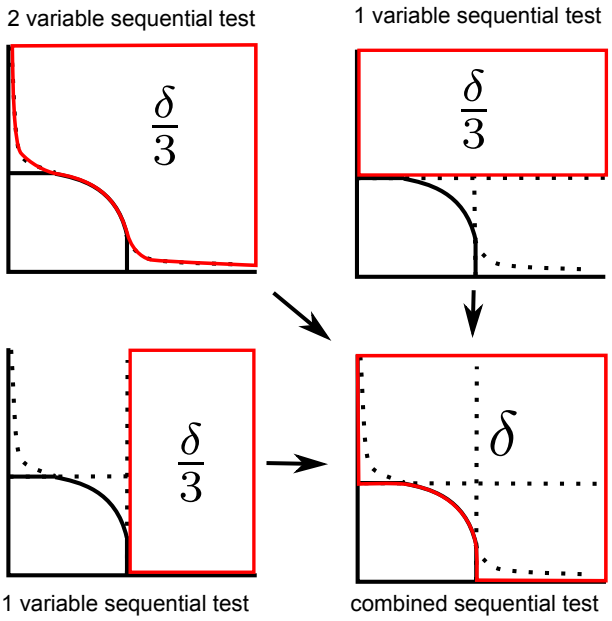

図 3.8: Sequential test combination. Regions surrounded by red lines represent events occurring with probabilities lower than δ or $\delta/3$. Axes represents u_t, v_t.

を組み合わせているので実装が複雑になるように思えるが、実際は前述の定理で確認したとおり 2 変数の sequential test は、1 変数の sequential test に分解して行っても良いので簡単に実装できる。最終的に、実際に 2 つのアーム間の平均の差が 0 かどうかをエラー率 δ 以下で検定する方法は、式 (3.13) で定義される ω を計算し、$\omega(n, m, \Delta) < \frac{\delta}{3}$ が成り立ったときに sequential test を終了というものである。

$$\omega(n,m,\Delta)=\max_{u+v=\Delta}\left\{\frac{h\left(\frac{1}{2}n\,u^2\right)h\left(\frac{1}{2}m\,v^2\right)}{u\,p(u)\,v\,p(v)}\middle|\frac{h\left(\frac{1}{2}n\,u^2\right)}{u\,p(u)}\leq 1,\frac{h\left(\frac{1}{2}m\,v^2\right)}{v\,p(v)}\leq 1\right\} \tag{3.13}$$

このアルゴリズムでは、最大値の探索を行う必要がある。探索が効率良く動作するためには、$\frac{h\left(\frac{1}{2}n\,u^2\right)h\left(\frac{1}{2}m\,v^2\right)}{u\,p(u)\,v\,p(v)}$ の極値が一つである必要がある。極値が一つであるということは数学的に証明できていないが、数値的に正しいということを確認した。本論文では黄金分割探索によって実装した。数値的な根拠を以下に示す。$\frac{h\left(\frac{1}{2}n\,u^2\right)h\left(\frac{1}{2}m\,v^2\right)}{u\,p(u)\,v\,p(v)}$ の対数を取ると以下のような式になる。

$$\log\left(\frac{h\left(\frac{1}{2}n\,u^2\right)}{u\,p(u)}\right)+\log\left(\frac{h\left(\frac{1}{2}m\,v^2\right)}{v\,p(v)}\right) \tag{3.14}$$

二つの項は式の形が同じなので、以下のような関数 $f_n(x)$ を考えると、$f_n(u) + f_m(v)$ と表現できる。

$$f_n(x) = \log\left(\frac{h\left(\frac{1}{2}n\ x^2\right)}{x\ p(x)}\right) \tag{3.15}$$

ここで、全ての $n \geq 1$ について $f_n(x)$ が $f_n(x) \leq 0$ となる範囲で凹であることが言えれば、元々の評価関数の極値は探索範囲内で一つ以下であることが言える。ただし、$f_n(x) \leq 0$ は探索範囲内という条件である。小さい $n = 1, 2, 3, 4$ についてのプロットを図 3.9 に示す。プロットの範囲では $f_n(x) \leq 0$ の範囲で凹になっていることが分かる。大きい n に関しては $f_n(x)$ をもう少

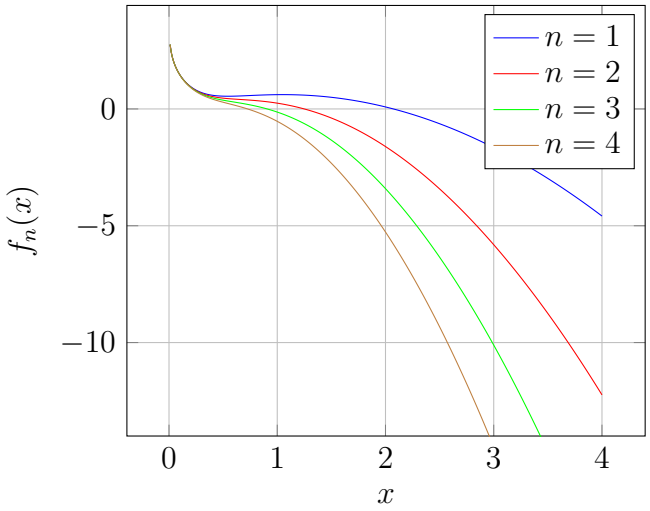

図 3.9: $f_1(x), f_2(x), f_3(x)$ and $f_4(x)$ plot.

し詳しく見ていくと、式 (3.16) のようになる。この式で支配的な項は先頭の $-\frac{1}{2}n\ x^2$ なので、n が大きくなるにつれて放物線に近づくことが分かる。従って、大きい n でも $f_n(x)$ は凹であることが予想される。

$$\begin{aligned} f_n(x) &= \log\left(\frac{h\left(\frac{1}{2}n\ x^2\right)}{x\ p(x)}\right) \\ &= \log\left(h\left(\frac{1}{2}n\ x^2\right)\right) - \log\left(x\ p(x)\right) \\ &= -\frac{1}{2}n\ x^2 + \log\left(\frac{\sqrt{\frac{1}{2}n\ x^2}}{\sqrt{\pi}\ \mathrm{erf}\left(\frac{1}{2}n\ x^2\right)}\right) - \log\left(x\ p(x)\right) \end{aligned} \tag{3.16}$$

3.4.2　数値計算による有効性の確認

2 つのアームの平均の差を、1 変数 sequential test を用いてそれぞれのアームの信頼区間に被りが無いかどうかで検定した場合と、2 変数 sequential test を用いて両方合わせて検定した場合とで必要サンプル数がどの程度違うかを確認する。具体的には、1 変数 sequential test の必要サンプル数に対しての 2 変数 sequential test の必要サンプル数の比が、2 つのアームのプル回数の比 $n/(n+m)$ によってどう変わるかを調べた。必要サンプル数は、平均の差 $\Delta = 0.1$ を

エラー率 $\delta = 0.1, 0.01$ で有意と言うために必要なサンプル数として計算した。結果を図 3.10 に示す。$\delta = 0.1$ については 2 変数 sequential test が常に必要サンプル数が少ないということはなく、2 つのアームのプル回数に偏りがあるときは 1 変数 sequential test の方が少なく、プル回数が同じくらいのときは 2 変数 sequential test の方が少ない。$\delta = 0.01$ では、ほとんどの領域で 2 変数 sequential test の方が必要サンプル数が少なくなっている。今回提案する LS1 elimination, LS2 elimination では sequential test を行う 2 つのアームのプル回数はほぼ同じになる。また、LS2 HP でも最後に残る 2 つのアームのプル回数はほぼ同じになるので、2 変数 sequential test の必要サンプル数を減らす効果が強く現われることが期待される。

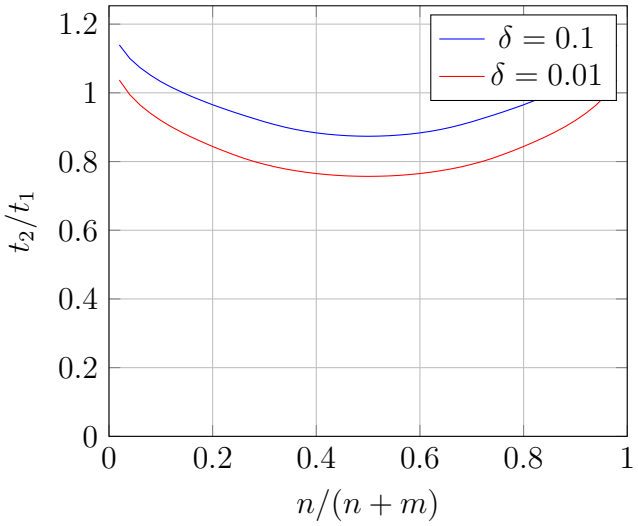

図 3.10: Ratio of required samples of 1 variable sequential test(t_1) and that of 2 variable sequential test(t_2) vs. two arms' pull count ratio $n/(n+m)$.

第4章
最適アーム探索アルゴリズムの提案

4.1 準備

アルゴリズムの説明のために使う記号は2章で定義したものを使う。それに加えて、式(4.1)のように1変数sequential test, 2変数sequential testから計算される信頼区間$U_1(t,\delta), U_2(n,m,\delta)$を用意する。それぞれ、定理1と式(3.13)を根拠にしている。$p(x), h(x)$は3章で定義したものを用いる。3章では報酬分布として1−subgaussianを仮定していたので、(1/2)−subgaussianに修正するための係数(1/2)が掛かっていることに注意。

$$
\begin{aligned}
U_1(t,\delta) &= \frac{1}{2}\arg\min_{0<u}\{\delta\, u\, p(u) \geq h\left(\frac{1}{2}t\, u^2\right)\} \\
U_2(n,m,\delta) &= \frac{1}{2}\arg\min_{0<u}\left\{\omega(n,m,u) \leq \frac{\delta}{3}\right\}
\end{aligned}
\tag{4.1}
$$

ここで、$U_1(t,\delta)$がtについて単調減少であることと、$U_2(n,m,\delta)$がn,mそれぞれについて単調減少であることを仮定する。さらに、それぞれのtについての逆関数を以下のように定義する。$U_2(n,m,\delta)$については$n=m$の場合の逆関数を定義する。

$$
\begin{aligned}
U_1^{-1}(\Delta,\delta) &= \arg\min_{1\leq t}\{U_1(t,\delta) \leq \Delta\} \\
U_2^{-1}(\Delta,\delta) &= \arg\min_{1\leq t}\{U_2(t,t,\delta) \leq \Delta\}
\end{aligned}
\tag{4.2}
$$

定理2の結果から、$U_1^{-1}(\Delta,\delta), U_2^{-1}(\Delta,\delta)$のオーダーは、どちらも$O(\Delta^{-2}\log(\delta^{-1}\log(\Delta^{-2})))$である。

4.2 LIL Stoppingの改良 (LS1, LS2)

関連研究で紹介したLS(LIL Stopping)の原理は、最適アームである可能性が低いアームの数を数えて、その数がアーム数Nに対して、$N-1$個になったら終了するというものである。あるアームが最適アームである確率を調べるという部分は、[4]で提案されているsequential testによって実装されているが、この部分を提案した1変数sequential testと2変数sequential testで置き換えることを考えた。提案したsequential testは、前項で確認したとおり有意差を言うために必要なサンプル数が、[4]で提案されるものも含めた従来に比べて少なくて済むので、これを用いてLSを行えば最適アーム探索としての必要サンプル数も少なくて済むことが期待される。

まず、LSで用いられているsequential testを1変数sequential testを用いたLS1を提案する。全体の第一種エラー率をδとして、個々のアームに割り振る第一種エラー率をδ/Nとする。式

(4.3) のように、アーム 1 の平均が下側信頼区間よりも大きく、その他のアームの平均が上側信頼区間よりも小さいと仮定する。ただし、$U_1(t,\delta)$ は 1 変数 sequential test の信頼区間列である。

$$\begin{aligned} \hat{\mu}_1 &\geq \mu_1 - U_1\left(T_1(t), \frac{\delta}{N}\right) \\ \hat{\mu}_i &\leq \mu_i + U_1\left(T_i(t), \frac{\delta}{N}\right) \quad (\forall i \neq 1) \end{aligned} \tag{4.3}$$

任意のアーム i に関して以下の事象 $e_1(i)$ を考える。

$$\hat{\mu}_i - \hat{\mu}_j + U_1\left(T_i(t), \frac{\delta}{N}\right) + U_1\left(T_j(t), \frac{\delta}{N}\right) \geq 0 \quad (\forall t, \forall j \neq i) \tag{4.4}$$

