CONTROL OF SAMPLE COMPLEXITY AND REGRET IN BANDITS USING FRACTIONAL MOMENTS

A Project Report

submitted by

ANANDA NARAYAN

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.
April 2012

THESIS CERTIFICATE

This is to certify that the thesis titled **CONTROL OF SAMPLE COMPLEXITY AND REGRET IN BANDITS USING FRACTIONAL MOMENTS**, submitted by **ANANDA NARAYAN**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr.B. Ravindran
Associate Professor
Dept. of Computer Science and Engineering

Dr.R. Aravind
Professor
Dept. of Electrical Engineering

IIT-Madras, 600036

Place: Chennai

Date: $10^{th} of May$, 2012

ACKNOWLEDGEMENTS

I am very thankful to my project guide Dr. Ravindran for his prolific encouragement and guidance. If not for his receptivity towards his students, this work would not have been possible. I would also like to thank Dr. Andrew, Dr. Aravind and Dr. Venkatesh for the courses they offered that stood me in good stead during the course of this project.

I would like to extend my profound gratitude to the faculty at the Department of Electrical Engineering, especially to Dr. Giridhar, Dr. Devendra Jalihal and Dr. Jagadeesh Kumar, and our then honourable Director Dr. M.S.Ananth, for their broad and liberal views towards promoting students to explore their curricula beyond defined horizons. I believe their never-ending support for excellence in students, unrestricted by all other means, was influential in shaping the dawn of my career. I thank the faculty at the Department of Electrical Engineering and the Department of Computer Science and Engineering for the wonderful mix of courses I could avail in the past five years. I also thank the staff at both Departments for all the facilities and support during this project.

I would also like to thank my friends and colleagues for making the 5 years of my stay here most cherishable. Their presence unfolded numerous eventful experiences that I will always remember.

Finally, I would like to thank my parents for all their love, affection and teaching. My learning from their life experiences has been instrumental in success of this endeavour.

Control of Sample Complexity and Regret in Bandits using Fractional Moments

Ananda Narayan

Abstract

One key facet of learning through reinforcements is the dilemma between exploration to find profitable actions and exploitation to act optimal according to the observations already made. We analyze this explore/exploit situation on Bandit problems in stateless environments. We propose a family of learning algorithms for bandit problems based on fractional expectation of rewards acquired. The algorithms can be controlled to behave optimal with respect to sample complexity or regret, through a single parameter. The family is theoretically shown to contain algorithms that converge on an ϵ-optimal arm and achieve $O(n)$ sample complexity, a theoretical minimum. The family is also shown to include algorithms that achieve the optimal logarithmic regret $O(\log(t))$ proved by Lai & Robbins (1985), again a theoretical minimum. We also propose a method to tune for a specific algorithm from the family that can achieve this optimal regret. Experimental results support these theoretical findings and the algorithms perform substantially better than state-of-the-art techniques in regret reduction like UCB-Revisited and other standard approaches.

Contents

List of Figures

Chapter 1

Introduction

Artificial Intelligence has been one of the promising observatories that the current world is in lookout for. It takes a leap of advancement for algorithms to move from solving automation, towards demonstrating intelligence. Intelligent agents not only should learn from their experiences, but also need to find how such experiences can be achieved. They have to act in such a way that, the associated experiences that follow prove noteworthy for improvement of their current learning of the system at hand. They have to repeatedly interact with the system to learn decision-making on the system. Reinforcement learning is one approach that provides for ample scope for such agents. We will now introduce some of the salient aspects of reinforcement learning and discuss some of its applications.

1.1 Reinforcement Learning

Some of the well known models of machine learning constituting supervised, unsupervised and semi-supervised approaches have been extensively used for classification, regression, clustering, finding frequenting patterns etc. But these approaches do not provide avenues for learning to cycle — learn decision-making on a system by repeatedly interacting with the same. Reinforcement learning is an approach that allows for such trial and error based learning.

Reinforcement learning is a learning paradigm that lets an agent to map situations (as perceived in the environment it acts on), to actions (to take on the environment), so as to maximize a numerical reward signal (from the environment). Figure 1.1.1 describes a simple outline of a reinforcement learner. A decision-making and learning

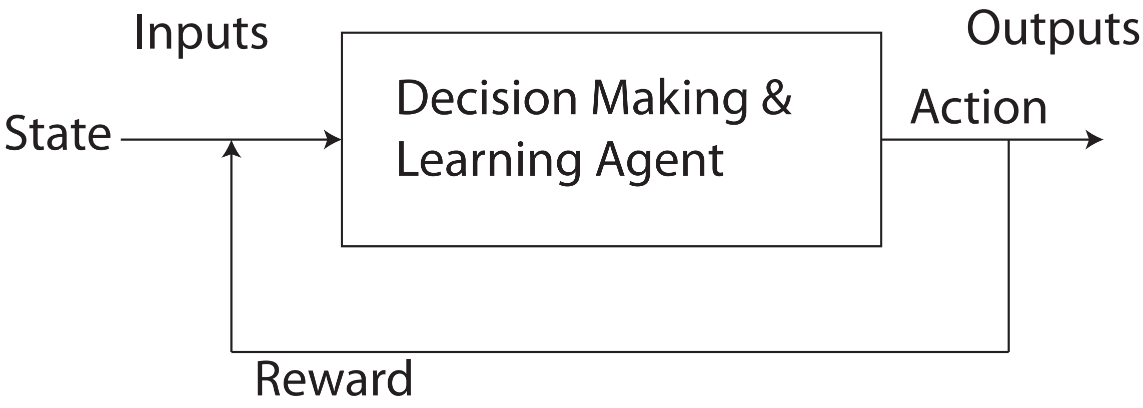


Figure 1.1.1: Reinforcement Learning framework

agent perceives states of its environment to perform a decision on a set of actions it can take on the environment. In deciding and taking an action, the agent receives feedback about how beneficial the action is for the corresponding state of the environment. This feedback is regarded as a 'Reward' signal, usually stochastic in nature. Higher the reward, better was the action selected for the perceived state of the environment. And in taking an action, the environment changes it state, which is again usually stochastic. The learning task is to find a mapping from states to actions. The optimal mapping may be stochastic or deterministic depending on the stochasticity in reward and state transitions on taking actions. In a nutshell, reinforcement learning enables an agent to perform decision-making on a system (environment), perceiving its state and the rewards it provides.

Example 1. Consider a stochastic maze environment and an agent that tries to solve it obtaining rewards at few destinations as in figure 1.1.2 [1]. When at any of the squares in the maze (state), the agent has a set of actions to choose from that is always a subset of the set $\mathbb{S} = \{forward,\ backward\ ,\ right,\ left,\ idle\}$. The particular set of actions available is dependent on the state of the agent in the environment. Taking one of the actions may not necessarily make the agent move in the same particular direction. The environment is stochastic in changing the agent's state (as indicated in the figure) and the agent may move in any direction, or may remain idle, for taking a particular action, say moving forward. After an action is taken, the agent receives a reward that is pertaining to the triple containing the past-state of the agent, action taken by the agent, and the future-state where the agent ended up due to the action taken. This reward is that numerical value indicated in the figure for the destination states and is zero for all other states. Once one of the destination states is reached, the maze

[1] Source: Dayan & Abbott (2001)

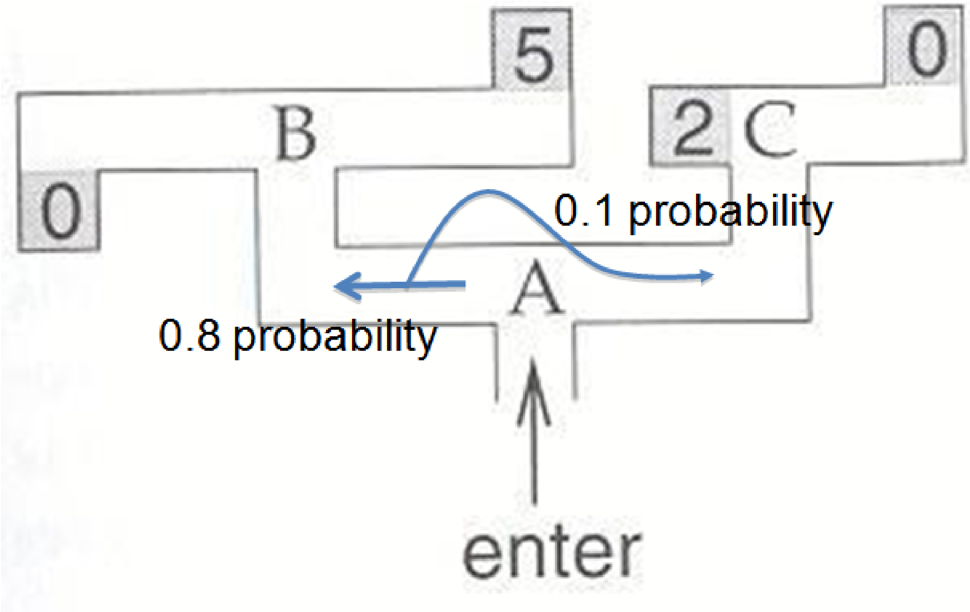


Figure 1.1.2: A stochastic maze environment

is solved (with a finishing reward that the destination state provides) and learning stops. Note that the reward for selecting a path towards the optimal destination is only available at the end of the sequence of action selections. Hence, one need to adhere every reward to not only the last action taken but to the entire sequence of actions that led to this reward. So the reward to selecting an action can be delayed. By performing these action selections, the task is to learn a mapping of states to actions so as to reach those destination states with the highest numerical reward.

To conclude, reinforcement learning is a framework to learn about a system through interaction. The learning is about optimal actions to take on the system, and is based on rewards (and punishments) alone that are fed-back from the system. No detailed supervision about the correctness of an action is available, as in supervised learning. The framework is hence a trial and error learning paradigm that may encounter delayed rewards. Also, a sequence of desirable actions is needed to obtain a desirable reward. The objective is to perform associative learning with the need to associate the learning of actions to states where it was taken. So it is about learning a sequence of actions, sometimes referred to as a policy, rather than learning just actions. The environment is typically a stochastic world, that may have a deterministic or stochastic optimal policy that needs to be learnt.

1.2 Multi-arm Bandits

A multi-arm bandit problem requires finding optimal learning strategies for a simplified version of the reinforcement learning agent described above. The simplification is in the state space. In the multi-arm bandit problem, the environment is stateless or can be said to be at the same state at all times. This reduces the complexity in learning by a large amount due to the absense of state transitions of the environment and the absense of reward dependence on state-action association. So the rewards are only a function of the action taken, and there is a set of actions from which an optimal action needs to be selected.