ここで、$i = 1$ であれば式 (4.3) より任意のアーム $j \neq i$ について、$\hat{\mu}_i - \hat{\mu}_j + U(T_i(t), \delta/N) + U(T_j(t), \delta/N) \geq \mu_i - \mu_j \geq 0$ であるので $e_1(i)$ は真である。対偶を取ると、$e_1(i)$ が偽であれば $i \neq 1$ であるということが分かる。ここで、1 つのアーム b を除いて全てのアーム $i \neq b$ に関して $e_1(i)$ が偽であれば、$b = 1$ であると結論付けることが出来る。また、前提としている式 (4.3) が成り立たない確率は δ 以下である。このように最適アームを選ぶ停止基準を LS1 と呼ぶことにする。

次に、LS1 で行ったことをそのまま 2 変数 sequential test に置き換えた LS2 を提案する。LS1 と違う点は、1 変数 sequential test は個々のアームについて信頼区間列を計算するのに対して、2 変数 sequential test は 2 つのアーム間について信頼区間列を計算する点である。式 (4.5) のように、最適アーム以外の任意のアーム $i \neq 1$ に対して、最適アームとの平均の差が 2 変数 sequential test の信頼区間列に収まっていると仮定する。エラー率 δ はアームごとでは無く 2 つのアーム間の sequential test に割り振るので、それぞれに割り振られるエラー率は $\delta/(N-1)$ とする。$U_2(n, m, \delta)$ は 2 変数 sequential test の信頼区間列である。

$$\hat{\mu}_1 - \hat{\mu}_i \geq \mu_1 - \mu_i - U_2\left(T_1(t), T_i(t), \frac{\delta}{N-1}\right) \quad (\forall i \neq 1) \tag{4.5}$$

$$\tag{4.6}$$

任意のアーム i に関して以下の事象 $e_2(i)$ を考える。

$$\hat{\mu}_i - \hat{\mu}_j + U_2\left(T_i(t), T_j(t), \frac{\delta}{N-1}\right) \geq 0 \quad (\forall t, \forall j \neq i) \tag{4.7}$$

$i = 1$ であれば式 (4.5) より任意のアーム $j \neq i$ について、$\hat{\mu}_i - \hat{\mu}_j + U_2\left(T_i(t), T_j(t), \frac{\delta}{N-1}\right) \geq \mu_i - \mu_j \geq 0$ であるので $e_2(i)$ は真である。後は LS1 と同様に、1 つのアーム b を除いて全てのアーム $i \neq b$ に関して $e_2(i)$ が偽であれば、$b = 1$ であると結論付けることが出来る。また、前提としている式 (4.5) が成り立たない確率は δ 以下である。このように最適アームを選ぶ停止基準を LS2 と呼ぶことにする。

4.3　Successive Elimination の改良

提案した LS1, LS2 はオリジナルの LS と同様、どのバンディッドアルゴリズムにも適用することが出来るが、適当なアルゴリズムを使った場合、最適アームを探し出すための必要サン

プル数が少なくなるとは限らない。関連研究の項で紹介した successive elimination[2] は、LS とほとんど同じ原理に基づいたアルゴリズムで、最適かも知れないアームの集合を持っておき、高確率で最適では無いと分かったアーム ($e_1(i), e_2(i)$ が偽) から順番に排除していき、最後に 1 つだけ残ったアームを最適アームとして出力するというものだが、排除する基準を 1 変数 sequential test、2 変数 sequential test で置き換えることを考えた。successive elimination とほぼ同じだが、具体的にそれぞれのアルゴリズムは以下のようになる。注意する点は、信頼区間列に $U_1(t,\delta), U_2(n,m,\delta)$ を用いている点と、2 変数 sequential test ではエラー率の配分が $\delta/(N-1)$ になる点である。

Algorithm 5 LS1 elimination

input: $N, \delta > 0$, bound function $U_1(t,\delta)$

intialize $t \leftarrow N, A \leftarrow \{1, 2, \cdots, N\}$
for $i = 1$ to N **do**
    pull arm i
end for
while $|A| > 1$ **do**
    $m \leftarrow \arg\max_{i \in A}\{\hat{\mu}_i(t)\}$
    $B \leftarrow \left\{ i \in A \mid \hat{\mu}_{m(t)}(t) - \hat{\mu}_i(t) > U_1\left(T_m(t), \frac{\delta}{N}\right) + U_1\left(T_i(t), \frac{\delta}{N}\right) \right\}$
    $A \leftarrow A \setminus B$
    $I(t) \leftarrow \arg\min_{i \in A}\{T_i(t)\}$
    pull $I(t)$ arm
    $t \leftarrow t + 1$
end while

output: an arm in A (algorithm promise $|A| = 1$)

Algorithm 6 LS2 elimination

input: $N, \delta > 0$, bound function $U_2(n,m,\delta)$

intialize $t \leftarrow N, A \leftarrow \{1, 2, \cdots, N\}$
for $i = 1$ to N **do**
    pull arm i
end for
while $|A| > 1$ **do**
    $m \leftarrow \arg\max_{i \in A}\{\hat{\mu}_i(t)\}$
    $B \leftarrow \left\{ i \in A \mid \hat{\mu}_{m(t)}(t) - \hat{\mu}_i(t) > U_2\left(T_{m(t)}(t), T_i(t), \frac{\delta}{N-1}\right) \right\}$
    $A \leftarrow A \setminus B$
    $I(t) \leftarrow \arg\min_{i \in A}\{T_i(t)\}$
    pull $I(t)$ arm
    $t \leftarrow t + 1$
end while

output: an arm in A (algorithm promise $|A| = 1$)

4.4　必要サンプル数の分析

関連研究の項で紹介したとおり、[2] では successive elimination の sample-complexity が、$O(\sum_{i=2}^{N} \Delta_i^{-1} \log(\Delta_i^{-2}\delta^{-1}N))$ であることが示されている。同様に LS1 elimination, LS2 elimination の sample-complexity を求めてみる。

LS1 elimination の sample-complexity

まず、LS1 elimination の sample-complexity を求める。式 (4.5) が成り立っていたと仮定する。排除されずに残っているアーム集合を A とする。A の中の任意の非最適アーム $i \neq 1$ について、排除されずに残っているということは $e_2(i)$ が真なので、式 (4.4) で $j = 1$ を代入すると

$$\hat{\mu}_i - \hat{\mu}_1 + U_1\left(T_i(t), \frac{\delta}{N}\right) + U_1\left(T_1(t), \frac{\delta}{N}\right) \geq 0 \quad (\forall t) \tag{4.8}$$

変形すると $U_1\left(T_i(t), \frac{\delta}{N}\right) + U_1\left(T_1(t), \frac{\delta}{N}\right) \geq \hat{\mu}_1 - \hat{\mu}_i$ となるので、式 (4.3) と合わせると

$$U_1\left(T_i(t), \frac{\delta}{N}\right) + U_1\left(T_1(t), \frac{\delta}{N}\right) \geq \Delta_i - U_1\left(T_1(t), \frac{\delta}{N}\right) - U_1\left(T_i(t), \frac{\delta}{N}\right) \tag{4.9}$$

$U_1(t, \delta)$ が t について単調減少であることを仮定しているので、それを利用して過大評価すると

$$2U_1\left(\min\{T_1(t), T_i(t)\}, \frac{\delta}{N}\right) \geq U_1\left(T_i(t), \frac{\delta}{N}\right) + U_1\left(T_1(t), \frac{\delta}{N}\right) \geq \frac{\Delta_i}{2} \tag{4.10}$$

式 (4.2) で定義される逆関数を用いて評価すると

$$\min\{T_1(t), T_i(t)\} \leq U_1^{-1}\left(\frac{\Delta_i}{4}, \frac{\delta}{N}\right) \tag{4.11}$$

アルゴリズムが停止していない、つまり $|A| > 1$ のときアーム i がプルされたと仮定すると、アルゴリズムの定義から $T_i(t) \leq T_1(T)$ であるので、式 (4.11) より

$$T_i(t) \leq U_1^{-1}\left(\frac{\Delta_i}{4}, \frac{\delta}{N}\right) \tag{4.12}$$

アーム i がプルされたときのみ $T_i(t)$ が増えるので、常に以下の式が成り立つ。

$$T_i(t) \leq 1 + U_1^{-1}\left(\frac{\Delta_i}{4}, \frac{\delta}{N}\right) \tag{4.13}$$

次に、アーム 1 がプルされたと仮定すると、任意の $i \in A$ に対してアルゴリズムの定義から $T_1(t) \leq T_i(T)$ である。$T_1(t)$ を式 (4.11) で評価するが、i として最も評価が悪くなってしまう場合は $i = 2$ の場合である。そのとき

$$T_i(t) \leq U_1^{-1}\left(\frac{\Delta_2}{4}, \frac{\delta}{N}\right) \tag{4.14}$$

アーム 1 がプルされたときのみ $T_1(t)$ が増えるので、常に以下の式が成り立つ。

$$T_1(t) \leq 1 + U_1^{-1}\left(\frac{\Delta_2}{4}, \frac{\delta}{N}\right) \tag{4.15}$$

以上の結果をまとめると、アルゴリズムが停止していない場合に以下の式が成り立つ。

$$t = \sum_{i=1}^{N} T_i(t) \leq N + U_1^{-1}\left(\frac{\Delta_2}{4}, \frac{\delta}{N}\right) + \sum_{i=2}^{N} U_1^{-1}\left(\frac{\Delta_i}{4}, \frac{\delta}{N}\right) \tag{4.16}$$

これは、アルゴリズムが停止する直前でも成り立つので、アルゴリズムが停止したときにはプル回数はこの評価式から 1 だけ多くなる。しかし、アルゴリズムが停止する直前では、どれかの $i \in A$ で式 (4.12)、または式 (4.14) が成り立つので、最後に増える分の 1 は消去することが出来る。そもそもの仮定である式 (4.3) が成り立たない確率は δ 以下なので、最終的にアルゴリズムの sample-complexity は停止時刻 τ を用いて以下のように書くことが出来る。

$$P\left[\tau > N + U_1^{-1}\left(\frac{\Delta_2}{4}, \frac{\delta}{N}\right) + \sum_{i=2}^{N} U_1^{-1}\left(\frac{\Delta_i}{4}, \frac{\delta}{N}\right)\right] \leq \delta \tag{4.17}$$

この式を見ると、アルゴリズムは $1-\delta$ 以上の確率で式 (4.16) の右辺以下の時刻で停止するが、δ 以下の確率で停止時刻が無限に大きくなるという可能性が残るので、期待値 $E[\tau]$ は押さえ込めていない。$U_1^{-1}(\Delta, \delta)$ のオーダーは $O(\Delta^{-2}\log(\delta^{-1}\log(\Delta^{-2})))$ であるので、sample-complexity のオーダーは $1-\delta$ 以上の確率で以下のようになる。

$$P\left[\tau = O\left(\sum_{i=1}^{N} \frac{1}{\Delta_i^2}\log\left(\frac{N}{\delta}\log\left(\frac{1}{\Delta_i^2}\right)\right)\right)\right] \geq 1-\delta \tag{4.18}$$

LS2 elimination の sample-complexity

次に、LS2 elimination の sample-complexity を求める。証明の流れは信頼区間列として $U_1(t, \delta)$ の代わりに $U_2(n, m, \delta)$ を使うことと、エラー率の配分を $\delta/(N-1)$ とすること以外はほとんど同じである。式 (4.5) が成り立っていたと仮定する。排除されずに残っているアーム集合を A とする。A の中の任意の非最適アーム $i \neq 1$ について、排除されずに残っているということは $e_1(i)$ が真なので、式 (4.7) で $j = 1$ を代入すると

$$\hat{\mu}_i - \hat{\mu}_1 + U_2\left(T_i(t), T_1(t), \frac{\delta}{N-1}\right) \geq 0 \quad (\forall t) \tag{4.19}$$

式 (4.5) と合わせると

$$U_2\left(T_1(t), T_i(t), \frac{\delta}{N-1}\right) \geq \Delta_i - U_2\left(T_1(t), T_i(t), \frac{\delta}{N-1}\right) \tag{4.20}$$

$U_2(n, m, \delta)$ が n, m それぞれについて単調減少であることを仮定しているので、それを利用して過大評価すると

$$U_2\left(\min\{T_1(t), T_i(t)\}, \min\{T_1(t), T_i(t)\}, \frac{\delta}{N-1}\right) \geq U_2\left(T_1(t), T_i(t), \frac{\delta}{N-1}\right) \geq \frac{\Delta_i}{2} \tag{4.21}$$

式 (4.2) で定義される逆関数を用いて評価すると

$$\min\{T_1(t), T_i(t)\} \leq U_2^{-1}\left(\frac{\Delta_i}{2}, \frac{\delta}{N-1}\right) \tag{4.22}$$