The multi-arm bandit takes it name from a slot machine (a gambling machine that rotates reels when a lever or arm is pulled). A traditional slot machine is a one-arm bandit and a multi-arm bandit has multiple levers or arms that can be pulled. Thus, there is a set of levers to choose from at all times, and the task is to find that lever that would give us the maximum return (or reward).

1.3 Problem formulation

An n-arm bandit problem is to learn to preferentially select a particular action (or pull a particular arm) from a set of n actions (arms) numbered $1, 2, 3, \ldots, n$. Each pull of an arm results in a random reward corresponding to the arm being pulled. Let this random reward be distributed according to a random variable R_i, with mean μ_i, usually stationary. Arms are pulled and rewards acquired until learning, to find the arm with the highest rewards corresponding to the mean $\mu^* = \max_i\{\mu_i\}$, converges.

1.3.1 Learning Objectives and Optimality

There have been two main objectives defined and analyzed for the multi-arm bandit problem in the literature. One of them releates to reducing the regret (or loss) while trying to learn the best arm. Another relates to reducing the number of samples (or pulls of arms) to be drawn to find an arm close to the optimal arm with a high probability. We define these quantifications of optimality formally below.

1.3.1.1 Regret Reduction

The traditional objective of the bandit problem is to maximize total reward (or gain) given a specified number of pulls to be made, hence minimizing regret (or loss). Recall that an arm corresponding to a mean equal to μ^* is considered the best arm as it is expected to give the maximum reward per pull. Let Z_t be the total reward acquired over t successive pulls. Then regret η is defined as the expected loss in total reward acquired had the best arm been repetitively chosen right from the first pull, that is, $\eta(t) = t\mu^* - E[Z_t]$. This quantification is also sometimes referred to as cumulative regret.

1.3.1.2 Sample Complexity Reduction

We will now introduce a Probably Approximately Correct (PAC) framework and the associated sample complexity, and an optimality criterion based on sample complexity reduction. We start with a definition for ϵ-optimality.

Definition 2. An arm j is considered an ϵ-optimal arm if $\mu_j > \mu^* - \epsilon$, where $\mu^* = \max_i\{\mu_i\}$.

We define the Probably Approximately Correct (PAC) quantification of optimality below.

Definition 3. Even-Dar et al. (2006) An algorithm is a (ϵ, δ)-PAC algorithm for the multi arm bandit with sample complexity $\mathcal{T}$, if it outputs an ϵ-optimal arm, with probability at least $1 - \delta$, when it terminates, and the number of time steps the algorithm performs until it terminates is bounded by $\mathcal{T}$

The number of time steps is also known as the number of samples (pulls of arms). Hence, the objective is then to reduce $\mathcal{T}$, often referred to as sample complexity.

1.4 Applications

The multi-arm bandit problem has numerous applications in decision theory, evolutionary programming, reinforcement learning, search, recommender systems, and even in a few interdisciplinary areas like industrial engineering and simulation. We now describe a very brief collection of examples of such applications.

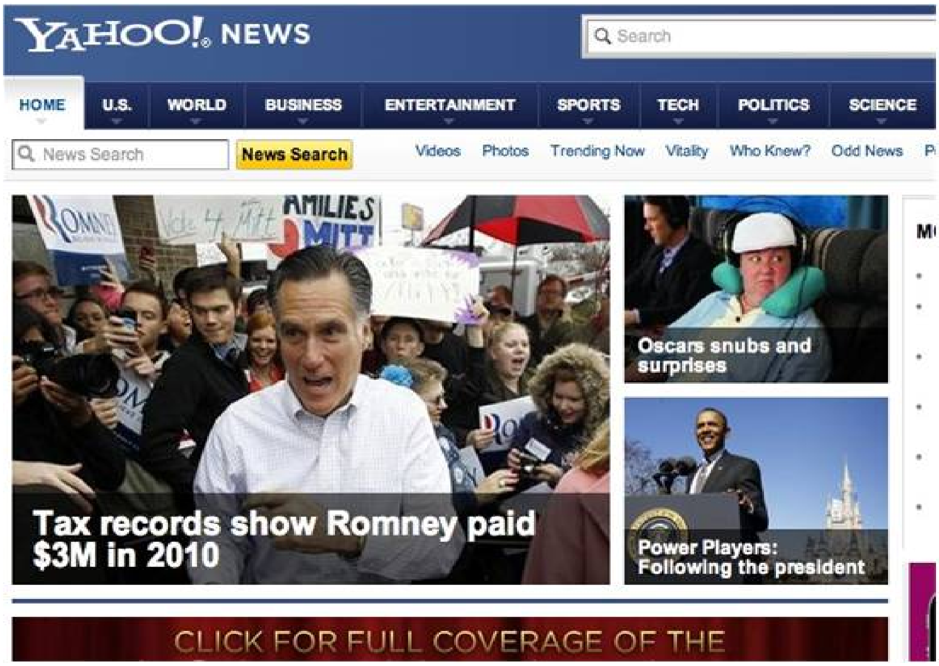


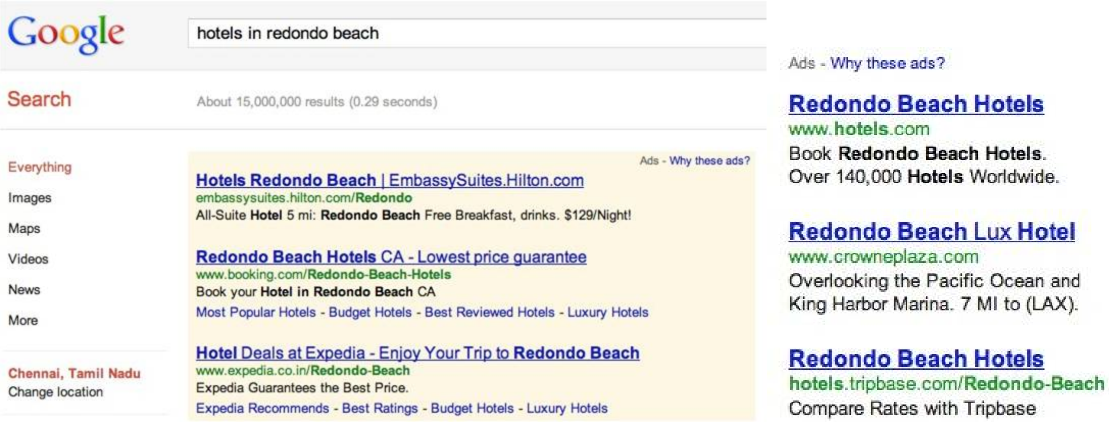
Figure 1.4.1: Yahoo Frontpage Today Module

Figure 1.4.1 [2] shows the home page of the popular web service provider Yahoo!, featuring its Frontpage Today Module. The top stories featured in this module uses a multi-arm bandit formulation to choose the top four articles that are frequently viewed by users who visit the page. A set of editor-picked stories (sometimes called a story-bucket) is made available to be the arms to choose from, and a bandit learner finds the top four optimal arms to display. This learning happens continuously, in an online manner, with the story-bucket dynamically changing, as and when more relevant news articles arrive. The algorithm we will propose in chapter 2, is compatible for implementation with all the requirements mentioned above. For matching news articles specific to user interests, demography etc., a set of input features specific to the user is often used in finding the best news stories to be displayed. This introduces us to a new formulation often called Contextual Bandits, due to the use of contextual information in finding the best arm.

Figure 1.4.2 [3] depicts a typical set of ads chosen for a specific search query by Google. This problem is many times analyzed using a multi-arm bandit formulation with millions of ads relating to the set of arms to choose from. Contextual information is ofcourse used specific to the search query at hand. This is one of the best examples of multi-arm bandit appications where sample complexity reduction is very important

[2] Source: www.yahoo.com

[3] Source: www.google.com



hotels in redondo beach

About 15,000,000 results (0.29 seconds)

Ads - Why these ads?

Hotels Redondo Beach | EmbassySuites.Hilton.com
embassysuites.hilton.com/**Redondo**
All-Suite **Hotel** 5 mi: **Redondo Beach** Free Breakfast, drinks. $129/Night!

Redondo Beach Hotels CA - Lowest price guarantee
www.booking.com/**Redondo-Beach-Hotels**
Book your **Hotel in Redondo Beach** CA
Most Popular Hotels - Budget Hotels - Best Reviewed Hotels - Luxury Hotels

Hotel Deals at Expedia - Enjoy Your Trip to **Redondo Beach**
www.expedia.co.in/**Redondo-Beach**
Expedia Guarantees the Best Price.
Expedia Recommends - Best Ratings - Budget Hotels - Luxury Hotels

Ads - Why these ads?

Redondo Beach Hotels
www.**hotels**.com
Book **Redondo Beach Hotels**.
Over 140,000 **Hotels** Worldwide.

Redondo Beach Lux **Hotel**
www.crowneplaza.com
Overlooking the Pacific Ocean and
King Harbor Marina. 7 MI to (LAX).

Redondo Beach Hotels
hotels.tripbase.com/**Redondo-Beach**
Compare Rates with Tripbase

Figure 1.4.2: Google Ads

since the number of arms (or ads, in this case) involved is very high.

A set of recommended films at the online movie database IMDB is seen in figure 1.4.3 [4] . This problem of selecting recommended items pertaining to some query is extensively researched under the title 'Recommender Systems'. Multi-arm bandit formulations are widely seen in recommender systems as well. Contextual information from the film names and categories can further improve the recommendations.

But the popular online shopping store Amazon selects recommended articles through previous user behavior as seen in figure 1.4.4 [5]. Does this account for a deterministic solution, with no need for any explore-exploit situation like in multi-arm bandits? No. The user frequented buckets of articles bought is usually large and the selection of top three recommended articles does need to be learnt. Thus, the multi-arm bandit problem relates to applications that have far-reaching significance and value additions.

1.5 Related Work

The multi-arm bandit problem captures general aspects of learning in an unknown environment [Berry & Fristedt (1985)]. Decision theoretic issues like how to minimize learning time, and when to stop learning and start exploiting the knowledge acquired are well embraced in the multi-arm bandit problem. This problem is of useful concern in different areas of artificial intelligence such as reinforcement learning [Sutton & Barto (1998)] and evolutionary programming [Holland (1992)]. The problem also

[4] Source: www.imdb.com

[5] Source: www.amazon.com

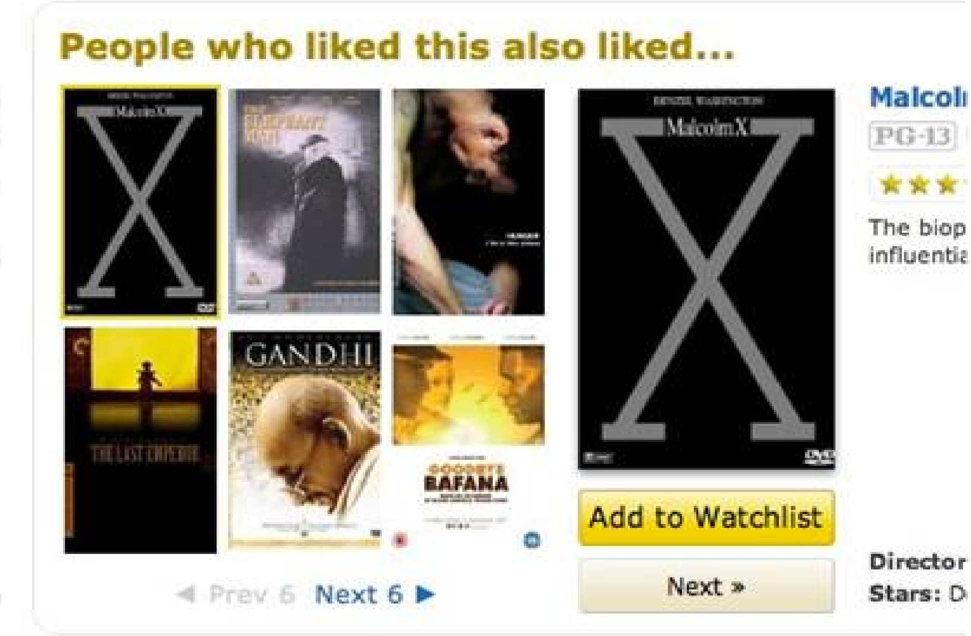
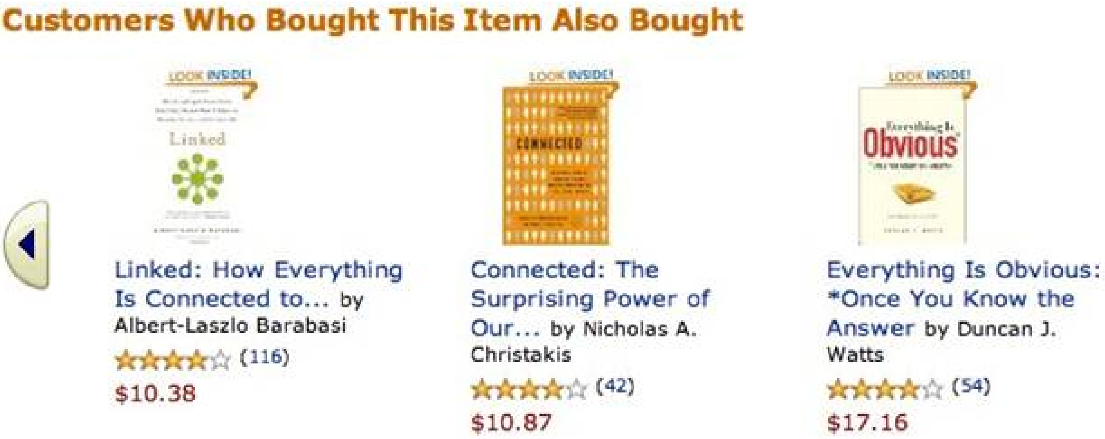


Figure 1.4.3: IMDB Movie Recommendations


Customers Who Bought This Item Also Bought

Linked: How Everything Is Connected to... by Albert-Laszlo Barabasi
★★★★☆ (116)
$10.38

Connected: The Surprising Power of Our... by Nicholas A. Christakis
★★★★☆ (42)
$10.87

Everything Is Obvious: *Once You Know the Answer by Duncan J. Watts
★★★★☆ (54)
$17.16

Figure 1.4.4: Amazon Product Recommendations based on customer behavior

has applications in many fields including industrial engineering, simulation and evolutionary computation. For instance, Kim & Nelson (2001) talk about applications with regard to ranking, selection and multiple comparison procedures in statistics and Schmidt et al. (2006) entail applications in evolutionary computation.

The traditional objective of the bandit problem is to maximize total reward given a specified number of pulls, l, to be made hence minimizing regret. Lai & Robbins (1985) showed that the regret should grow at least logarithmically and provided policies that attain the lower bound for specific probability distributions. Agrawal (1995) provided policies achieving the logarithmic bounds incorporating sample means that are computationally more efficient and Auer et al. (2005) described policies that achieve the bounds uniformly over time rather than only asymptotically. Meanwhile, Even-Dar et al. (2006) provided another quantification of the objective measuring quickness in determining the best arm. A Probably Approximately Correct (PAC) framework is incorporated quantifying the identification of an ϵ-optimal arm with probability $1 - \delta$. The objective is then to minimize the sample complexity l, the number of samples required for such an arm identification. Even-Dar et al. (2006) describe *Median Elimination Algorithm* achieving $O(n)$ sample complexity. This is further extended to finding m arms that are ϵ-optimal with high probability by Kalyanakrishnan & Stone (2010).

We propose a family of learning algorithms for the Bandit problem and show the family contains not only algorithms attaining $O(n)$ sample complexity but also algorithms that achieve the theoretically lowest regret of $O(\log(t))$. In addition, the algorithm presented allows to control learning to reduce sample complexity or regret, through a single parameter that it takes as input. To the best of our knowledge, this is the first work to introduce control on sample complexity or regret while learning. We address the n-arm bandit problem with generic probability distributions without any restrictions on the means, variances or any higher moments that the distributions may possess. Experiments show the proposed algorithms perform substantially better compared to state-of-the-art algorithms like Median Elimination Algorithm [Even-Dar et al. (2006)] and UCB-Revisited [Auer & Ortner (2010)]. While Even-Dar et al. (2006) and Auer & Ortner (2010) provide algorithms that are not parameterized for tunability, we propose a single-parametric algorithm that is based on fractional moments[6] of rewards acquired. To the best of our knowledge ours is the first work[7] to

[6] The i^{th} moment of a random variable R is defined as $E[R^i]$. Fractional moments occur when the exponent i is fractional (rational or irrational).

[7] A part of this work appeared recently in UAI 2011 [Narayan & Ravindran (2011)]

use fractional moments in bandit problems, while recently we discovered its usage in the literature in other contexts. Min & Chrysostomos (1993) describe applications of such fractional (low order) moments in signal processing. It has been employed in many areas of signal processing including image processing [Achim et al. (2005)] and communication systems [Xinyu & Nikias (1996); Liu & Mendel (2001)] mainly with regard to achieving more stable estimators. We theoretically show that the proposed algorithms can be controlled to attain $O(n)$ sample complexity in finding an ϵ-optimal arm or to incur the optimal logarithmic regret, $O(\log(t))$. We also discuss the controllability of the algorithm. Experiments support these theoretical findings showing the algorithms incurring substantially low regrets while learning. A brief overview of what is presented: chapter 2 describes motivations for the proposed algorithms, followed by theoretical analysis of optimality and sample complexity in chapter 3. Then we prove the proposed algorithms incur the theoretical lower bound for regret shown by Lai & Robbins (1985) in chapter 4. We then provide a detailed analysis of the algorithms presented with regard to their properties, performance and tunability in chapter 5. Finally, experimental results and observations are presented in chapter 6 followed by conclusions and scope for future work.

Chapter 2

Proposed Algorithm & Motivation

2.1 Notation

Consider a bandit problem with n arms with action a_i denoting choice of pulling the i^{th} arm. An experiment involves a finite number of arm-pulls in succession. An event of selecting action a_i results in a reward that is sampled from a random variable R_i, having a bounded support. In a particular such experiment, consider the event of selecting action a_i for the k^{th} time. Let $r_{i,k}$ be a sample of the reward acquired in this selection.

2.2 Motivation

When deciding about selecting action a_i over any other action a_j, we are concerned about the rewards we would receive. Though estimates of $E(R_i)$ and $E(R_j)$ are indicative of rewards for the respective actions, estimates of variances $E[(R_i - \mu_i)^2]$ and $E[(R_j - \mu_j)^2]$ would give more information in the beginning stages of learning when the confidence in estimates of expectations would be low. In other words, we wouldn't have explored enough for the estimated expectations to reflect true means.

Though mean and variance together provide full knowledge of the stochasticity for some distributions, for instance Gaussian, we would want to handle generic distributions hence requiring to consider additional higher moments. It is the distribution after all, that gives rise to all the moments completely specifying the random variable. Thus we look at a generic probability distribution to model our policy to pull arms.

Consider selection of action a_i over action a_j. For action a_i to have a higher reward than action a_j, the probability $P(R_i > R_j)$ holds, and to compute its estimate we propose the following discrete approximation: After selecting actions a_i, a_j for m_i, m_j times respectively, we can, in general, compute the probability estimate

$$\hat{P}(R_i > R_j) = \sum_{r_{i,k} \in N_i} \left\{ \hat{P}(R_i = r_{i,k}) \sum_{r_{j,k'} \epsilon L^j_{i,k}} \hat{P}(R_j = r_{j,k'}) \right\}$$

where sets N_i and $L^j_{i,k}$ are given by,

$$\begin{aligned} N_i &= \{r_{i,k} : 1 \leq k \leq m_i\} \\ L^j_{i,k} &= \{r_{j,k'} : r_{j,k'} < r_{i,k}, 1 \leq k' \leq m_j\} \end{aligned} \tag{2.2.1}$$

with random estimates $\hat{P}(R_i = r_{i,k})$ of the true probability $P(R_i = r_{i,k})$ calculated by

$$\hat{P}(R_i = r_{i,k}) = \frac{|\{k' : r_{i,k'} = r_{i,k}\}|}{m_i}. \tag{2.2.2}$$