式 (4.11) と同じ形の式にすることが出来たので、後は LS1 と同様に考えていくと、最終的にアルゴリズムの sample-complexity は停止時刻 τ を用いて以下のように書くことが出来る。

$$P\left[\tau > N + U_2^{-1}\left(\frac{\Delta_2}{2}, \frac{\delta}{N-1}\right) + \sum_{i=2}^{N} U_2^{-1}\left(\frac{\Delta_i}{2}, \frac{\delta}{N-1}\right)\right] \leq \delta \tag{4.23}$$

LS1 と同様に、アルゴリズムは $1-\delta$ 以上の確率で式 (4.23) の時刻で停止するが、δ 以下の確率で停止時刻が無限に大きくなるという可能性が残るので、期待値 $E[\tau]$ は押さえ込めていない。$U_2^{-1}(\Delta, \delta)$ のオーダーは $O(\Delta^{-2}\log(\delta^{-1}\log(\Delta^{-2})))$ であるので、LS1 と同様、sample-complexity のオーダーは $1-\delta$ 以上の確率で以下のようになる。

$$P\left[\tau = O\left(\sum_{i=1}^{N} \frac{1}{\Delta_i^2}\log\left(\frac{N}{\delta}\log\left(\frac{1}{\Delta_i^2}\right)\right)\right)\right] \geq 1-\delta \tag{4.24}$$

4.5　新しい最適アーム探索アルゴリズムの提案

2 変数 sequential test を用いた新しい最適アーム探索アルゴリズム LS2 Highest P Algorithm(LS2 HP) を提案するが、それを考えるに至った動機が 2 つある。

一つ目はエラー率 δ の配分の問題である。提案した LS1 elimination, LS2 elimination は、各アームまたは各アーム対に対してエラー率を配分しているが、そのときの配分比率は均等に δ/N または $\delta/(N-1)$ である。また、sample-complexity の分析によると、$\log(N/\delta)$ という項が入っているので、アーム数 N が大きくなるにつれてこの項の影響が増えていく。この影響を減らすために、エラー率 δ を不均等に配分するということを考えた。ただし、アルゴリズム実行前の時点では全てのアームは対等なので、エラー率 δ をどのように不均等に割り振れば良いか分からない。そのため、アルゴリズムを開始してアームをプルしていく中で、エラー率 δ の配分比率を動的に変化させていくという方法を取る。

二つ目は、アルゴリズムが停止しない可能性についてである。successive elimination, lil'UCB, 提案手法である LS1 elimination と LS2 elimination は、高い確率で有限のサンプル数でアルゴリズムが停止することを保証できるが、ごくまれにアルゴリズムが停止しない可能性についてなんの保証も出来ない。発生する確率が低いので実験で再現することは難しいが、実際に停止しない例を考えることは出来る。例えば、$N = 3$ の問題で、アーム 1 の報酬期待値が 1、アーム 2, 3 の報酬期待値が両方とも 0.9 だったケースを考える。この問題で、最初にアーム 1 をプルしたときに運悪く物凄く低い報酬が得られたとする。その場合、アーム 1 は 2 度とプルされることは無く、その後はアーム 2, 3 だけがプルされる。しかし、アーム 2, 3 の報酬期待値は等しいので、サンプル数をいくら増やしてもアルゴリズムがその差を識別できることは無い。同じ sample-complexity を持つアルゴリズムなら、停止しないものよりも停止するものの方が、ロバスト性や信頼性の観点から考えて使いやすい。アルゴリズムを停止させるために、successive elimination のようにアームを排除していくのではなく、後述のようにアームごとに評価値 $q_i(t)$ を割り当て、その評価値 $q_i(t)$ が最も高いアームをプルしていくという方式を考えた。$q_i(t)$ はプルすると減少していくように設計してあるので、t が十分大きくなれば全てのアームがプルされて $q_i(t)$ が減少していく。従って、2 度とプルされないアームが生じることは無い。

$q_i(t)$ は主に $p_i(t)$ によって決まるので、近似的には最も高い $p(i)$ のアームをプルしていくアルゴリズムということで、LS2 Highest P Algorithm(LS2 HP) と呼ぶことにする。残念ながら sample-complexity を分析することはできなかったが、第一種エラー率が δ 以下に収まるということと、アルゴリズムが必ず停止するということは証明することが出来た。

4.5.1　アルゴリズム

まず、各アーム i に対して p 値 $p_i(t)$ を以下のように定義する。これは、平均が最大のアーム m と任意のアーム $i \neq m$ との間で、平均に差があるかどうかを 2 変数 sequential test で検定したときの p 値であり、おおまかに $p_i(t)$ が大きいほど m と i は近いという傾向がある。

$$p_i(t) = \inf\{0 < p \leq 1 \ : \ |\hat{\mu}_{m(t)}(t) - \hat{\mu}_i(t)| > U_2(T_{m(t)}(t), T_i(t), p)\} \tag{4.25}$$

さらに、$q_i(t)$ をアルゴリズムのパラメータ $r > 0$ を用いて以下のように定義する。

$$q_i(t) \leftarrow p_i(t)(T_{m(t)}(t)/T_i(t))^r \tag{4.26}$$

アルゴリズムを以下に示す。

Algorithm 7 LS2 Highest P Algorithm

input: $N, 0 < \delta \leq 1, 0 < r, 0 < \beta < 1$

intialize $t \leftarrow N, \theta_i \leftarrow 0$
for $i = 1$ to N **do**
  pull arm i
end for
loop
  $m \leftarrow \arg\max_{1 \leq i \leq N} \{\hat{\mu}_i(t)\}$
  calclate p_i
  $q_i \leftarrow p_i (T_m / T_i)^r$
  $j \leftarrow \arg\max_{i \neq m} \{q_i\}$
  if $q_j \leq \theta_j$ **then**
    break loop
  end if
  if $q_j \leq \beta\delta$ and $\theta_j = 0$ **then**
    $\theta_j \leftarrow \beta\delta$
    $\delta \leftarrow \delta - \theta_j$
  end if
  if $T_j \leq T_m$ **then**
    $I(t) \leftarrow j$
  else
    $I(t) \leftarrow m$
  end if
  pull arm $I(t)$
  $t \leftarrow t + 1$
end loop

output: arm m

4.5.2　動的なエラー率配分

エラー率を均等に配分するとアーム数 N が増えるに従って、一つ一つのアーム対に割り当てられるエラー率は減っていく。それを解決するために、動的にエラー率を配分することにした。それぞれのアーム対に割り当てられたエラー率を $\{\theta_i(t)\}_{i=2,3,\cdots,N}$ とする。それぞれの $\theta_i(t)$ は単調増加で $\theta_i(1) = 0$ であり、図 4.1 のように合計について、$\sum_{i=2}^{N} \theta_i(t) \leq \delta$ とする。図 4.2 のように、$q_i(t)$ がスレッショルド $\beta \cdot \delta$ を下回るのを待ち構えていて、あるアーム i で $q_i(t) \leq \beta \cdot \delta$ となって下回った場合に、$\theta_i(t+1) = \beta \cdot \delta$ とする。$0 < \beta < 1$ は δ をどのくらいの割合ずつ配分していくかを決めるパラメータである。$q_i(t)$ はアーム i のプル回数が増えるにつれて減少する傾向があるので、次の時刻 $t+1$ でも $q_i(t+1) \leq \theta_i(t+1) = \beta \cdot \delta$ となる可能性が高い。つまり、このようにスレッショルド $\beta \cdot \delta$ を下回ったもののみにエラー率を配分すると、高確率で次の時刻 $t+1$ でアルゴリズムが終了することが期待できる。ただし、その確率の定量的な分析は出来ていない。δ からは配分した分を差し引いて $\delta \leftarrow \delta - \theta_i(t+1)$ として、また新しい $\beta \cdot \delta$ で待ち構える。従って、割り当てられるエラー率は割り当てた順に、$\beta \cdot \delta, \beta(1-\beta) \cdot \delta, \beta(1-\beta)^2 \cdot \delta, \cdots$

となる。

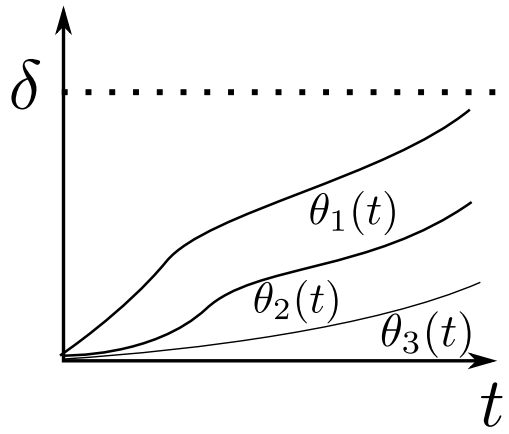

図 4.1: $\theta_i(t)$: division of type I error rate.

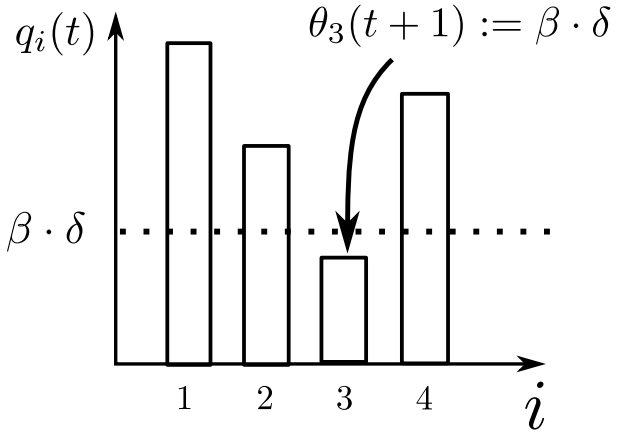

図 4.2: Delayed assignment of type I error rate.

4.5.3　停止基準が正しいことの証明

アルゴリズムの第一種エラー率が δ 以下に収まるということと、アルゴリズムが必ず停止するということを証明する。

準備

t が関わってこないときは (t) を省略して書く。全てのアーム $i > 1$ に対して式 (4.27) のように $\{\omega_i(t)\}$ を定義する。

$$\omega_i(t) = \sup\{\omega \ : \ \hat{\mu}_1(t) - \hat{\mu}_i(t) \geq \Delta_i - U_2(T_1(t), T_i(t), \omega)\} \tag{4.27}$$

各アームの停止判定を行う基準 (エラー率) を $\{\theta_i(t)\}_{i=2,3,\cdots,N}$ とする。ただし、各 $\theta_i(t)$ は時刻 $t-1$ までの情報によって決まるということと、$\sum_{i=2}^{N} \sup_{0<t}\{\theta_i(t)\} \leq \delta$ が成り立つことを仮定する。この仮定は、前述の動的なエラー率配分に従う限り成り立つ。ω の定義と 2 変数 sequential test の定理 3 から

$$\mathbb{P}[\exists t > 0, \quad \omega_i(t) \leq \theta_i(t)] \leq \sup_{0<t}\{\theta_i(t)\} \tag{4.28}$$

Step 1

ここで、$m \neq 1$ と仮定する。つまり、経験的最適アームが本当の最適アームでは無いと仮定する。このとき $i = m$ についての式 (4.27) と $\hat{\mu}_m - \hat{\mu}_1 \geq 0$ であることから以下の式が成り立つ。

$$\begin{aligned} U_2(T_1, T_m, \omega_m) &\geq \Delta_m + \hat{\mu}_m - \hat{\mu}_1 && (4.29) \\ &\geq \Delta_m + U_2(T_m, T_1, p_1) && (4.30) \end{aligned}$$

$\Delta_m \geq 0$ なので、$U_2(t_1, t_2, \delta)$ が δ に対して単調減少することを利用すると $\omega_m \leq p_1$ が成り立つ。ここで、任意のアーム $i \neq m$ がプルされたと仮定すると、アルゴリズムの定義から以下の式が成り立つ。

$$\frac{p_i}{T_i^r} \geq \frac{p_1}{(\min T_1, T_m)^r} \geq \frac{\omega_m}{T_m^r} \tag{4.31}$$