Note that thus far we are only taking into account the probability, ignoring the magnitude of rewards. That is, if we have two instances of rewards, $r_{j,l}$ and $r_{m,n}$ with $P(R_j = r_{j,l}) = P(R_m = r_{m,n})$, then they contribute equally to the probabilities $P(R_i > R_j)$ and $P(R_i > R_m)$, though one of the rewards could be much larger than the other. For fair action selections we need a function monotonically increasing in rewards, and let us then formulate the preference function for action a_i over action a_j as,

$$A_{ij} = \sum_{r_{i,k} \in N_i} \left\{ \hat{P}(R_i = r_{i,k}) \sum_{r_{j,k'} \epsilon L^j_{i,k}} (r_{i,k} - r_{j,k'})^{\beta} \hat{P}(R_j = r_{j,k'}) \right\} \tag{2.2.3}$$

where β determines how far we want to distinguish the preference functions with regard to magnitude of rewards. This way, for instance, arms constituting a higher variance are given more preference. For $\beta = 1$, we would have A_{ij} proportional to,

$$E\left[R_i - R_j | R_i > R_j\right] . P(R_i > R_j)$$

when the estimates $\hat{P}$ approach true probabilties. See that we have started with a simple choice for the monotonic function, the polynomial function, without restricting the exponent. Now, to find among a set of arms, one single arm to pull, we need a quantification to compare different arms. With the introduction of the polynomial dependence on the reward magnitudes, A_{ij}'s are no more probabilities. To find a quantification to base the arm pulls, we propose a preference function to choose a_i over all other actions to be

$$A_i = \prod_{j \neq i} A_{ij}. \tag{2.2.4}$$

2.3 The Algorithm

For an n-arm bandit problem, our proposed class of algorithms first choose each arm l times, in what is called as an initiation phase. After this initiation phase, at all further action selections, where arm i has been chosen for m_i times, finding a reward $r_{i,k}$ when choosing i^{th} arm for the k^{th} time, the class of algorithms computes the sets

$$\begin{aligned} N_i &= \{r_{i,k} : 1 \leq k \leq m_i\} \\ L^j_{i,k} &= \{r_{j,k'} : r_{j,k'} < r_{i,k}, 1 \leq k' \leq m_j\} \end{aligned}$$

and using

$$\hat{P}(R_i = r_{i,k}) = \frac{|\{k' : r_{i,k'} = r_{i,k}\}|}{m_i}$$

computes the indices

$$A_i = \prod_{j \neq i} A_{ij},$$

$$A_{ij} = \sum_{r_{i,k} \in N_i} \left\{ \hat{P}(R_i = r_{i,k}) \sum_{r_{j,k'} \epsilon L^j_{i,k}} (r_{i,k} - r_{j,k'})^\beta \hat{P}(R_j = r_{j,k'}) \right\}$$

to further choose arms.

A new class of Algorithms, based on conditional fractional expectations of rewards acquired, can now use A_i in 2.2.4 to perform action selections. A specific instance that is greedy after the exploring initiation phase (Algorithm 2.1), henceforth referred to as Fractional Moments on Bandits (FMB), picks arm i that has the highest A_i. All analysis presented in this work are with reference to the algorithm FMB. A probabilistic variant, picking arm i with probability A_i(normalized) is also proposed, but not analyzed. We call this probabilistic action selection on the quantities A_i, or simply pFMB. The class of algorithms are based on the set of rewards previously acquired but can

Algorithm 2.1 Fractional Moments on Bandits (FMB)

```
INPUT: $l, \beta$
INITIALIZATION: Choose each arm $l$ times
DEFINE: $r_{i,k}$, the reward acquired for $k^{th}$ selection of arm $i$, and $m_i$, the number of selections made for arm $i$, and the sets $N_i = \{r_{i,k} : 1 \leq k \leq m_i\}$and $L^j_{i,k} = \{r_{j,k'} : r_{j,k'} < r_{i,k}, 1 \leq k' \leq m_j\}$
LOOP:
    1. $\hat{p}_{ik} = \hat{P}(R_i = r_{i,k}) = \frac{|\{k' : r_{i,k'} = r_{i,k}\}|}{m_i}$ for $1 \leq i \leq n$, $1 \leq k \leq m_i$
    2. $A_{ij} = \sum_{r_{i,k} \in N_i} \left\{ \hat{p}_{ik} \sum_{r_{j,k'} \epsilon L^j_{i,k}} (r_{i,k} - r_{j,k'})^\beta \hat{p}_{jk'} \right\}$ for $1 \leq i, j \leq n$, $i \neq j$
    3. $A_i = \prod_{j \neq i} A_{ij}$ $\forall 1 \leq i \leq n$
    4. Pull arm $i$ that has the highest $A_i$
```

be incrementally implemented (Results discussed in chapter 6 use an incremental implementation). Also, the algorithms are invariant to the type of reward distributions. Besides incremental implementations, the algorithms and the associated computations simplify greatly when the underlying reward is known to follow a discrete probability distribution.

The family (of algorithms FMB, for various values of l, β)[1] is shown to contain algorithms that achieve a sample complexity of $O(n)$ in chapter 3. The algorithm 2.1 is also shown to incur the asymptotic theoretical lower bound for regret in chapter 4. Further, the algorithm is analyzed in detail in chapter 5.

[1] Note that FMB, pFMB, and any other algorithm that uses A_i for action selection constitute a class of algorithms. FMB for varying input parameters constitute a family of algorithms.

Chapter 3

Sample Complexity

In this chapter, we present theoretical analysis of sample complexity on the proposed algorithms. We show that given ϵ and δ, a sample complexity, as defined in 3, that is of the order $O(n)$ exists for the proposed algorithm FMB.

3.1 Algorithm formulation

Our problem formulation considers a set of n bandit arms with R_i representing the random reward on pulling arm i. We assume that the reward is binary, $R_i \in \{0, r_i\} \forall i$. Since the proposed algorithm assumes a discrete approximation for the reward probability distributions, an analysis on the bernoulli formulation is easily extendible further. This is true even when true reward distributions are continuous, as the algorithm always functions on the discrete approximation on the rewards it has already experienced. Also, denote $p_i = P\{R_i = r_i\}$, $\mu_i = E[R_i]$=$r_i p_i$, and let us have for simplicity,

$$\mu_1 > \mu_2 > \cdots > \mu_n. \tag{3.1.1}$$

Define,

$$I_{ij} = \begin{cases} 1 & if\ r_i > r_j \\ 0 & otherwise \end{cases}$$

and,

$$\delta_{ij} = \begin{cases} \frac{r_j}{r_i} & if\ r_i > r_j \\ 1 & otherwise \end{cases}.$$

We then have from 2.2.3, 2.2.2 and 2.2.1,

$$\begin{aligned} A_{ij} &= I_{ij}\hat{p_i}\hat{p_j}(r_i - r_j)^{\beta} + \hat{p_i}(1-\hat{p_j})(r_i - 0)^{\beta} \\ &= \hat{p_i}\hat{p_j}(r_i - \delta_{ij}r_i)^{\beta} + \hat{p_i}(1-\hat{p_j})r_i^{\beta} \\ &= \hat{p_i}r_i^{\beta}\big\{1 + \hat{p_j}[(1-\delta_{ij})^{\beta} - 1]\big\}. \end{aligned}$$

The action selection function 2.2.4 becomes,

$$A_i = (\hat{p_i}r_i^{\beta})^{n-1}\prod_{j\neq i}\big\{1 + \hat{p_j}[(1-\delta_{ij})^{\beta} - 1]\big\}. \tag{3.1.2}$$

We now discuss Chernoff bounds and its extension to dependent random variables before analysis of the proposed algorithm.

3.2 Chernoff Hoeffding Bounds

We start with a brief introduction to the Chernoff Hoeffding bounds on a set of independent random variables.

Lemma 4. *Hoeffding (1963) Let $X_1, X_2, \dots X_n$ be random variables with common range $[0,1]$ and such that $E[X_t|X_1,\dots,X_{t-1}] = \mu$ for $1 \leq t \leq n$. Let $S_n = \frac{X_1+\cdots+X_n}{n}$. Then for all $a \geq 0$ we have the following,*

$$\begin{aligned} P\{S_n &\geq \mu + a\} \leq \exp(-2a^2 n) \\ P\{S_n &\leq \mu - a\} \leq \exp(-2a^2 n). \end{aligned}$$

3.2.1 Modified Chernoff Hoeffding Bounds

Chernoff Hoeffding bounds on a set of dependent random variables is discussed here for analysis of the proposed algorithm. We start by introducing Chromatic and Fractional Chromatic Number of Graphs.

Definition 5. A proper vertex coloring of a graph $G(V, E)$ is assignment of colors to vertices such that no two vertices connected by an edge have the same color. Chromatic number $\chi(G)$ is defined to be the minimum number of distinct colors required for proper coloring of a graph.

A k-fold proper vertex coloring of a graph is assignment of k distinct colors to each vertex such that no two vertices connected by an edge share any of their assigned colors. k-fold Chromatic number $\chi_k(G)$ is defined to be the minimum number of distinct colors required for k-fold proper coloring for a graph. The Fractional Chromatic number is then defined as,

$$\chi'(G) = \lim_{k\to\infty} \frac{\chi_k(G)}{k}.$$

Clearly, $\chi'(G) \leq \chi(G)$. We now look at Chernoff Hoeffding Bounds when the random variables involved are dependent.

Lemma 6. *Dubhashi & Panconesi (2009) Let $X_1, X_2, \dots X_n$ be random variables, some of which are independent, have common range $[0,1]$ and mean μ. Define $S_n = \frac{X_1+\dots+X_n}{n}$ and a graph $G(V,E)$ with $V = \{1,2,\dots,n\}$. Edges connect vertices i and j, $i \neq j$ if and only if X_i is dependent on X_j. Let $\chi'(G)$ denote the fractional chromatic number of graph G. Then for all $a \geq 0$ we have the following,*

$$\begin{aligned} P\{S_n &\geq \mu + a\} \leq \exp(-2a^2 n/\chi'(G)) \\ P\{S_n &\leq \mu - a\} \leq \exp(-2a^2 n/\chi'(G)). \end{aligned}$$

We now discuss bounds on the proposed formulation that incorporates dependent random variables.

Definition 7. Define an n-tuple set $\mathcal{S} = \{(\nu_1, \nu_2, \dots, \nu_n) : 1 \leq \nu_k \leq m_k, m_k \in \mathbb{N} \forall 1 \leq k \leq n\}$ and a one to one function $f : S \to \{1, 2, \dots, q : q = \prod_{k=1}^{n} m_k\}$ that finds a unique index for every element in $\mathcal{S}$. For each $\mathbf{e} \in S$, define random variable U_i with $i = f(\mathbf{e})$, so that for $\mathbf{e_1}, \mathbf{e_2} \in \mathcal{S}$, U_{i_1} and U_{i_2} are dependent if and only if atleast one dimension of $\mathbf{e_1} - \mathbf{e_2}$ is null. Construct a graph $G(V,E)$ on $V = \{U_i : 1 \leq i \leq q\}$ with edges connecting every pair of dependent vertices.