変形すると以下の式が成り立つ。

$$\omega_m \leq p_i \frac{T_m^r}{T_i^r} \tag{4.32}$$

式 (4.32) の右辺を用いて以下の式のように停止基準を作る。最初の等号は、Step 2 で確認するがこの停止基準で停止する確率が 1 ということによって成り立つ。全体の第一種エラー率が δ 以下となることが分かる。

$$\mathbb{P}\left[m(t) \neq 1 | \exists t \geq 1,\ \ p_{I(t)}(t)\frac{T_{m(t)}^r(t)}{T_{I(t)}^r(t)} \leq \theta_{m(t)}(t) \text{ and } m(t) \neq I(t)\right] \tag{4.33}$$

$$= \mathbb{P}\left[\exists t \geq 1,\ \ p_{I(t)}(t)\frac{T_{m(t)}^r(t)}{T_{I(t)}^r(t)} \leq \theta_{m(t)}(t); m(t) \neq 1 \text{ and } m(t) \neq I(t)\right] \tag{4.34}$$

$$\leq \mathbb{P}\left[\exists t \geq 1, \omega_{m(t)}(t) \leq \theta_{m(t)}(t); m(t) \neq 1 \text{ and } m(t) \neq I(t)\right] \tag{4.35}$$

$$\leq \mathbb{P}\left[\exists i \geq 2, \exists t \geq 1, \omega_i(t) \leq \theta_i(t)\right] \tag{4.36}$$

$$\leq \sum_{i=2}^{N} \sup_{0<t}\{\theta_i(t)\} \leq \delta \tag{4.37}$$

Step 2

次に Step 1 の証明を完成するために、この停止基準で停止する確率が 1 ということを示す。アルゴリズムが停止していないと仮定する。すると任意のアーム $i \neq m$ に対して以下の式が成り立つ。

$$\theta_m \leq p_i \frac{T_m^r}{T_i^r} \tag{4.38}$$

$m = 1$ の場合、任意のアーム $i \neq m$ がプルされたと仮定すると $p_i \leq 1$ より

$$\theta_m^{1/r} T_i \leq T_1 \tag{4.39}$$

次にアーム 1 がプルされると仮定したら

$$T_1 \leq \min_{1<i}\{T_i\} \tag{4.40}$$

$\theta_m(t) > 0$ は t について単調増加を仮定するので、式 (4.39), 式 (4.40) から時刻 t が増えるに従って全てのアームのプル回数 T_i は増えていくことが分かる。ここで、ω, p_i の定義と $\hat{\mu}_1 - \hat{\mu}_i \geq 0$ より以下の式が成り立つ。

$$2U_2(T_1, T_i, \min\{\omega_i, p_i\}) \geq U_2(T_1, T_i, \omega_i) + U_2(T_1, T_i, p_i) \geq \Delta_i \tag{4.41}$$

全てのアームの T_i が増えていくとすると、この式が成り立つためには $min\{\omega_i, p_i\}$ が小さくなる必要がある。$\omega_i > 0$ であるので、t を十分大きくすれば全てのアームの p_i を任意に小さくすることが出来る。すると、式 (4.38) に矛盾する。

Step 3

次に、$m \neq 1$ の場合を考える。任意のアーム $i \neq m$ がプルされたと仮定したら。

$$\frac{p_i}{T_i^r} \geq \frac{p_1}{(\min T_1, T_m)^r} \tag{4.42}$$

$$\geq \frac{\omega_m}{(\min T_1, T_m)^r} \tag{4.43}$$

変形して

$$\omega_m^{1/r} T_i \leq \min T_1, T_m \tag{4.44}$$

アーム m がプルされたとしたら、$T_m \leq T_i$ である。アーム 1 がプルされたとしたら、$T_1 \leq T_m$ である。従って、t が十分大きくなれば全てのアームのプル回数が増えていく。そうすると、ω の定義から成り立つ以下の式がいつかは満たされなくなるので矛盾する。

$$U_2(T_1, T_m, \omega_m) \quad \geq \quad \Delta_m \tag{4.45}$$

Step 2, 3 を合わせると、十分大きい t で必ずアルゴリズムが停止することが分かる。以下のように停止する確率が 1 であることが分かったので、Step 1 の式 (4.37) が正しいことが分かった。

$$\mathbb{P}\left[\exists t \geq 1, \;\; p_{I(t)}(t)\frac{T^r_{m(t)}(t)}{T^r_{I(t)}(t)} \leq \delta_{m(t)}(t) \text{ and } m(t) \neq I(t)\right] = 1 \tag{4.46}$$

第5章
シミュレーションによる評価

5.1 概要

提案したLS1, LS2の停止基準としての有効性、LS1 elimination, LS2 elimination, LS2 HPアルゴリズムの有効性を、シミュレーションによる従来手法との比較によって評価する。評価はsample-complexityを比較することによって行う。アルゴリズムのsample-complexityは、アームの報酬期待値分布、報酬分布、アーム数、エラー率に依存して変わることが予想されるが、それらを様々に変えて評価する。

5.2 実験条件

5.2.1 評価尺度

比較に使うsample-complexityの計算方法を説明する。まず、複数回最適アーム探索を行って、アルゴリズムが停止した時刻の平均値を求める。具体的には最大時刻Tを用意して、時刻Tまでに停止しなかった場合の時刻は$t = T$として平均値を求める。アルゴリズムが停止する時刻は問題の難易度が上がるにつれて増えていくが、異なる難易度の問題に対するsample-complexityを比較しやすくするために、関連研究の項で紹介した問題の難しさを表す指標H_1を用いて、時刻の平均値を正規化したものをsample-complexityとする。また、一部の実験では停止時刻のばらつきを見るために、正規化した停止時刻の標準偏差も評価に用いる。

5.2.2 アーム報酬期待値分布と報酬分布について

アームの報酬期待値分布として以下の4つ(3種類)を用いる。

- α-parameterで特徴付けされた分布($\alpha = 0.3, 0.6$)
- 最適アームの報酬期待値のみ非ゼロ(1-sparse)
- 判別を難しくした離散分布(difficult)

α-parameterで特徴付けされた分布は文献[19]で提案されているものであり、パラメータ$\alpha > 0$に対してアームの報酬分布$\{\Delta_i\}_{i=2,3,\cdots,N}$を式(5.1)とするものである。$\alpha$が大きいと問題の難易度は上がり、小さいと問題の難易度は下がるが、今回は$\alpha = 0.3, 0.6$の場合の2つの分布を用いる(図5.1)。1-sparseは$\alpha = 0.3, 0.6$と、1-sparseの場合の報酬分布は分散0.25のガウス分布とする。これら3つの分布は文献[4]のシミュレーション実験で用いられた分布と同じである。残りの判別を難しくした離散分布(difficult)は文献[4]では用いられていない分布で、最適アームの報酬分布を$\mathbb{P}[X = 0] = 0.8, \mathbb{P}[X = 1] = 0.2$として、その他のアームの報酬分布を

$\mathbb{P}[X = 0.01] = 1$ としたものである。期待値は最適アームが 0.2、非最適アームが 0.01 となる。最初の方は高確率で非最適アームの方の平均が高くなるので、判別が難しいことが期待される。また、difficult は報酬分布がガウス分布では無いので、報酬分布がガウス分布では無いときに sample-complexity がどう変わるかも部分的に調べることが出来る。

H_1 はアームの報酬期待値分布から式 (2.6) で計算することが出来るが、α-parameter で特徴付けされた分布の場合はアーム数 N と α に対して式 (5.2) のような関係がある [19]。これに従うと、$\alpha = 0.3$ のとき $H_1 = O(N)$ であり $\alpha = 0.6$ のとき $H_1 = O(N^{1.2})$ である。また、1-sparse と difficult の場合は $H_1 = O(N)$ である。

$$\Delta_i = \left(\frac{i-1}{N-1}\right)^{\alpha} \quad (1 < i) \tag{5.1}$$

$$H_1 = \begin{cases} O(N) & (\alpha < 1/2) \\ O(N \log(N)) & (\alpha = 1/2) \\ O(N^{2\alpha}) & (\alpha > 1/2) \end{cases} \tag{5.2}$$

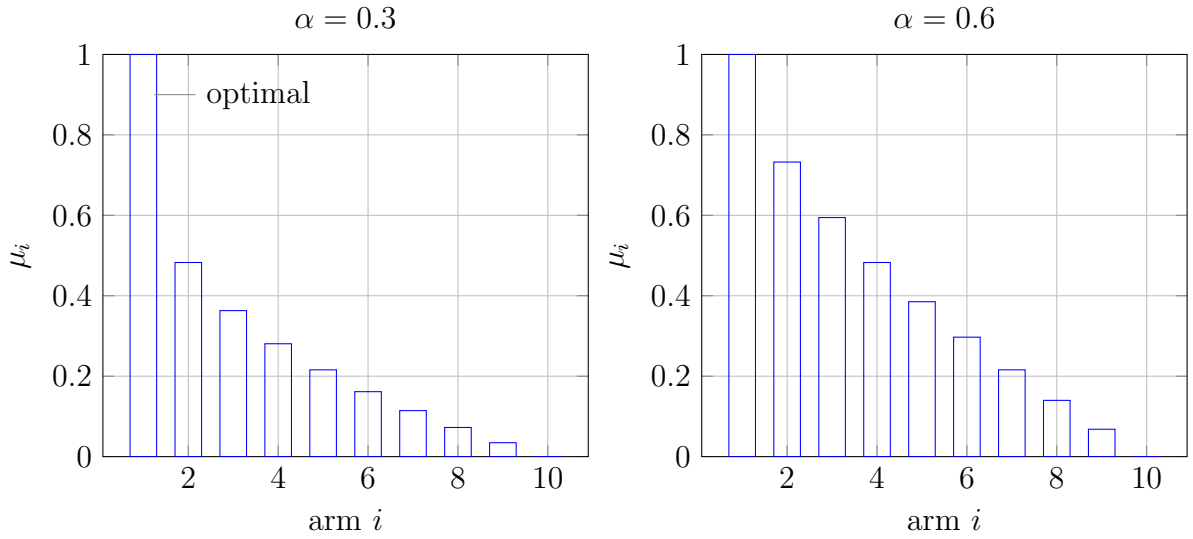

図 5.1: Distribution of the expectation of arm rewards μ_i. $(N = 10)$

5.2.3　比較するアルゴリズムと停止基準

比較対象のアルゴリズムを表 5.1 に記した。アルゴリズムは、元から最適アーム探索を行うように設計されたものと、元々はリグレット最小化等の別の目的で作られたが LS と組み合わせることによって最適アーム探索を行えるようになったものに分けることが出来る。表の中で algorithm specific stopping criteria が○になっているものは、最適アーム探索として設計されたものである。uniform は全てのアームを均等にプルするアルゴリズムであり、最も単純なアルゴリズムであるということで比較対象に加える。UCB1[7] は、リグレット最小化アルゴリズムとして有名なものなので、これも比較対象に加える。関連研究で紹介したように LUCB は元々は最適アーム探索アルゴリズムでは無いが、LS と組み合わせたときに良い性能を出すということ

が、[4] で示されているので比較対象とする。successive elimination[16] は、LS1 elimination, LS2 elimination の改良元となったアルゴリズムなので、比較のために加える。lil'UCB[4] は最適な sample-complexity を持つ手法である。さらに、提案手法 LS1 elimination, LS2 elimination, LS2 Highest P(LS2 HP) を比較する。Exp-Gap は lil'UCB と同じように最適な Sample-Complexity を持つアルゴリズムだが、文献 [4] によると、実際の性能が他の手法と比べて 100-1000 倍悪いということが示されているので、今回は比較対象から省いた。

表 5.1: Algorithms used in experiments.

algorithm	algorithm specific stopping criteria
uniform	×
UCB1[7]	×
LUCB[5]	×
successive elimination[2]	○
lil'UCB[4]	○
LS1 elimination(proposed)	○
LS2 elimination(proposed)	○
LS2 HP(proposed)	○

また、比較する停止基準一覧を以下に示す。

- LS(original)
- LS1
- LS2
- specific(個々のアルゴリズム特有の停止基準)

LS(original) は文献 [4] で提案されている元々の LS である。LS1, LS2 は 4 章で提案した手法である。元々最適アーム探索アルゴリズムとして設計されたものが持つ停止基準を specific と呼ぶことにする。

5.2.4　アルゴリズムと停止基準のおおまかな比較

全てのアルゴリズムと停止基準の組み合わせを全ての実験で行うと、実験の量が多くなりすぎてしまうので、ここで限られた実験設定で全ての組み合わせを実験して、有望なものを選別する。実験条件を表 5.2.4 に示す。

表 5.2: Experimental settings 1.

T	N	δ	distribution
$1000 \cdot H_1$	10	0.1	$\alpha = 0.3, \alpha = 0.6$, 1-sparse, difficult

5.2.5　アーム数 N への依存性

4 章で分析したように、LS1 elimination, LS2 elimination の sample-complexity はアーム数 N が増えるに少しずつ悪化することが予想される。実際にどの程度のものかを調べるために、

他の条件を固定した上で N を変化させながら sample-complexity の変化を見る。実験条件を表 5.2.5 に示す。

表 5.3: Experimental settings 2.

T	N	δ	distribution
$1000 \cdot H_1$	2, 3, 6, 12, 25, 50, 100	0.1	$\alpha = 0.3, \alpha = 0.6$, 1-sparse, difficult

5.2.6　エラー率 δ への依存性

successive elimination, lil'UCB, LS1 elimination, LS2 elimination に於いて、sample-complexity のオーダーはエラー率 δ に対して $O(H_1 \log(1/\delta))$ である。従って、横軸 δ、縦軸 sample-complexity の片対数プロットは直線になることが予想される。また、この直線の傾きを調べると sample-complexity の中で $H_1 \log(1/\delta)$ に掛かる係数を推定することが出来る。特に、理論的な分析の出来ていない LS2 HP についても、実験的にこの係数を推定することが出来るのは興味深い。他の条件を固定した上で δ を変化させながら sample-complexity の変化を見る。実験条件を表 5.2.6 に示す。

表 5.4: Experimental settings 3.

T	N	δ	distribution
$1000 \cdot H_1$	10	$10^{-1}, 10^{-2}, \cdots, 10^{-10}$	$\alpha = 0.3, \alpha = 0.6$, 1-sparse, difficult

5.2.7　Sample-complexity の分布

前述のように、successive elimination, lil'UCB, LS1 elimination, LS2 elimination はほとんどの場合、それぞれ理論的に保証されたサンプル数で停止するが、ごくまれにサンプル数が ∞ になってしまう可能性がある。わずかな確率なので実験的に捕捉できるか分からないが、捕捉できないにしても sample-complexity のばらつきの程度は、アルゴリズムの信頼性に関わってくるので、ここでは sample-complexity の分布を見てみる。実験条件を表 5.2.6 に示す。

表 5.5: Experimental settings 4.

T	N	δ	distribution
$100 \cdot H_1$	10	0.1	$\alpha = 0.3, \alpha = 0.6$, 1-sparse, difficult

5.3　実験結果

5.3.1　アルゴリズムと停止基準のおおまかな比較

アーム期待値分布を $\alpha = 0.3, \alpha = 0.6$, 1-sparse, difficult としたときの結果を順に表 5.6, 表 5.7, 表 5.8, 表 5.9 に示す。表には、100 回アルゴリズムを実行して得られた正規化 sample-complexity の平均値と標準偏差を記した。N/A はアルゴリズム特有の停止基準が存在しないことを表す。

まず、4 つの分布に全てについて提案手法である LS2 HP + specific の sample-complexity が最も小さい。従来手法の中で比較的 sample-complexity が小さいものは、文献 [4] の結果と同じく、LUCB + LS(original) であるが、これと LS2 HP との sample-complexity の比は 5〜6 になっている。この実験条件に於いては、従来手法と比べて sample-complexity を 5〜6 倍減らすことが出来た。停止基準に注目すると、LS(original) と LS2 を使ったときの sample-complexity の比が最も小さいのは LUCB で 3〜4 になっている。最も小さいときでさえ比が 3〜4 になっているということは、LS(original) から LS2 に変えたときの停止基準の寄与は、sample-complexity を 3〜4 倍減らす程度であるといえる。他の実験に使う組み合わせとして、LS1 elimination + specific, LS2 elimination + specific, LS2 HP + specific, LUCB + LS2 を使うことにする。

表 5.6: Sample-complexity mean and standard deviation. ($\alpha = 0.3, N = 10, \delta = 0.1, T = 1000 \cdot H1$)

	LS(original)	LS1	LS2	specific
uniform	112 ± 22.9	31.9 ± 11.3	22.3 ± 9.16	N/A
UCB1	305 ± 46.4	20.1 ± 4.83	16.8 ± 4.41	N/A
LUCB	46.5 ± 6.46	15.8 ± 3.82	13.4 ± 2.54	N/A
succ elimination	108 ± 26.2	29.7 ± 9.75	21.9 ± 8.56	724 ± 376
lil'UCB	64.3 ± 10.9	17.5 ± 4.34	15.9 ± 5.57	362 ± 21.9
LS1 elimination	724 ± 393	16.9 ± 4.44	13.7 ± 4.02	16.9 ± 4.44
LS2 elimination	963 ± 149	18.1 ± 12.7	10.6 ± 3.14	10.6 ± 3.14
LS2 HP	46.5 ± 6.07	13.7 ± 2.76	9.66 ± 3.02	8.14 ± 2.52

表 5.7: Sample-complexity mean and standard deviation. ($\alpha = 0.6, N = 10, \delta = 0.1, T = 1000 \cdot H1$)

	LS(original)	LS1	LS2	specific
uniform	198 ± 42.7	56.1 ± 21.1	37.5 ± 19.1	N/A
UCB1	404 ± 86.0	27.1 ± 9.62	19.6 ± 6.74	N/A
LUCB	57.2 ± 8.49	19.6 ± 6.14	14.6 ± 4.51	N/A
succ elimination	136 ± 50.9	49.3 ± 20.1	34.2 ± 16.9	416 ± 344
lil'UCB	85.2 ± 10.9	22.7 ± 7.88	23.2 ± 9.50	346 ± 27.7
LS1 elimination	870 ± 272	18.3 ± 5.43	15.2 ± 4.98	18.3 ± 5.43
LS2 elimination	988 ± 86.9	33.9 ± 106	12.1 ± 4.34	12.1 ± 4.34
LS2 HP	55.9 ± 8.18	15.5 ± 4.29	10.8 ± 3.97	8.93 ± 3.63

5.3.2　アーム数 N への依存性

アーム数 N を変化させたときの sample-complexity の変化を図 5.2 に示す。100 回アルゴリズムを実行して得られた正規化 sample-complexity の平均値をプロットした。プロットの精度についてだが、期待値の信頼区間幅を相対値にしたもの (標準偏差 / 平均値 / $\sqrt{100}$) は最大で 5.3% であった。全体的に N が小さいとき ($N < 10$) に sample-complexity が高い傾向があ

表 5.8: Sample-complexity mean and standard deviation. (1-sparse, $N = 10, \delta = 0.1, T = 1000 \cdot H1$)

	LS(original)	LS1	LS2	specific
uniform	64.0 ± 9.48	20.0 ± 5.31	15.7 ± 4.87	N/A
UCB1	145 ± 16.8	15.2 ± 3.02	13.1 ± 2.91	N/A
LUCB	44.2 ± 4.34	14.1 ± 2.93	13.0 ± 2.54	N/A
succ elimination	64.1 ± 9.26	21.9 ± 6.29	14.9 ± 5.22	801 ± 376
lil'UCB	47.5 ± 6.04	15.0 ± 3.53	12.8 ± 3.28	375 ± 23.5
LS1 elimination	768 ± 370	15.1 ± 4.31	13.9 ± 4.31	15.1 ± 4.31
LS2 elimination	974 ± 136	67.6 ± 216	10.4 ± 3.28	10.4 ± 3.28
LS2 HP	48.2 ± 6.60	12.5 ± 3.40	9.02 ± 2.72	7.40 ± 3.32

表 5.9: Sample-complexity mean and standard deviation. (difficult, $N = 10, \delta = 0.1, T = 1000 \cdot H1$)

	LS(original)	LS1	LS2	specific
uniform	53.3 ± 6.56	15.9 ± 3.48	11.0 ± 2.92	N/A
UCB1	105 ± 2.69	11.2 ± 1.54	9.66 ± 1.35	N/A
LUCB	41.6 ± 2.49	11.9 ± 1.45	10.4 ± 1.21	N/A
succ elimination	54.3 ± 5.72	15.2 ± 3.42	10.4 ± 3.14	819 ± 362
lil'UCB	40.0 ± 3.38	11.1 ± 1.99	9.40 ± 2.40	378 ± 5.97
LS1 elimination	593 ± 433	15.0 ± 3.63	9.84 ± 2.69	15.0 ± 3.63
LS2 elimination	922 ± 232	12.8 ± 5.13	10.6 ± 3.19	10.6 ± 3.19
LS2 HP	53.4 ± 6.27	15.0 ± 3.18	11.1 ± 3.19	7.98 ± 2.74

る。LS1 elimination, LS2 elimination, LUCB + LS2 の 3 手法は、N が大きくなるにつれて $(10 < N)$ 少しずつ sample-complexity が上がっている。反対に LS2 HP は N が大きくなるにつれて sample-complexity が下がっている。

5.3.3　エラー率 δ への依存性

エラー率 δ を変化させたときの sample-complexity の変化を図 5.3 に示す。100 回アルゴリズムを実行して得られた正規化 sample-complexity の平均値をプロットした。プロットの精度についてだが、期待値の信頼区間幅を相対値にしたもの (標準偏差 / 平均値 / $\sqrt{100}$) は最大で 3.9% であった。

5.3.4　Sample-complexity の分布

Sample-complexity の分布を図 5.4 に示す。1000 回アルゴリズムを実行して H_1 で正規化した停止時刻の分布を求めた。ごくまれにアルゴリズムが停止しないということは再現できなかった。手法間で分布を比較すると LS2 HP は difficult での LUCB + LS2 を除いて全ての手法よりも左側に来ている

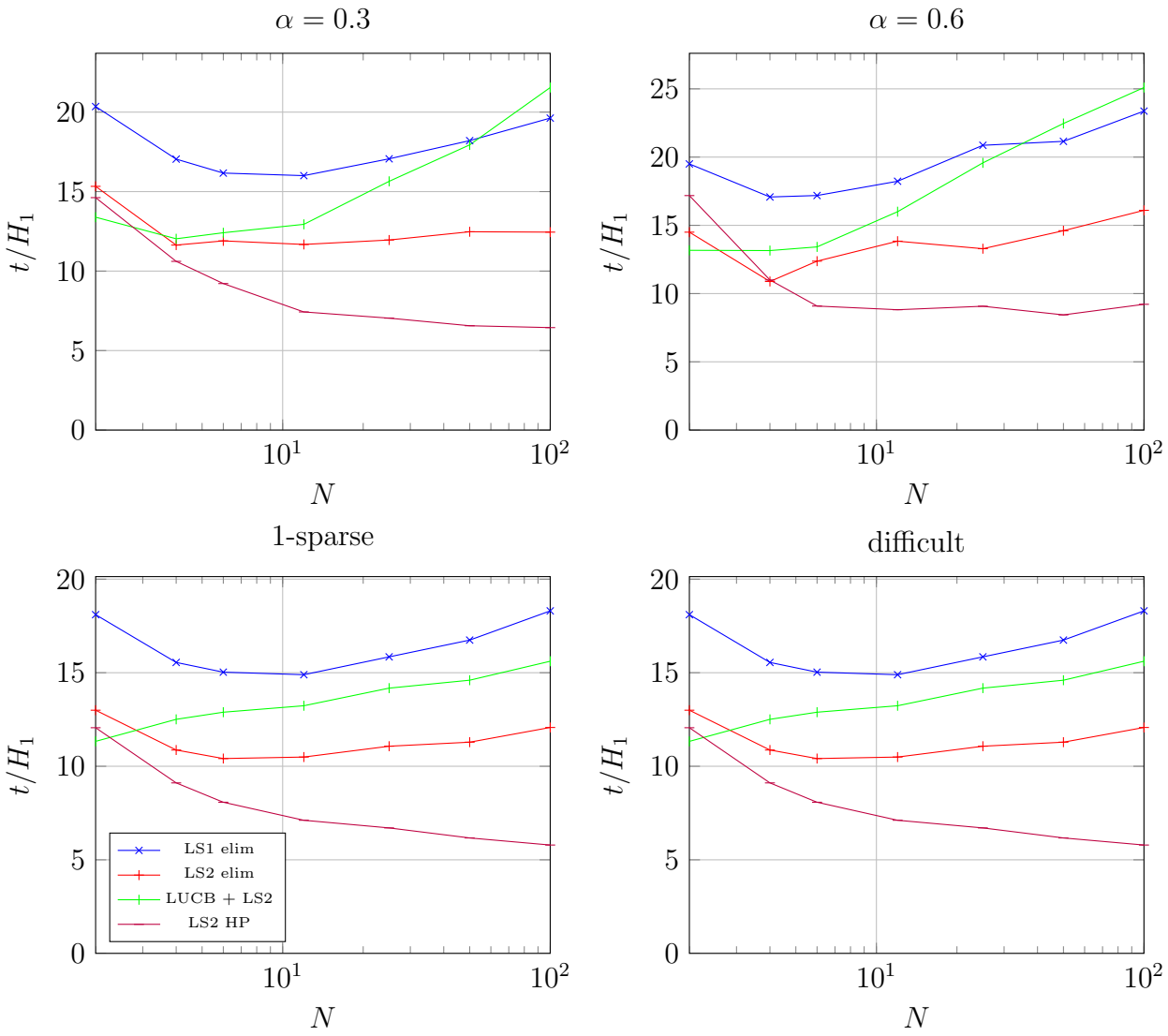

図 5.2: Sample-complexity vs. N, ($\delta = 0.1$)