Remark 8. Consider n arms where k^{th} arm is sampled for m_k times. Let $X_u^{(k)}$ be a random variable, with mean $\mu^{(k)}$, related to the reward acquired on the u^{th} pull of k^{th} arm. Then, for $\{m_k : 1 \leq k \leq n\}$, $X_1^{(k)}, X_2^{(k)}, \dots X_{m_k}^{(k)}$ are random variables (with common range within $[0,1]$, say) such that $E[X_u^{(k)} | X_1^{(k)}, \dots, X_{u-1}^{(k)}] = \mu^{(k)} \forall k, u : 1 \leq u \leq m_k$ and $X_u^{(i)}$ is independent of $X_v^{(j)}$ for $i \neq j$. Define a random variable $S_{m_k}^{(k)} = \frac{X_1^{(k)}+\dots+X_{m_k}^{(k)}}{m_k}$ capturing the estimate of $X^{(k)}$. Define a random variable $T = \prod_k S_{m_k}^{(k)}$

and consequently on expanding this product we will find,

$$T = \frac{\sum_1^q U_i}{q}$$

with mean $\mu_T = E[T] = \prod_k \mu^{(k)}$, where U_i have properties described in Definition 7 with its associated graph $G(V, E)$.

Corollary 9. *For a sum of dependent random variables, U_i, and an associated graph $G(V, E)$, as defined in Definition 7, and a random variable*

$$T = \frac{\sum_1^q U_i}{q}$$

we have for all $a \geq 0$,

$$\begin{aligned} P\{T &\geq \mu_T + a\} \leq \exp\left(\frac{-2a^2q}{\chi'(G)}\right) \leq \exp\left(\frac{-2a^2q}{\chi(G)}\right) \\ P\{T &\leq \mu_T - a\} \leq \exp\left(\frac{-2a^2q}{\chi'(G)}\right) \leq \exp\left(\frac{-2a^2q}{\chi(G)}\right) \end{aligned}$$

with mean $\mu_T = E[T] = \prod_k \mu^{(k)}$ and $\chi(G)$ and $\chi'(G)$ the chromatic and fractional chromatic numbers of graph G respectively.

3.3 Bounds on Sample Complexity

Theorem 10. *For a n-arm bandit problem and a given ϵ and δ, algorithm FMB is an (ϵ, δ)-PAC algorithm, incurring a sample complexity of*

$$O\left(n\left(\frac{1}{\mu_T^4}\ln^2\left(\frac{1}{\delta}\right)\right)^{\frac{1}{n}}\right)$$

where

$$\mu_T = \min_{i:\mu_i<\mu^*-\epsilon} |E[A_i - A^*]|$$

with A_i defined in 2.2.4, 2.2.3, 2.2.2 and 2.2.1, and μ^*, A^* being those values that correspond to the optimal arm.

Following constraint 3.1.1 for simplicity, we will now prove the above theorem. To start with, we will state a few lemmas that would be helpful in proving theorem 10.

Lemma 11. *For a graph $G(V,E)$ constructed on vertices U_i pertainining to Definition 7, the chromatic number is bounded as*

$$\chi(G) \leq 1 + \sqrt{n\Big(q - \prod_k (m_k - 1)\Big)}.$$

Proof. See that $|V| = q$, and each vertex is connected to $q - \prod_k (m_k - 1)$ other vertices. Then, the total number of edges in the graph becomes, $|E| = \frac{1}{2} n \left(q - \prod_k (m_k - 1)\right)$.

The number of ways a combination of two colors can be picked is given by, ${}^{\chi(G)}C_2 = \frac{1}{2}\chi(G)\left(\chi(G) - 1\right)$. For optimal coloring, there must exist at least one edge connecting any such pair of colors picked, and we have

$$\begin{aligned} \frac{1}{2}\chi(G)\left(\chi(G)-1\right) &\leq |E| \\ \chi(G)\left(\chi(G)-1\right) &\leq n\Big(q - \prod_k (m_k - 1)\Big). \end{aligned}$$

With $\left(\chi(G) - 1\right)^2 \leq \chi(G)\left(\chi(G) - 1\right)$, this turns into

$$\begin{aligned} \left(\chi(G)-1\right)^2 &\leq n\Big(q - \prod_k (m_k - 1)\Big) \\ \chi(G) &\leq 1 + \sqrt{n\Big(q - \prod_k (m_k - 1)\Big)}. \end{aligned}$$

□

Next, we state a lemma on the boundedness of a function which we will come across later in proving theorem 10.

Lemma 12. *For $n \in \mathbb{N}$, and a function*

$$g(n) = (n \ln^2 n)^{\frac{1}{n}},$$

$\exists n_0 \in \mathbb{N}$ *such that* $g(n)$ *is upper bounded* $\forall n > n_0 \in \mathbb{N}$.

Proof. This follows from,

$$\begin{aligned} \ln g(n) &= \frac{\ln(n \ln^2 n)}{n} = \frac{\ln n + 2\ln\ln n}{n} \\ \implies \frac{1}{g(n)} g'(n) &= \frac{1 + \frac{2}{\ln n} - (\ln n + 2\ln\ln n)}{n^2}. \end{aligned}$$

Thus, g is a decreasing function for $n > n_0 \in \mathbb{N}$ with the limit at infinity governed by

$$\begin{aligned}
\lim_{n\to\infty} \ln(g(n)) &= \lim_{n\to\infty} \frac{\ln n + 2\ln\ln n}{n} \\
&= \lim_{n\to\infty} \frac{\frac{\partial}{\partial n}(\ln n + 2\ln\ln n)}{\frac{\partial}{\partial n} n} \\
&= \lim_{n\to\infty} \left(\frac{1}{n} + \frac{2}{n\ln n}\right) \\
&= 0 \\
\implies \lim_{n\to\infty} g(n) &= 1.
\end{aligned}$$

□

Now we prove *theorem* 10 with the help of the lemmas defined above.

Proof. of theorem 10. Let $i \neq 1$ be an arm which is not ϵ-optimal, $\mu_i < \mu_1 - \epsilon$. To prove that FMB is an (ϵ, δ)-PAC algorithm, we need to bound for all i, the probability of selecting a non-ϵ-optimal arm, $P(A_i > A_j : \mu_i < \mu_1 - \epsilon, \forall j)$, which is in turn bounded as

$$P(A_i > A_j : \mu_i < \mu_1 - \epsilon, \forall j) \leq P(A_i > A_1 : \mu_i < \mu_1 - \epsilon)$$

since arm 1 is optimal. In other words, we need to bound the probability of event

$$A_i > A_1$$

to restrict the policy from choosing non-ϵ-optimal arm i, with A_i given by 3.1.2. See that A_i is in the form of $\prod_k S^{(k)}_{m_k}$ in remark 8. Defining the random variable $T_i = A_i - A_1$, we expand the products, to have

$$T_i = \frac{\sum_1^q U_k}{q}$$

where U_k have properties described in definition 7 with its associated graph $G(V, E)$. So we have a sum of $q = \prod_j m_j$ dependent random variables U_k, with a true mean $\mu_{T_i} = E[T_i] = E[A_i] - E[A_1]$. Applying corollary 9 to T_i, to bound the probability of the event $T_i \geq 0$, we have using the modified Chernoff-Hoeffding Bounds,

$$P(T_i \geq 0) \;=\; P(T_i \geq \mu_{T_i} - \mu_{T_i}) \leq \exp(-2\mu_{T_i}^2 q/\chi(G)).$$

From definition 7, we then have,

$$P(T_i \geq 0) \quad \leq \quad \exp\left(\frac{-2\mu_{T_i}^2 q}{1+\sqrt{n\left(q-\prod_k (m_k-1)\right)}}\right). \tag{3.3.1}$$

In a simple case when each arm is sampled l times, $m_k = l \; \forall k$ and

$$\begin{aligned} P(T_i \geq 0) \quad &\leq \quad \exp\left(\frac{-2\mu_{T_i}^2 l^n}{1+\sqrt{n\left(l^n-(l-1)^n\right)}}\right) \\ &\leq \quad \exp\left(\frac{-2\mu_{T_i}^2 l^n}{2\sqrt{nl^n}}\right) \\ &\leq \quad \exp\left(\frac{-\mu_{T_i}^2 l^{n/2}}{\sqrt{n}}\right). \end{aligned}$$

To get the sample complexity, sample each arm so that

$$\begin{aligned} \exp\left(\frac{-\mu_T^2 l^{n/2}}{\sqrt{n}}\right) \quad &= \quad \frac{\delta}{n} \\ \implies l \quad &= \quad \left(\frac{n}{\mu_T^4}\ln^2\left(\frac{n}{\delta}\right)\right)^{\frac{1}{n}} \end{aligned} \tag{3.3.2}$$

where

$$\mu_T \quad = \quad \min_{i:\mu_i<\mu_1-\epsilon} |\mu_{T_i}|. \tag{3.3.3}$$

Sample complexity corresponding to all arms is then

$$nl = n\left(\frac{n}{\mu_T^4}\ln^2\left(\frac{n}{\delta}\right)\right)^{\frac{1}{n}}.$$

We can improve the complexity further through successive elimination method discussed by Even-Dar et al. (2006). But instead, from lemma 12, we see that $\exists C$: $g(n) = (n\ln^2 n)^{\frac{1}{n}} < C \forall n > n_0 \in \mathbb{N}$. With some numerical analysis, we see that $\forall n > 5$, $g'(n) < 0$, and that g attains a maximum at $n = 5$, with $g(n) = 1.669$. A plot of g versus n is shown in Figure 1. So $g(n) < 1.67 \; \forall n \in \mathbb{N}$ and hence we have

$$O(n(n\ln^2 n)^{\frac{1}{n}}) = O(n).$$

Hence FMB finds ϵ-optimal arms with probability $1-\delta$ incurring a sample complexity

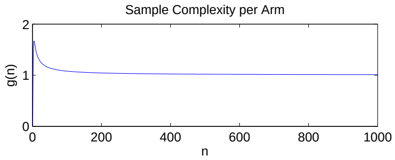


Figure 3.3.1: $g(n)$ versus n

of

$$O\left(n\left(\frac{1}{\mu_T^4}\ln^2\left(\frac{1}{\delta}\right)\right)^{\frac{1}{n}}\right).$$

$\square$

Thus, sample complexity dependence on n, of the proposed algorithms is essentially $O(n)$ since $O(nc^{1/n}) = O(n)$ for any constant c. The sample complexity depends on ϵ through μ_T and this is discussed in detail in 5.1.

Chapter 4

Regret Analysis

In this chapter, we show that FMB (Algorithm 2.1) incurs a regret of $O(\log(t))$, a theoretical lower bound shown by Lai & Robbins (1985). We will start with the definition of regret and then state and prove the regret bounds of FMB.

4.1 Definition of Regret

Define random variable $F_i(t)$ to be the number of times arm i was chosen in t plays. The expected regret, η is then given by

$$\eta(t) = (n - E[F_1(t)])\mu_1 - \sum_{i \neq 1} \mu_i E[F_i(t)]$$

which is,

$$\begin{aligned} \eta(t) &= \mu_1 \sum_{i \neq 1} E[F_i(t)] - \sum_{i \neq 1} \mu_i E[F_i(t)] \\ &= \sum_{i \neq 1} (\mu_1 - \mu_i) E[F_i(t)] \\ &= \sum_{i \neq 1} \Delta_i E[F_i(t)], \qquad (4.1.1) \end{aligned}$$

say.

4.2 Bounds on Regret

We will now state and prove the boundedness of regret incurred by FMB.

Theorem 13. *The total expected regret $\eta(t)$, for t trials in an n-arm bandit problem, incurred by algorithm FMB is upper bounded by*

$$\sum_{i \neq 1} \Delta_i O\left(l + \log(t)\right)$$

where l is an input parameter concerning the initiation phase of the algorithm.

To prove theorem 13, we will now state a few lemmas that would be helpful. To start with, we restate some of the definitions that were used in chapter 3 for convenience.

Definition 14. For an n-arm bandit problem where arm i has been chosen for m_i times (including the choice of each arm l times in the initiation phase), finding a reward $r_{i,k}$ when choosing i^{th} arm for the k^{th} time, FMB computes the sets

$$\begin{aligned} N_i &= \{r_{i,k} : 1 \leq k \leq m_i\} \\ L^j_{i,k} &= \{r_{j,k'} : r_{j,k'} < r_{i,k}, 1 \leq k' \leq m_j\} \end{aligned}$$

and using

$$\hat{P}(R_i = r_{i,k}) = \frac{|\{k' : r_{i,k'} = r_{i,k}\}|}{m_i}$$

computes the indices

$$A_i = \prod_{j \neq i} A_{ij},$$

$$A_{ij} \;=\; \sum_{r_{i,k} \in N_i} \left\{ \hat{P}(R_i = r_{i,k}) \sum_{r_{j,k'} \epsilon L^j_{i,k}} (r_{i,k} - r_{j,k'})^{\beta} \hat{P}(R_j = r_{j,k'}) \right\}$$

to further choose arms. While doing this to find an ϵ-optimal arm with probability $1 - \delta$, FMB incurs a sample complexity of

$$O\left(n \left(\frac{1}{\mu_T^4} \ln^2 \left(\frac{1}{\delta} \right) \right)^{\frac{1}{n}} \right)$$

where

$$\begin{aligned}\mu_T &= \min_{i:\mu_i<\mu^*-\epsilon} |\mu_{T_i}| \\ \mu_{T_i} &= E[T_i] \\ T_i &= A_i - A^*\end{aligned}$$

with $\mu^*$and A^* being those values that correspond to the optimal arm.

Corollary 15. *For a learning experiment completing $\tau-1$ plays on an n-arm bandit problem, with arm i being chosen m_i times, the probability that applying algorithm FMB for the τ^{th} play would result in choice of a non-ϵ-optimal arm is given by 3.3.1. Note that m_i is a function of τ in itself and 3.3.1 presumes knowledge of $m_i(\tau)$ is made available. See that $m_i(\tau)$ for a particular such experiment is a sample from the random variable $F_i(\tau)$. For the purpose of the regret analysis, denote A_i computed for the τ^{th} play by $A_i(\tau)$ and the corresponding m_i by $m_i(\tau)$ and T_i by $T_i(\tau)$. Rewrite 3.3.1 as*

$$P(T_i(\tau)\geq 0|m_i(\tau)) \leq \exp\left(\frac{-2\mu_{T_i}^2 q(\tau)}{1+\sqrt{n\left(q(\tau)-\prod_k(m_k(\tau)-1)\right)}}\right) \tag{4.2.1}$$

where $q(\tau)=\prod_k m_k(\tau)$ and μ_{T_i} conforms to definition 14.

Lemma 16. *Given the event $F_i(t-1)=y$, the probability that FMB selects arm i at t^{th} play is bounded as,*

$$P\{arm\ i\ selected\ at\ t|F_i(t-1)y\} \leq \exp\left(\frac{-\mu_{T_i}^2\sqrt{y(t-y-1-l(n-2))l^{n-2}}}{\sqrt{n}}\right)$$

where l, μ_{T_i} conform to definition 14.

Proof. See that

$$\begin{aligned}P\{arm\ i\ selected\ at\ t\} &= P\{A_i(t)>A_j(t)\forall j\} \\ &\leq P\{A_i(t)>A_1(t)\}\end{aligned}$$

Given any partial information about $F_i(\tau)$ for some $\tau<t$ and some i, this probability can be further bounded. Given that the event $F_i(t-1)=y$ is known to have occured,

this bound improves as

$$\begin{aligned} P\{arm\ i\ selected\ at\ t|F_i(t-1)=y\} &= P\{A_i(t) > A_j(t)\forall j|F_i(t-1)=y\} \\ &\leq P\{A_i(t) > A_1(t)|F_i(t-1)=y\} \\ &\leq P\{T_i(t) = A_i(t) - A_1(t) > 0|F_i(t-1)=y\} \\ &\leq P\{T_i(t) > 0|m_i(t) \in S_{m(t)}\} \qquad (4.2.2) \end{aligned}$$

with set $S_{m(t)}$ given by

$$S_{m(t)} = \{(m_1(t), m_2(t), \ldots, m_k(t), \ldots, m_n(t))|m_i(t) = y, \sum_k m_k(t) = t-1\}.$$

So we further bound,

$$P\{A_i(t) > A_1(t)|F_i(t-1)=y\} \leq \max_{S_{m(t)}}\{P(T_i(t) \geq 0)\}. \qquad (4.2.3)$$

Now, the upper bound of $P(T_i(t) \geq 0)$ would be maximum when $\prod_k m_k$ is minimum. This is observed from (3.3.1) as

$$\begin{aligned} P(T_i(t) \geq 0) &\leq \exp\left(\frac{-2\mu_{T_i}^2 q}{1+\sqrt{n\left(q - \prod_k (m_k(t)-1)\right)}}\right) \\ &\leq \exp\left(\frac{-2\mu_{T_i}^2 q}{1+\sqrt{nq}}\right) \\ &\leq \exp\left(\frac{-\mu_{T_i}^2 \sqrt{q}}{\sqrt{n}}\right) \\ &= \exp\left(\frac{-\mu_{T_i}^2 \sqrt{\prod_k m_k(t)}}{\sqrt{n}}\right). \qquad (4.2.4) \end{aligned}$$

The minimum that $\prod_k m_k(t)$ would take for a given $\sum_k m_k(t) = t-1$ is when $m_i(t)$ are most unevenly distributed. When each arm is chosen l times in the initiation phase of the algorithm, we would have

$$\prod_k m_k(t) \geq y(t-1-l(n-2)-y) \cdot l^{n-2} \qquad (4.2.5)$$

which occurs when all the instances of arm choices (other than $m_i(t) = y$ and the initiation phase) correspond to a single particular arm other than i, say j. From

(4.2.2), (4.2.3), (4.2.4) and (4.2.5), we would have

$$\begin{aligned}
P\{arm\ i\ selected\ at\ t|F_i(t-1)y\} &\leq P\{A_i(t) > A_1(t)|F_i(t-1)=y\} \\
&\leq P\left(T_i(t) \geq 0|m_k(t) = \begin{cases} y & if\ k=i \\ t-1-l(n-2)-y & if\ k=j \\ l & otherwise \end{cases}\right) \\
&\leq \exp\left(\frac{-\mu_{T_i}^2\sqrt{y(t-y-1-l(n-2))l^{n-2}}}{\sqrt{n}}\right).
\end{aligned}$$

□

Lemma 17. *For $a, b \in \mathbb{N}$, the sum of square roots of consequtive numbers is lower bounded as*

$$\sum_{h=a}^{b} \sqrt{h} \geq \frac{2}{3}(b^{3/2} - (a-1)^{3/2}).$$

Proof. The proof uses bounds on integrals as,

$$\begin{aligned}
\sum_{h=a}^{b} \sqrt{h} &\geq \int_{a-1}^{b} \sqrt{\upsilon} d\upsilon \\
&= \frac{2}{3}(b^{3/2} - (a-1)^{3/2}).
\end{aligned}$$

□

Using the above stated lemmas, we now prove theorem *13*.

Proof. of Theorem 13. From (4.1.1), we see that to bound the expected regret is to bound $E[F_i(t)]$ for all $i \neq 1$. Then the event $\{arm\ i\ selected\ at\ t\}$ is $\{A_i(t) > A_j(t)\forall j\}$ of course with $j \neq i$. Consider

$$\begin{aligned}
E[F_i(t)] &= E\left[E[F_i(t)|F_i(t-1)=y]\right] \\
&= E[(y+1)\, P\left\{A_i(t) > A_j(t)\forall j|F_i(t-1)=y\right\} \\
&\quad +y(1-P\{A_i(t) > A_j(t)\forall j|F_i(t-1)=y\})] \\
&\leq E[(y+1)P\{A_i(t) > A_j(t)\forall j|F_i(t-1)=y\} + y]. \qquad (4.2.6)
\end{aligned}$$

where the first step uses law of total (or iterated) expectations. Choose ϵ such that there exists only one ϵ-optimal arm, then the summation in (4.1.1) goes through only non-ϵ-optimal arms and for all such i, we can use the bounds on $P(T_i(t) \geq 0|m_i(t))$ from corollary 15 and lemma 16. Hence bounding the probability of the event $P\{arm\ i\ selected\ at\ t|F_i(t-1)=y\}$, (4.2.6) turns into,

$$E[F_i(t)] \leq E\left[(y+1)\exp\left(\frac{-\mu_{T_i}^2\sqrt{y(t-y-1-l(n-2))l^{n-2}}}{\sqrt{n}}\right)+y\right].$$

Denoting $\frac{\mu_{T_i}^2}{\sqrt{n}}$ as a positive constant λ_i and using $E[y] = E[F_i(t-1)]$, we have the relation

$$E[F_i(t)] \leq E[F_i(t-1)] + E[(y+1)\exp(-\lambda_i\sqrt{y(t-y-1-l(n-2))l^{n-2}})].$$