5.4　考察

おおまかな比較の結果によると、提案手法 LS2 HP が従来手法よりも 5〜6 倍 sample-complexity を削減できているので、提案手法の有効性を示す結果になった。また、LS(original) から LS2 に変えた場合の sample-complexity 削減効果は 3〜4 倍だったので、停止基準だけでも従来手法と比べて改善できていると言える。また LS2 は、組み合わせるアルゴリズムに依らずに sample-complexity が小さいのでロバストだと言える。

sample-complexity のアーム数 N への依存性について、提案手法 LS2 HP は N が増えるにつれて sample-complexity が下がるという結果が出た。これは、エラー率を N で配分しなくても良いということ以上に下がっているのかも知れない。LS1 elimination、LS2 elimination について sample-complexity の理論的な分析から推測すると、N が増えるにつれて少しずつ sample-complexity が増えるという予想が立つが、実験結果もそれに従う結果となった。アーム数が少ない ($N < 10$) 部分で sample-complexity が高くなってしまうことは、どの手法でも観測された。この原因は、H_1 による問題の難易度見積もりが N が小さいときに易しく見積もりすぎている、もしくは N が大きいときに難しく見積もりすぎている、もしくはどの手法も N が小さいとき

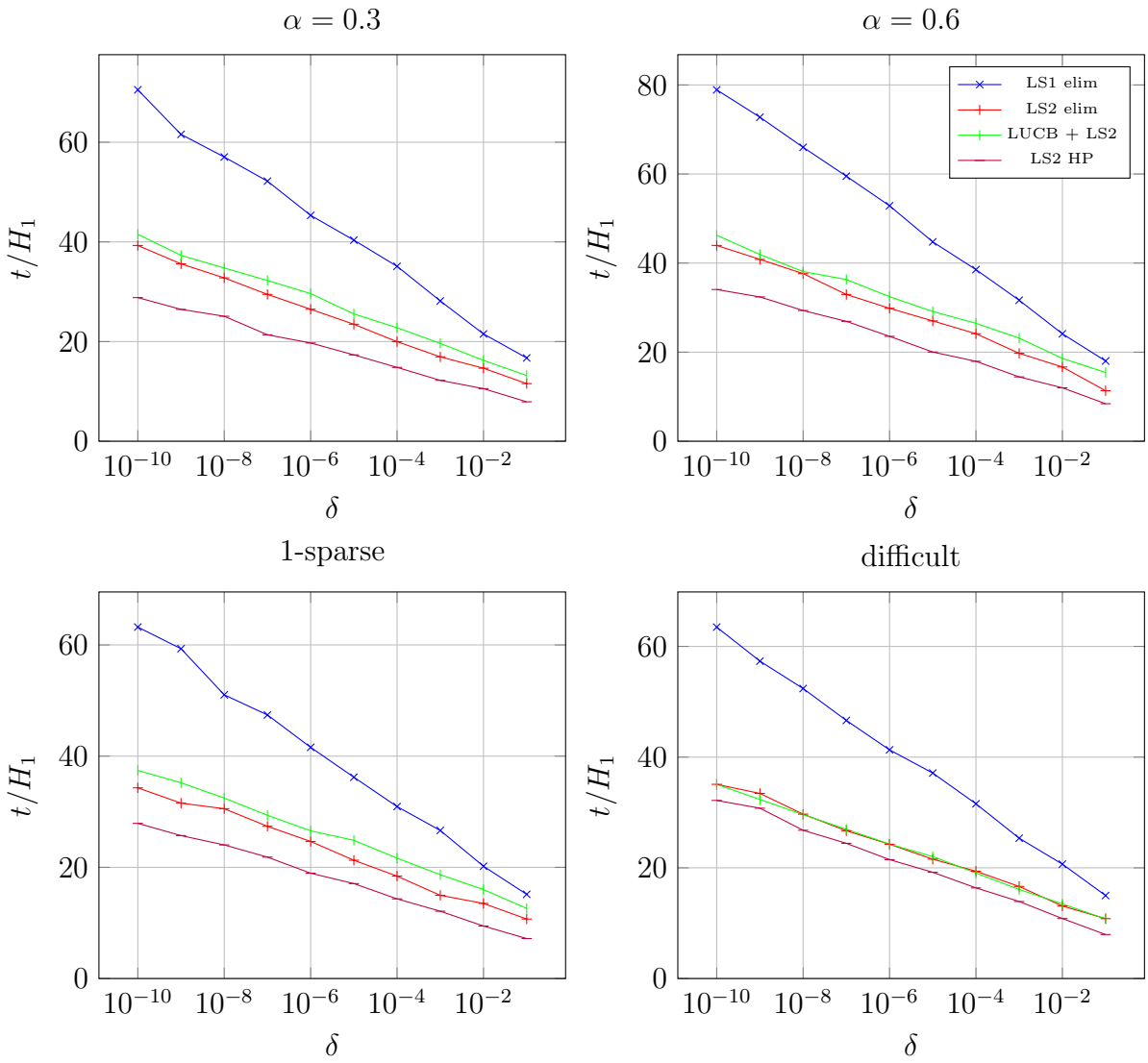

図 5.3: Sample-complexity vs. δ. ($N = 10$)

には効率が悪いという理由が考えられる。$N = 2$ のケースを考えると最適アーム探索はただの sequential test に帰着するので、その sequential test で提案した最適な sequential test を使っていることを考えると、$N = 2$ で効率が悪いということは考えづらい。一方で、式 (2.5) の下限の式は H_1 に比例する形になっているので、sample-complexity が下限に一致しているとすると、難易度の見積もりが間違っているということも考えづらいが、下限なので易しく見積もりすぎているということは考えられる。

sample-complexity のエラー率 δ への依存性についてだが、プロットの傾きに注目すると LS1 elimination だけ傾きが大きい。この傾きから sample-complexity の中で $H_1 \log(1/\delta)$ に掛かる係数を推定できるはずだが、sample-complexity に関する理論的な分析が出来なかった LS2 HP について見てみると、LS2 elimination とほぼ同じなので $H_1 \log(1/\delta)$ に掛かる係数に関しては、両者はほぼ同等と言える。一方で、$\alpha = 0.3, \alpha = 0.6$, 1-sparse では LS2 HP の方が LS2 elimination と比べて下にシフトした形になっている。この実験では N を固定しているので、LS2 elimination の $H_1 \log(N/\delta) = H_1 \log(N) + H_1 \log(1/\delta)$ の中の $H_1 \log(N)$ という項がシフトに関わっているのかも知れない。また、difficult では LS2 HP と LS2 elimination の差は小さくなっている。このことから、LS2 HP はある特定の場合には LS2 elimination よりも $H_1 \log(N)$ の分だけ sample

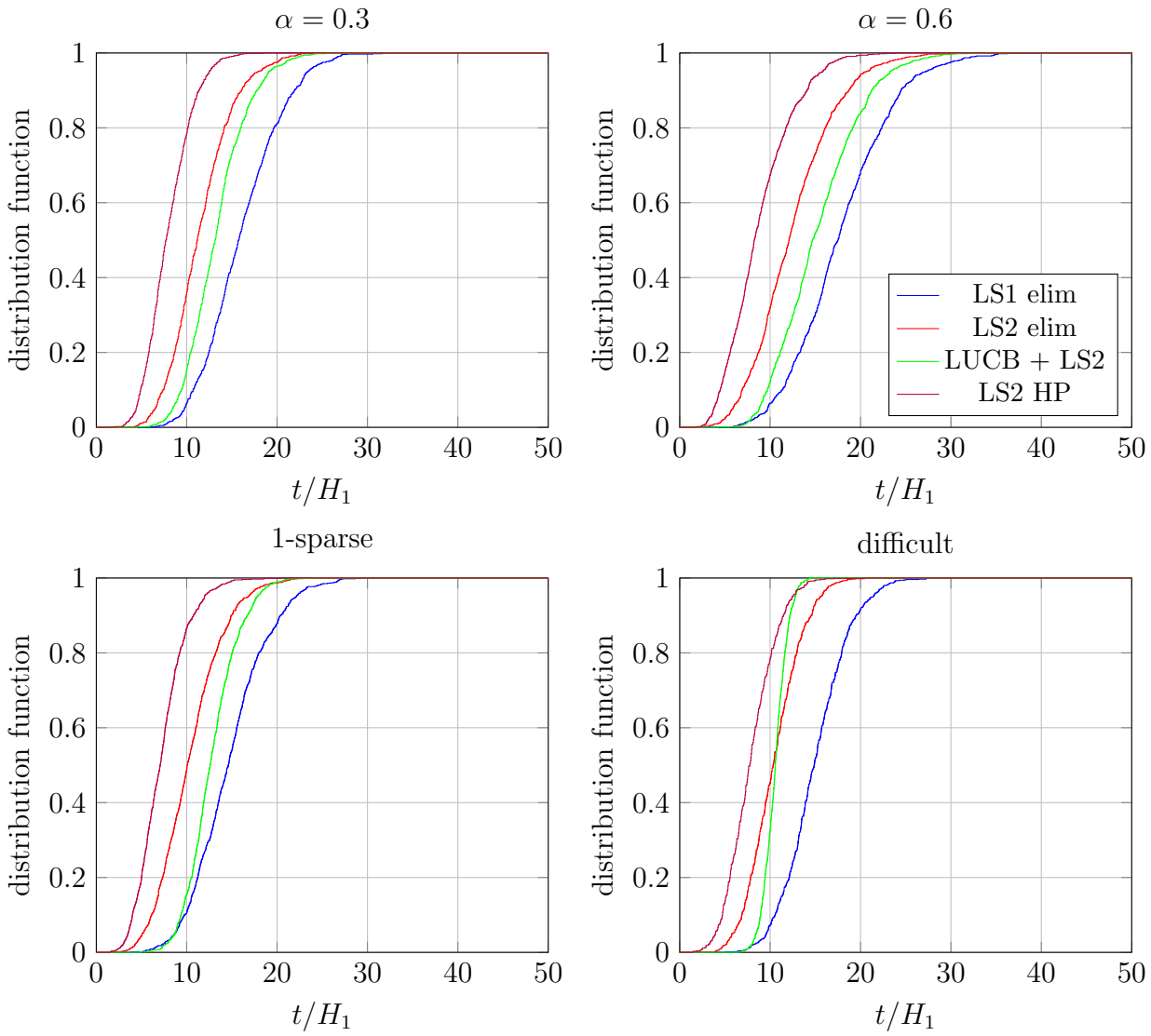

図 5.4: Distribution of sample-complexity. $(N = 10, \delta = 0.1)$

complexity が下がり、最悪の場合には LS2 elimination と同程度の sample complexity になるという予想が立つ。

sample-complexity の分布について、アルゴリズムが停止しないせいで停止時刻が物凄く大きくなってしまうということは再現できなかったので、理論的に期待値を押さえ込めなかったとしても実用上は問題無いと言える。LS2 HP の分布が他の手法の分布よりも左側に来ているので、期待値は下がるけれども分散が大きくなってしまうと言ったトレードオフ無しで、他の手法よりも sample-complexity が低いということが言える。

まとめると、LS1 elimination, LS2 elimination の理論的分析によると、sample-complexity のオーダーは従来のもの (lil'UCB) に劣るが、実際に実験すると従来よりも 3〜4 倍削減出来ている。LS2 HP は従来と比べて sample-complexity を 5〜6 倍削減できている。

第6章
結論

6.1 まとめ

今回は、確率的マルチアームドバンディットに対する最適アーム探索の必要サンプル数を改善するために、subgaussian 確率変数の平均に関して従来よりも必要サンプル数が小さくなるような sequential test を 2 つ提案した。うち 1 つは LIL の観点から考えて最適であることも示した。それら 2 つの sequential test を用いて successive elimination アルゴリズムの改良を行った。改良したアルゴリズムの sample-complexity はオーダー的に最適では無いが、実際には従来よりも小さくなるということをシミュレーションによって示した。これが例えば臨床試験に実際に応用されれば、被験者数、試験にかかる時間、コストが減ることが期待される。具体的には比較対象が 10 個程度で、エラー率 10% のときに、従来と比べて sample-complexity を 3 倍程度減らすことが出来る。