Unrolling the above recurrence gives

$$\begin{aligned} E[F_i(t)] &\leq E[F_i(nl)] + \sum_{w=nl+1}^{t} E[(y_w+1)\exp(-\lambda_i\sqrt{y_w(w-y_w-1-l(n-2))l^{n-2}})] \\ &\leq l + \sum_{w=nl+1}^{t} E[(y_w+1)\exp(-\lambda_i\sqrt{y_w(w-y_w-1-l(n-2))l^{n-2}})] \end{aligned}$$

where y_w corresponds to number of times the arm under consideration was chosen in $w-1$ plays. Defining

$$G_i(w) = E[(y_w+1)\exp(-\lambda_i\sqrt{y_w(w-y_w-1-l(n-2))l^{n-2}})]$$

we see the regret bounded as,

$$\eta(t) \leq \sum_{i\neq 1}\Delta_i(l+\sum_{w=nl+1}^{t}G_i(w)). \tag{4.2.7}$$

Now,

$$\begin{aligned} G_i(w) &= \sum_{d=l}^{w-1-l(n-1)} [(d+1)\exp(-\lambda_i\sqrt{d(w-d-1-l(n-2))l^{n-2}}) \,\cdot \\ &\qquad P\{\|\tau : A_i(\tau) > A_j(\tau)\forall j, nl+1 \leq \tau \leq w-1\| = d-l\}] \end{aligned} \tag{4.2.8}$$

where the probability is that which corresponds to arm i being selected $d-l$ times between $nl+1^{th}$ and $w-1^{th}$ plays, inclusive of the ends, with $\|S\|$ denoting the cardinality of set S. Denoting the plays (or epochs) in which arm i could have been selected by the indicator variable $I(\tau)$, we have

$$I(\tau) = \begin{cases} 1 & A_i(\tau) > A_j(\tau) \forall j \\ 0 & otherwise \end{cases}.$$

Hence, there are $\binom{w-nl-1}{d-l}$ possible ways of selecting arm i for a total number of $d-l$ times between $nl+1^{th}$ and $w-1^{th}$ plays. Define the set of possible values that $I(\tau)$ can take by $S_I = \{I(\tau) : nl+1 \leq \tau \leq w-1, \ \sum_\tau I(\tau) = d-l\}$. Call as event $\mathcal{E}$, the event $\{\|\tau : A_i(\tau) > A_j(\tau)\forall j, nl+1 \leq \tau \leq w-1\| = d-l\}$. We now bound the probability of event $\mathcal{E}$ in (4.2.8) as

$$\begin{aligned} P(\mathcal{E}) &= \sum_{I \in S_I} P(\mathcal{E}|I)P(I) \\ &\leq \sum_{I \in S_I} P(\mathcal{E}|I) \\ &\leq \|S_I\| \max_{I \in S_I} P(\mathcal{E}|I). \end{aligned}$$

Defining

$$\begin{aligned} P_{max} &= \max_{I \in S_I} P(\mathcal{E}|I), \\ I_{max} &= \arg\max_{I \in S_I} P(\mathcal{E}|I) \end{aligned}$$

we see that $P(\mathcal{E})$ is bounded by

$$P(\mathcal{E}) \leq \binom{w-nl-1}{d-l} P_{max} \tag{4.2.9}$$

with

$$\begin{aligned} P_{max} &= \prod_{\tau : I_{max}(\tau)=1} P\{A_i(\tau) > A_j(\tau)\forall j | I_{max}\} \cdot \\ & \quad \prod_{\tau : I_{max}(\tau)=0} P\{A_i(\tau) < A_j(\tau) \ for \ some \ j | I_{max}\} \\ &\leq \prod_{\tau : I_{max}(\tau)=1} P\{A_i(\tau) > A_j(\tau)\forall j | I_{max}\} \qquad (4.2.10) \end{aligned}$$

where $P\{A_i(\tau) > A_j(\tau)\forall j|I_{max}\} = P(T_i(\tau) \geq 0)$ can be bounded using (4.2.4) as follows. Also,

$$\begin{aligned}\prod_{\tau:I_{max}(\tau)=1} P\{A_i(\tau) > A_j(\tau)\forall j|I_{max}\} &\le \prod_{\tau:I_{max}(\tau)=1} P\{A_i(\tau) > A_1(\tau)|I_{max}\} \\ &\le \exp\left(-\lambda_i \sum_{\tau:I_{max}(\tau)=1} \sqrt{\prod_k m_k(\tau)}\right).\end{aligned}$$

A maximum upper bound would require $\prod_k m_k(\tau)$ to be at its minimum for all τ. This would occur as $I_{max}(\tau) = 1\forall\tau : nl + 1 \le \tau \le nl + (d - l)$, when arm i is chosen repeatedly in the first $d - l$ turns just after the initiation phase. Then, $\prod_k m_k(\tau)$ for $\tau : I_{max}(\tau) = 1$ are given by,

$$\prod_k m_k(\tau) = (l + \tau - nl)l^{n-1}\ \forall\tau : nl + 1 \le \tau \le nl + (d - l).$$

So,

$$\begin{aligned}\prod_{\tau:I_{max}(\tau)=1} P\{A_i(\tau) > A_j(\tau)\forall j|I_{max}\} &\le \exp\left(-\lambda_i\left(\sum_{\tau=nl+1}^{nl+(d-l)} \sqrt{(l+\tau-nl)l^{n-1}}\right)\right) \\ &\le \exp\left(-\lambda_i\left(l^{\frac{n-1}{2}} \sum_{k=l+1}^{d} \sqrt{k}\right)\right) \qquad (4.2.11)\end{aligned}$$

which can be refined further using lemma 17. Using (4.2.9), (4.2.10), (4.2.11) and lemma 17, we bound $G_i(w)$ in (4.2.8) as

$$G_i(w) \le \sum_{d=l}^{w-1-l(n-1)} (d+1)\binom{w-nl-1}{d-l} \exp(-\lambda_i(\sqrt{d(w-d-1-l(n-2))l^{n-2}}+\frac{2}{3}l^{\frac{n-1}{2}}(d^{3/2}-l^{3/2}))). \qquad (4.2.12)$$

Back to the original problem, if we show that $E[F_i(t)]$ grows slower than $\log(t)$ asymptotically, for all i, then it is sufficient to prove the regret is $O(\log(t))$. For this, see that

$$E[F_i(t)] \le l + \sum_{w=nl+1}^{t} G_i(w)) = g(t),$$

say. Then for regret to grow slower than logarithmically in t, it is sufficient to show that the derivative of the upper bound $g(t)$ grows slower than $1/t$. Seeing that $g(t) - g(t-1) = G_i(t)$, it is enough to show that $G_i(t)$ is bounded by $1/t$ asymptotically. Consider $\Gamma_d = (d+1)\binom{w-nl-1}{d-l} \exp(-\lambda_i(\sqrt{d(w-d-1-l(n-2))l^{n-2}}+\frac{2}{3}l^{\frac{n-1}{2}}(d^{3/2}-l^{3/2})))$. $\exists d^* :$

$\Gamma_{d^*} \geq \Gamma_d \forall d \neq d^*$, and so we bound

$$G_i(w) \leq (w - ln)\Gamma_{d^*}.$$

Now

$$\begin{aligned} \lim_{t\to\infty} \frac{G_i(t)}{\frac{1}{t}} &= \lim_{t\to\infty} tG_i(t) \\ &\leq \lim_{t\to\infty} \frac{t(t-ln)(d^*+1)\binom{t-nl-1}{d^*-l}}{\exp(\lambda_i(\sqrt{d^*(t-d^*-1-l(n-2))l^{n-2}} + \frac{2}{3}l^{\frac{n-1}{2}}(d^{*3/2}-l^{3/2})))}. \end{aligned}$$

Since each term in the numerator is bounded by a polynomial while the exponential in the denominator is not,

$$\lim_{t\to\infty} \frac{G_i(t)}{\frac{1}{t}} = 0.$$

Since growth of $E[F_i(t)]$ is bounded by $O(\log(t))$ for all i, we see the proposed algorithm FMB has an optimal regret as characterized by Lai & Robbins (1985), with the total incurred regret till time t given by

$$\sum_{i\neq 1} \Delta_i O\left(l + \log(t)\right).$$

□

Chapter 5

Analysis of the Algorithm FMB

In this chapter, we analyze the proposed algorithm FMB, with regard to its properties, performance and tunability.

5.1 Sample Complexity dependence on ϵ

We now look at the sample complexity incurred for every arm by FMB in 3.3.2 and its dependance on ϵ. Unlike MEA, this dependence of sample complexity is quite different for FMB, and is illustrated in Figure 5.1.1. A decrease in epsilon would lead to increase in complexity only when the set $\{i : \mu_i < \mu_1 - \epsilon\}$ in 3.3.3 changes due to the decrease in ϵ. The set of arms that determine the complexity is related to the true means in the original problem. We also note that this should be the case ideally, in that a decrease in ϵ that does not lead to any decrease in the number of ϵ-optimal arms, should not increase the sample complexity. This is because the algorithm still performs to select exactly the same ϵ-optimal arms, as was the case before the decrease in ϵ.

5.2 Simple Vs. Cumulative Regrets

While cumulative regret, referred to regret in general, is defined in the expected sense,

$$\eta(t) = t\mu^* - E[Z_t]$$

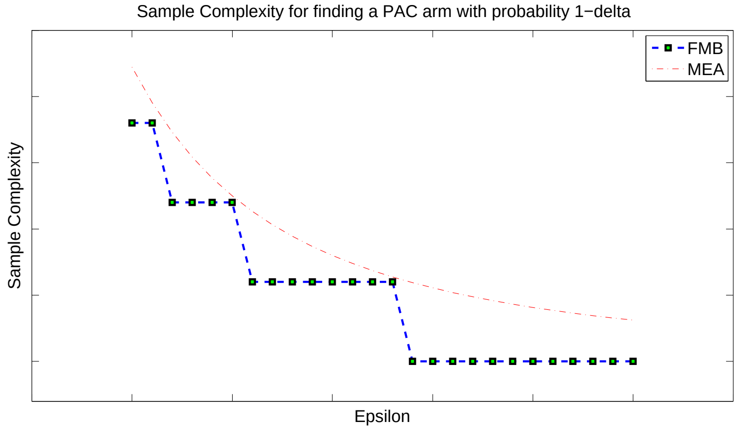


Figure 5.1.1: Comparison of FMB with MEA on the functional dependence of Sample Complexity on ϵ

where Z_t is the total reward acquired for t pulls and $\mu^* = \max_i \mu_i$, there exists another quantification for regret that is related to the sample complexity in some sense. Bubeck et al. (2009) discuss links between cumulative regrets and this new quantification, called simple regret, defined as

$$\phi(t) = \mu^* - \mu_{\psi_t}$$

where $\mu_{\psi_t} = \sum_i \mu_i \psi_{i,t}$ with $\psi_{i,t}$, the probability of choosing arm i as ouput by the policy learning algorithm for the $t+1^{th}$ pull. Essentially, ψ_t is the policy learnt by the algorithm after t pulls. Note that $\phi(t)$ after an ϵ, δ-PAC guaranteeing exploration is essentially related to ϵ, in the sense that

$$\phi(t) < \epsilon(1-\delta) + \delta(\mu^* - \min_i \mu_i).$$

Bubeck et al. (2009) find dependencies between $\phi(t)$ and $\eta(t)$ and state that the smaller $\phi(t)$ can get, the larger $\eta(t)$ would have to be. We now analyze how FMB attains this relation.