6.2 今後の展望

今後の研究の方向性としては、今まで提案された中で最も良いオーダーを持つ Exp-Gap[3] や lil'UCB[4] と同じオーダーを持ち、なおかつ提案手法と同じ程度の実性能を持つアルゴリズムを模索することや、提案した LS2 HP はアルゴリズムが必ず停止することを保証しているが、sample-complexity についての分析が出来ていないので、その分析を行うことが考えられる。また、実験で用いた LS2 HP のパラメータはあまり吟味されていないものなので、パラメータ調整についての研究が考えられる。アーム数 N を変えて sample-complexity の変化を見る実験では、LS2 HP の sample-complexity は N が増えるに従って減少したが、これは LS2 HP のエラー率を等配分しなくても良いという特徴を差し引いても減少しているように見える。この原理を解明して、LS2 HP をさらに改良するという方向も考えられる。

付録 A
Sequential Test に関する定理

A.1 $q(x)$ の数値積分

定義 2 の $p(x)$ を計算するためには $q(x)$ の積分値 Q を計算する必要があるが、積分区間が無限なのと $\lim_{x\to 0} p(x) = \infty$ なので、そのまま数値積分すると精度が良くならない。式 (A.1) に $q(x)$ を示す。ただし、$a = 2.085$ とする。

$$\int_0^\infty q(x)dx = \int_0^\infty \frac{1}{x(ax + \log(1 + 1/x)\log^2(1 + \log(1 + 1/x)))}dx$$

ここで、$y = \log(1 + 1/x), \tan(z) = \log(1 + y)$ と二段階に置換すると

$$\begin{aligned}\int_0^\infty q(x)dx &= \int_0^\infty \frac{1+x}{ax + y\log^2(1+y)}dy \\ &= \int_0^{\pi/2} \frac{(1+x)(1+y)(1+tan^2(z))}{ax + y\tan^2(z)}dz\end{aligned}$$

置換後の被積分関数を $r(z) = \frac{(1+x)(1+y)(1+tan^2(z))}{ax+y\tan^2(z)}$ と置くと、$\lim_{z\to 0} r(z) = 1/a, \lim_{z\to \pi/2} r(z) = 1$ となり端点で有限なので、精度良く計算出来ることが期待される。$r(z)$ のプロットを図 A.1 以下に示す。これを数値積分して式 (A.1) の値が得られた。

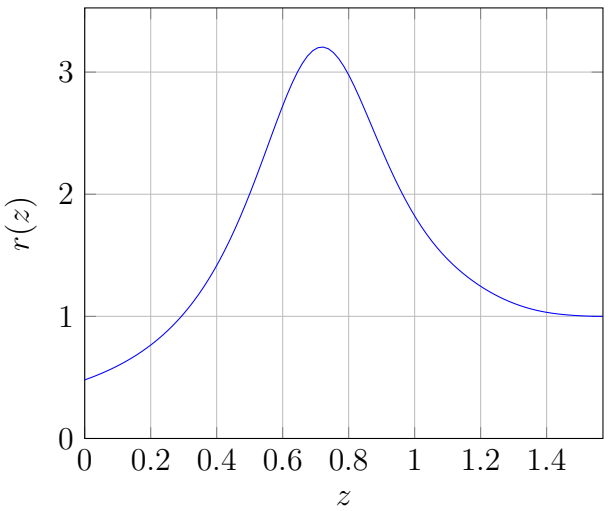

図 A.1: $r(z)$ plot.

$$Q = \int_0^\infty q(x)dx = \int_0^{\pi/2} r(z)dz \approx 2.519 \tag{A.1}$$

A.2　定理 1 の証明

確率空間 $(\Omega, \mathcal{F}, P)$ と、離散時間フィルトレーション $\{\mathcal{F}_t\}_{t=0,1,2,\cdots}$ を考える。$\{\Delta_t\}_{t=1,1,2,\cdots}$ を各 Δ_t が 1-subgaussian に従う martingale 差分列とする。このとき $X_t = \sum_{k=1}^{t} \Delta_k$ を考えると martingale になっている。ただし、$X_0 = 0$ とする。

補題 1. 実数 $s > 0$ に対して以下のような確率変数列 $\{Z_t\}_{t=0,1,2,\cdots}$ を考えると、Z_t は super martingale になっている。

$$Z_t = \exp\left(sX_t - \frac{s^2}{2}t\right)$$

証明.

$$\begin{aligned} E[Z_t|F_{t-1}] &= E\left[\exp\left(sX_t - \frac{s^2}{2}t\right)|F_{t-1}\right] \\ &= E\left[\exp\left(s\Delta_t - \frac{s^2}{2}\right)|F_{t-1}\right]\exp\left(sX_{t-1} - \frac{s^2}{2}(t-1)\right) \\ &\leq \exp\left(sX_{t-1} - \frac{s^2}{2}(t-1)\right) = Z_{t-1} \end{aligned}$$

□

定理 1. $u_t = \frac{1}{t}X_t$ として、$p(x) > 0$ を $0 < x$ の範囲で定義された $\int_0^\infty p(x)dx = 1$ である凸関数とする。また、$\{\delta\}_{t=1,2,\cdots}$ を各 $\delta_t > 0$ が $\mathcal{F}_{t-1}$ 可測である確率変数列とする。このとき以下の式が成り立つ。

$$\mathbb{P}\left[\exists t > 0, \quad \delta_t\, u_t\, p(u_t) \geq h\left(\frac{1}{2}t\, u_t^2\right)\right] \leq \sup_{0<t}\{\delta_t\}$$

証明. 任意の $s > 0$ に対して Z_t は super martingale になるので、色々な s に対する Z_t を加重平均したものも super martingale になる。以下の式のように、$p(s) > 0$ で加重平均したものを M_t と置く。super martingale を加重平均するというアイデアは、[18], [6] にヒントを得た。

$$M_t = \int_0^\infty p(s)\exp\left(sX_t - \frac{s^2}{2}t\right)ds$$

$X_0 = 0$ より $M_0 = 1$ なので任意の停止時刻 τ と $T > 0$ に対して以下の式が成り立つ。

$$\begin{aligned} 1 &\geq E[M_\tau] \\ &\geq E[M_\tau; \tau \leq T] \\ &= E[M_\tau|\tau \leq T]P(\tau \leq T) \end{aligned} \tag{A.2}$$

ここで、停止時刻 τ を以下のように定義する。

$$\tau = \min\left\{t \in [1, T] : M_t > \frac{1}{\delta_t}\right\}$$

ここで、次のように $E[M_\tau]$ に注目する。

$$\begin{aligned} E[M_\tau|\tau \le 1] &\ge E[\frac{1}{\delta_1}|\mathcal{F}_1] = \frac{1}{\delta_1} \\ E[M_\tau|\tau \le t] &\ge \frac{1}{P(\tau \le t)}\left(P(\tau \le t-1)E[M_\tau|\tau \le t-1] + P(\tau = t)E[\frac{1}{\delta_t}|\mathcal{F}_t]\right) \\ &= \frac{1}{P(\tau \le t)}\left(P(\tau \le t-1)E[M_\tau|\tau \le t-1] + P(\tau = t)\frac{1}{\delta_t}\right) \quad (2 \le \forall t) \end{aligned}$$

$P(\tau \le t) = P(\tau \le t-1) + P(\tau = t)$ であり $M_t > 0$ なので以下が成り立つ。

$$E[M_\tau|\tau \le T] \ge \inf_{0<t}\left\{\frac{1}{\delta_t}\right\} \tag{A.3}$$

式 (A.2), (A.3) を用いると

$$1 \ge E[M_\tau|\tau \le T]P(\tau \le T) \ge \inf_{0<t}\left\{\frac{1}{\delta_t}\right\} P(\tau \le T)$$

よって

$$P(\tau \le T) < \sup_{0<t}\{\delta_t\} \tag{A.4}$$

T は任意に大きくしても良いので、τ を sequential test の停止基準として用いると第一種エラー率を δ 以下に抑えることが出来る。実際に τ で停止判定するためには M_t を計算する必要があるので、今度は M_t を以下のように計算しやすい式で評価していく。ただし、$g(\mu, \sigma, x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x-\mu}{2\sigma^2}\right)$ は正規分布の確率密度関数である。

$$\begin{aligned} M_t &= \int_0^\infty p(s)\exp\left(sX_t - \frac{s^2}{2}t\right)ds \\ &= \int_0^\infty p(s)\exp\left(-\frac{t}{2}\left(s - \frac{X_t}{t}\right)^2 + \frac{X_t^2}{2t}\right)ds \\ &= \sqrt{\frac{2\pi}{t}}\exp\left(\frac{X_t^2}{2t}\right)\int_0^\infty p(s)g\left(\frac{X_t}{t}, \frac{1}{\sqrt{t}}, s\right)ds \end{aligned}$$

ここで $p(s)$ は凸であり $0 < x < \frac{X_t}{t}$ に対して $p\left(\frac{X_t}{t} - x\right) + p\left(\frac{X_t}{t} + x\right) \ge 2\,p\left(\frac{X_t}{t}\right)$ が成り立つので、以下のように評価できる。ただし、$\mathrm{erf}(x)$ は誤差関数である。

$$\begin{aligned} M_t &\ge \sqrt{\frac{2\pi}{t}}\exp\left(\frac{X_t^2}{2t}\right)p\left(\frac{X_t}{t}\right)\int_0^{\frac{2X_t}{t}} g\left(\frac{X_t}{t}, \frac{1}{\sqrt{t}}, s\right)ds \\ &= \sqrt{\frac{2\pi}{t}}\exp\left(\frac{X_t^2}{2t}\right)\mathrm{erf}\left(\frac{X_t}{\sqrt{2t}}\right)p\left(\frac{X_t}{t}\right) \end{aligned}$$

$u_t = \frac{X_t}{t}$ を用いて書き直すと

$$M_t \ge \sqrt{\frac{2\pi}{t}}\exp\left(\frac{1}{2}tu_t^2\right)\mathrm{erf}\left(\sqrt{\frac{1}{2}tu_t^2}\right)p\left(u_t\right)$$

ここで、以下のような関数 $h(x)$ を用意する。

$$h(x) = \frac{\sqrt{x}}{\sqrt{\pi}\mathrm{erf}(\sqrt{x})}\exp(-x)$$

この $h(x)$ を用いて書き直すと

$$M_t \;\geq\; u_t p\left(u_t\right)\left(h\left(\frac{1}{2}tu_t^2\right)\right)^{-1}$$

これと式 (A.4) を用いて停止基準を作ると以下のように第一種エラー率を $\sup_{0<t}\{\delta_t\}$ 以下にすることが出来る。

$$\begin{aligned}
& \mathbb{P}\left[\exists t>0, \quad \delta_t\, u_t\, p(u_t) \geq h\left(\frac{1}{2}t\, u_t^2\right)\right] \\
= \;& \mathbb{P}\left[\exists t>0, \quad \frac{1}{\delta_t} \leq u_t p\left(u_t\right)\left(h\left(\frac{1}{2}tu_t^2\right)\right)^{-1}\right] \\
\leq \;& \mathbb{P}\left[\exists t>0, \quad \frac{1}{\delta_t} \leq M_t\right] \\
\leq \;& \sup_{0<t}\{\delta_t\}
\end{aligned}$$

□

A.3　定理 2 の証明

まず、定義 1 の $h(x) = \sqrt{x}/(\sqrt{\pi}\,\mathrm{erf}(\sqrt{x}))$ を上から評価するために以下の補題を証明する。

補題 2. $x \geq 2$ のとき以下の式が成り立つ。

$$\frac{x}{\mathrm{erf}(x)} \leq \frac{5}{4}x \tag{A.5}$$

また、このことから定義 1 の $h(x)$ を以下のように評価できる。

$$h(x) \leq \frac{5\sqrt{x}}{4\sqrt{\pi}}\exp(-x) \quad (x \geq 2) \tag{A.6}$$

証明. [20] によると、$x > -1$ のとき以下の式が成り立つ。

$$1 - \frac{2\sqrt{8/\pi}}{3x + \sqrt{x^2+8}}\exp(-x^2/2) < \mathrm{erf}(x)$$

従って、$x \geq 2$ のとき以下のように評価できる。

$$\begin{aligned}
\frac{x}{\mathrm{erf}(x)} \;&\leq\; \frac{x}{1 - \frac{2\sqrt{8/\pi}}{3x+\sqrt{x^2+8}}\exp(-x^2/2)} \\
&\leq\; \frac{x}{1 - \frac{2\sqrt{8/\pi}}{3\cdot 2+\sqrt{2^2+8}}\exp(-2^2/2)} \\
&\leq\; \frac{x}{1 - \frac{2\sqrt{2}}{3(3+\sqrt{3})}\cdot\frac{1}{7}} \\
&\leq\; \frac{x}{1-\frac{1}{5}} < \frac{5}{4}x
\end{aligned}$$

□

次に、定理 2 の証明に入る。

定理 2. 任意の $0 < \delta < 1$ に対して、以下のような関数 $u(t)$ を考える。

$$u(t) = \sqrt{\frac{2}{t}} \sqrt{-\frac{1}{2} W_{-1}\left(-\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2} \right)} \tag{A.7}$$

このとき $T = \frac{25a^2Q^2}{2\delta^2 e^2 \pi}$ として、全ての整数 $t > T$ で以下の式が成り立つ。

$$\delta u(t) p(u(t)) \geq h\left(\frac{1}{2}tu^2(t)\right) \tag{A.8}$$

また、$t \to \infty$ で以下のような性質を持つ。

$$\limsup_{t\to\infty} \frac{u(t)}{\sqrt{t \log\log(t)}} = \sqrt{2} \tag{A.9}$$

証明. まず、$u(t)$ を関数 $f(t) \geq 2$ を用いて以下の式のように表す。

$$u(t) = \sqrt{\frac{2f(t)}{t}} \tag{A.10}$$

次に、式 (A.8) を変形する。

$$\delta u(t) p(u(t)) \geq h\left(\frac{1}{2}tu^2(t)\right) \Leftrightarrow \frac{\delta}{Q} \geq \frac{1}{u(t)q(u(t))} h\left(\frac{1}{2}tu^2(t)\right) \tag{A.11}$$

この式の右辺を過大評価していく。

$$\begin{aligned}
& \frac{1}{u(t)q(u(t))} h\left(\frac{1}{2}tu^2(t)\right) \\
= \;& \left(a\sqrt{\frac{2f(t)}{t}} + \log(1+\sqrt{\frac{t}{2f(t)}})\log^2(1+\log(1+\sqrt{\frac{t}{2f(t)}})) \right) h(f(t)) \\
\leq \;& \left(a\sqrt{\frac{2f(t)}{t}} + \log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}})) \right) h(f(t)) \\
\leq \;& 2\max\left\{ a\sqrt{\frac{2f(t)}{t}}, \log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}})) \right\} h(f(t))
\end{aligned} \tag{A.12}$$

この条件を書き直すと以下のように表現できる。

$$\frac{\delta}{2Q} \geq a\sqrt{\frac{2f(t)}{t}} h(f(t)) \tag{A.13}$$

$$\text{and} \quad \frac{\delta}{2Q} \geq \log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))h(f(t)) \tag{A.14}$$

$$\Rightarrow \quad \delta u(t) p(u(t)) \geq h\left(\frac{1}{2}tu^2(t)\right)$$

まず、最初の条件式 (A.13) を見てみる。補題 2 を用いると

$$\frac{\delta}{2b} \geq a\sqrt{\frac{2f(t)}{t}}h(f(t)) \tag{A.15}$$

$$\Leftarrow \quad \frac{\delta}{2b} \geq a\sqrt{\frac{2f(t)}{t}}\frac{5\sqrt{f(t)}}{4\sqrt{\pi}}\exp(-f(t)) \tag{A.16}$$

$$\Leftarrow \quad -f(t)\exp(-f(t)) \geq -\frac{2\delta\sqrt{\pi}}{5aQ}\sqrt{\frac{t}{2}} \tag{A.17}$$

ランベルトの W 関数を用いると

$$-f(t)\exp(-f(t)) \geq -\frac{2\delta\sqrt{\pi}}{5aQ}\sqrt{\frac{t}{2}} \tag{A.18}$$

$$\Leftarrow \quad \frac{2\delta\sqrt{\pi}}{5aQ}\sqrt{\frac{t}{2}} \geq \frac{1}{e} \tag{A.19}$$

$$\text{or} \quad f(t) \geq \max\left\{2, -W_{-1}\left(-\frac{2\delta\sqrt{\pi}}{5aQ}\sqrt{\frac{t}{2}}\right)\right\} \tag{A.20}$$

次に二つ目の条件式 (A.14) を見てみる。一つ目の条件と同様に補題 2 を用いると

$$\frac{\delta}{2Q} \geq \log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))h(f(t)) \tag{A.21}$$

$$\Leftarrow \quad \frac{\delta}{2b} \geq \log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\frac{5\sqrt{f(t)}}{4\sqrt{\pi}}\exp(-f(t)) \tag{A.22}$$

$$\Leftarrow \quad -2f(t)\exp(-2f(t)) \geq -\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2} \tag{A.23}$$

同様にランベルトの W 関数を用いると

$$-2f(t)\exp(-2f(t)) \geq -\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2} \tag{A.24}$$

$$\Leftarrow \quad \frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2} \geq \frac{1}{e} \tag{A.25}$$

$$\text{or} \quad f(t) \geq \max\left\{2, -\frac{1}{2}W_{-1}\left(-\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2}\right)\right\} \tag{A.26}$$

$t \to \infty$ のときに $f(t)$ がどういうオーダーになるかを調べる。まず、十分大きい T に対して $t > T$ で一つ目の条件 $\frac{2\delta\sqrt{\pi}}{5aQ}\sqrt{\frac{t}{2}} \geq \frac{1}{e}$ は満たされる。従って、$f(t)$ は二つ目の条件によって決まる。また十分大きい t で、$f(t) \geq 2$ も満たされる。等号を成立させてしまうと

$$f(t) = -\frac{1}{2}W_{-1}\left(-\frac{8\delta^2\pi}{25Q^2\left(\log(1+\sqrt{\frac{t}{2}})\log^2(1+\log(1+\sqrt{\frac{t}{2}}))\right)^2}\right) \tag{A.27}$$

$W_{-1}(x) = \log(-x) + O(\log(-\log(-x)))$ であることが知られているので、十分大きい t に対して

$$f(t) = -\frac{1}{2}\left(2\log(\delta/\log(t)) + O(\log(-\log(\delta/\log(t))))\right) \tag{A.28}$$

$$= \log(\log(t)/\delta) + O(\log(\log(\log(t)/\delta))) \tag{A.29}$$

最終的に $u(t)$ の式として書くと

$$u(t) = \sqrt{\frac{2}{t}}\sqrt{\log(\log(t)/\delta) + O(\log(\log(\log(t)/\delta)))} \tag{A.30}$$

$u(t)$ は law of the iterated logarithm から考えられる最適なオーダーと係数を持つことが分かった。 □

A.4　定理 3 の証明 (2 変数 sequential test)

定理 3. $\{\Delta X_n\}_{n=1,2,\cdots}, \{\Delta Y_m\}_{m=1,2,\cdots}$ を各 $\Delta X_n, \Delta Y_m$ が平均 0 で 1-subgaussian に従う確率変数列とする。$\{n_t\}_{t=0,1,2,\cdots}, \{m_t\}_{t=0,1,2,\cdots}$ を $n_0 = 0, m_0 = 0$ であり毎時刻どちらかだけが 1 増える確率変数列とする。$u_t = \frac{1}{n_t}\sum_{k=1}^{n_t}\Delta X_k, v_t = \frac{1}{m_t}\sum_{k=1}^{m_t}\Delta Y_k$ とする。$p(x) > 0$ を $0 < x$ の範囲で定義された $\int_0^\infty p(x)dx = 1$ である凸関数とする。また、$\{\delta\}_{t=1,2,\cdots}$ を各 $\delta_t > 0$ が $\mathcal{F}_{t-1}$ 可測である確率変数列とする。このとき以下の式が成り立つ。

$$\mathbb{P}\left[\exists t > 0, \quad \delta_t\, u_t\, p(u_t)\, v_t\, p(v_t) \geq h\left(\frac{1}{2}n_t\, u_t^2\right) h\left(\frac{1}{2}m_t\, v_t^2\right)\right] \leq \sup_{0<t}\{\delta_t\} \tag{A.31}$$

証明. まず、$X_{n_t} = \sum_{k=1}^{n_t}\Delta X_k, Y_{m_t} = \sum_{k=1}^{m_t}\Delta Y_k$ を考える。ただし、$X_0 = Y_0 = 0$ とする。任意の実数 $u, v > 0$ に対して、以下の確率変数列 $\{Z_t\}_{n=0,1,2,\cdots}$ を考えると、毎時刻 X_{n_t}, Y_{m_t} の片方しか増えないので、1 変数のときと同様に super martingale になっている。

$$Z_t = \exp\left(uX_{n_t} + vY_{m_t} - \frac{1}{2}u^2 n_t - \frac{1}{2}v^2 m_t\right) \tag{A.32}$$

これを関数 $p(u)\cdot p(v)$ で加重平均したものを M_t とすると、M_t も super martingale となっている。

$$M_t = \int_0^\infty\int_0^\infty p(u)p(v)\exp\left(uX_{n_t} + vY_{m_t} - \frac{1}{2}u^2 n_t - \frac{1}{2}v^2 m_t\right)du\, dv \tag{A.33}$$

u, v についての積分にまとめてしまうと

$$M_t = \left(\int_0^\infty p(u)\exp\left(uX_{n_t} - \frac{1}{2}u^2 n_t\right)du\right)\left(\int_0^\infty p(v)\exp\left(vY_{m_t} - \frac{1}{2}v^2 m_t\right)dv\right) \tag{A.34}$$

1 変数のときの積分が 2 つ掛け算されている形なので、1 変数のときと同様に $h(x)$ で表現すると最終的に以下のように sequential test の停止基準を作ることが出来る。

$$\begin{aligned}
&\mathbb{P}\left[\exists t > 0, \quad \delta_t\, u_t\, p(u_t)\, v_t\, p(v_t) \geq h\left(\frac{1}{2}n_t\, u_t^2\right) h\left(\frac{1}{2}m_t\, v_t^2\right)\right] \\
\leq\ &\mathbb{P}\left[\exists t > 0, \quad \frac{1}{\delta_t} \leq M_t\right] \\
\leq\ &\sup_{0<t}\{\delta_t\}
\end{aligned}$$

□

謝辞

まず、本研究を進めていくにあたって、常日頃からご指導を賜りました指導教員の若原恭教授に、心から感謝致します。そして、本研究において数多くの助言やご指導を頂いた、中山雅哉准教授、小川剛史准教授、関谷勇司准教授、妙中雄三助教、宮本大輔助教に深く感謝しております。また、研究室の皆様方にも日頃から研究活動を様々な面で支えてくださったことを感謝致します。

2015 年 2 月 5 日
福勢 晋

参考文献

[1] Edward Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *Ann. Math. Statist.*, Vol. 35, No. 1, pp. 174–180, 03 1964.

[2] Eyal Even-dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *In Fifteenth Annual Conference on Computational Learning Theory (COLT)*, pp. 255–270, 2002.

[3] Zohar Shay Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *ICML (3)*, Vol. 28 of *JMLR Proceedings*, pp. 1238–1246. JMLR.org, 2013.

[4] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *Volume 35: Proceedings of The 27th Conference on Learning Theory*, pp. 423–439, 2014.

[5] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In John Langford and Joelle Pineau, editors, *In proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 655–662, New York, NY, USA, June-July 2012. Omnipress.

[6] Akshay Balsubramani. Sharp uniform martingale concentration bounds. *CoRR*, 2014. arXiv:1405.2639v3 [math.PR].

[7] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, Vol. 47, No. 2-3, pp. 235–256, May 2002.

[8] Aurelien Garivier. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *In Proceedings of COLT*, 2011.

[9] Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *In Proceedings of the Twenty-third Conference on Learning Theory (COLT) 2010*, pp. 67–79. Omnipress, 2010.

[10] W. R. Thompson. On the Likelihood that one Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, Vol. 25, pp. 285–294, 1933.

[11] Emilie Kaufmann, Aurelien Garivier, and Telecom Paristech. On bayesian upper confidence bounds for bandit problems. In *In AISTATS*, 2012.

[12] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, Vol. 6, No. 1, pp. 4–22, 1985.

[13] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23th annual conference on Computational Learning Theory*, Haífa (Israél), Jun 2010.

[14] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Leon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pp. 3221–3229, 2012.

[15] S Mannor and JN Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, Vol. 5, pp. 623–648, 2004.

[16] 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*, Vol. 7, pp. 1079–1105, 2006.

[17] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML)*, pp. 511–518, 2010.

[18] Tze Leung Lai. Martingales in sequential analysis and time series, 1945?1985. *Electronic J. History Probab. Statist*, 2009.

[19] Jamieson Kevin, Malloy Matthew, Nowak Robert, and Bubeck Sébastien. On finding the largest mean among many. *CoRR*, 2013. arXiv:1306.3917v1 [stat.ML].

[20] Stanislaw J. Szarek and Elisabeth Werner. A nonsymmetric correlation inequality for gaussian measure. *Journal of Multivariate Analysis*, Vol. 68, No. 2, pp. 193 – 211, 1999.