Consider improving the policy learning algorithm with respect to $\phi(t)$. This could be achieved by keeping β constant, and increasing l. This may either reduce μ_T as seen from (3.3.2) and hence ϵ as seen from (3.3.3), or simply reduce δ as seen from (3.3.2), both ways improving the simple regret. On the other hand, regret $\eta(t)$, otherwise specifically called cumulative regret, directly depends on l as seen from (4.2.7). For a constant number of pulls, t, the summation of $G_i(w)$ in (4.2.7) runs over lesser number of terms for an increasing l. As cumulative regret is dominated by regret incurred during the initiation phase, it is seen evidently that, by increasing l we find worse bounds on regret. Thus, we evidence a Simple vs. Cumulative Regret trade-off in the proposed algorithm FMB.

5.3 Control of Complexity or Regret with β

From (3.3.2), consider the sample complexity,

$$nl = n\left(\frac{n}{\mu_T^4}\ln^2\left(\frac{n}{\delta}\right)\right)^{\frac{1}{n}} \tag{5.3.1}$$

where μ_T is given by

$$\mu_T = \min_{i:\mu_i<\mu*-\epsilon} |\mu_{T_i}|$$

with

$$\begin{aligned} \mu_{T_i} &= E[A_i - A_1] \\ &= \mu_i^{n-1}\prod_{j\neq i}[r_i^{\beta-1} + \frac{\mu_j}{r_i r_j}(I_{ij}(r_i - r_j)^\beta - r_i^\beta)] \\ &\quad -\mu_1^{n-1}\prod_{j\neq 1}[r_1^{\beta-1} + \frac{\mu_j}{r_1 r_j}(I_{1j}(r_1 - r_j)^\beta - r_i^\beta)]. \end{aligned}$$

Consider

$$\begin{aligned} E[A_i] &= \mu_i^{n-1}\prod_{j\neq i}[r_i^{\beta-1} + \frac{\mu_j}{r_i r_j}(I_{ij}(r_i - r_j)^\beta - r_i^\beta)] \\ &= \zeta_{i,1}\prod_{j\neq i}[\zeta_{i,2}\zeta_{i,3}^\beta + \zeta_{i,4}\zeta_{i,5}^\beta + \zeta_{i,6}\zeta_{i,7}^\beta] \end{aligned}$$

where $\zeta_{i,j}$ are constants given the problem at hand, essentially μ_i and r_i for all i. This can be further written as

$$E[A_i] = \zeta_{i,1}\sum_{(\theta_1,\theta_2,\theta_3):\sum_j\theta_j=n-1}\left\{\zeta_{i,2}^{\theta_1}\zeta_{i,4}^{\theta_2}\zeta_{i,6}^{\theta_3}\left(\zeta_{i,3}^{\theta_1}\zeta_{i,5}^{\theta_2}\zeta_{i,7}^{\theta_3}\right)^\beta\right\}$$

exhibiting monotonicity on β. Hence we see that μ_{T_i} can be broken down into a finite summation,

$$\mu_{T_i} = \sum_l(\kappa_{i,1,l}\kappa_{i,2,l}^\beta - \kappa_{i,3,l}\kappa_{i,4,l}^\beta)$$

where $\kappa_{i,j,l}$ are constants given the problem at hand. Thus, by tuning beta, we can control μ_{T_i} and hence μ_T. For a better sample complexity, a lower l, we need a higher μ_T. This can be achieved by increasing β since μ_T is monotonic in β. But there is a definite restriction on variation of the paramter β. Correctness of the algorithm requires $E[A_i] > E[A_j]$ for every $\mu_i > \mu_j$. Let us call the set adhering to this restriction, β_S. Since $E[A_i]$ is monotonic in β, we see β_S must be an interval on the real line.

Now consider the bounds on regret, given together by (4.2.7) and (4.2.12). Given sample complexity, or equivalently given l, regret can still be improved as it depends on $\lambda_i = \frac{\mu_{T_i}^2}{\sqrt{n}}$ that the algorithm has control on, through β. So, β can be increased to improve the bounds on regret through an increase in λ_i, keeping l constant. But with increase in λ_i or equivalently $|\mu_{T_i}|$, μ_T is expected to increase as well. In this respect, we believe β for the best regret, given an l, would be fractional, and be that value which when increased by the smallest amount will result in a decrease in sample complexity l by 1. While it has been observed experimentally that the best regret occurs for a fractional value of β, we cannot be sure whether 'The β' for the best regret was indeed achieved.

To summarize, we can control the algorithm FMB with regard to sample complexity or regret. But this tuning will inevitably have trade-offs between simple and cumulative regrets, $\phi(t)$ and $\eta(t)$ respectively, consistent with the findings of Bubeck et al. (2009). Nevertheless, there is a definite interval of restriction β_S, defined by the problem at hand, on the tuning of β.

5.4 How low can l get

Is it possible to get l as low as 1, and still get a Complexity or Regret achieving algorithm so as to reduce the cumulative regret incurred in the initiation phase? While this was accomplished experimentally, we see from a theoretical perspective of when this would be possible. We have

$$l = \left(\frac{n}{\mu_T^4} \ln^2 \left(\frac{n}{\delta} \right) \right)^{\frac{1}{n}}.$$

For $l = 1$, we would have

$$\begin{aligned} \mu_T^4 &= n \ln^2 \left(\frac{n}{\delta} \right) \\ \delta &= n e^{-\mu_T^2/\sqrt{n}}. \end{aligned}$$

For $\delta < 1$, we require

$$\begin{aligned} \exp(\mu_T^2/\sqrt{n}) &> n \\ \mu_T &> \sqrt{\sqrt{n}\ln n}. \end{aligned}$$

Thus, if some $\beta \in \beta_S$ could achieve $\mu_T > \sqrt{\sqrt{n}\ln n}$, then we would have an algorithm achieving $O(n)$ sample complexity for $l = 1$.

Chapter 6

Experiment & Results

A 10-arm bandit test bed with rewards formulated as Gaussian (with varying means and variances) or Bernoulli was developed for the experiments. Probabilistic and Greedy action selections (pFMB and FMB respectively) were performed on the quantities A_i.

6.1 Comparison with Traditional approaches

Here we describe comparisons of FMB and pFMB with traditional approaches that do not necessarily guarantee sample complexity or regret bounds. A 10-arm bandit test bed with rewards formulated as Gaussian with varying means and a variance of 1 was developed for the experiments. Figure 6.1.1 shows Average rewards and Cumulative Optimal Selections with plays averaged over 2000 tasks on the test bed. Greedy A_i and Exploring A_i are the curves corresponding to performances of FMB and pFMB

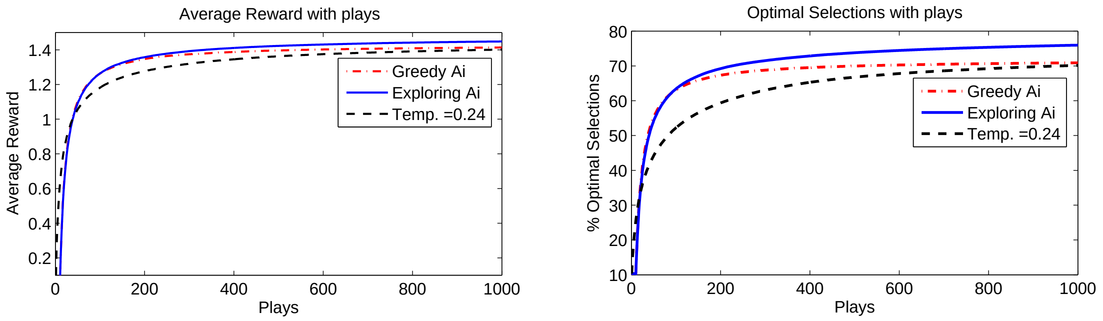


Figure 6.1.1: Two variants of the proposed algorithm, Greedy and Probabilistic action selections on A_i, are compared against the SoftMax algorithm

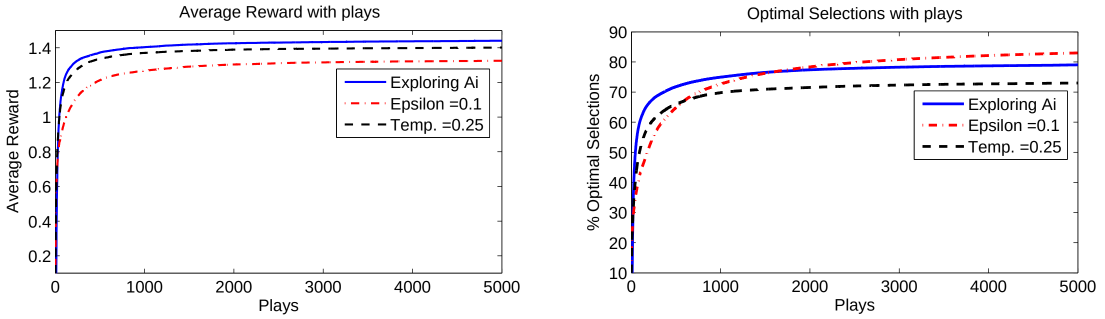


Figure 6.1.2: Asymptotic performance showing low Regret

respectively. $\beta = 0.85$ was empirically chosen without complete parameter optimization (though 4 trials of different β were made). The temperature of the SoftMax algorithm, $\tau = 0.24$ was observed to be best among the 13 different temperatures that were attempted for parameter-optimization of the SoftMax procedure. This temperature value was also seen to better ϵ-greedy action selection with $\epsilon = 0.1$. To see a more asymptotic performance, the number of plays was further increased to 5000 with Gaussian rewards incorporating varying means as well as variances and the corresponding plots are shown in Figure 6.1.2. As can be seen, though the proposed algorithms could not keep up in optimal selections, but yet are reaping higher cumulative rewards even 3000 turns after ϵ-greedy finds better optimal selections (at around 1500^{th} turn).

6.2 Comparison with State-of-the-art approaches

We now compare FMB with state-of-the-art algorithms that guarantee either sample complexity or regret. Comparisons of FMB with UCB-Revisited, henceforth called UCB-Rev, that was shown to incur low regrets [Auer & Ortner (2010)] are depicted in Figure 6.2.1. The experiments were conducted on the 10-arm bandit test bed with specific random seeds (Guassian or Bernoulli) and the results were averaged over 200 different tasks or trials. Note that UCB-Rev was provided with the knowledge of the horizon, the number of plays T, which aids in its optimal performance. Furthermore, no parameter optimization was performed for FMB and the experiments were conducted with $\beta = 0.85$. We see that FMB performs substantially better in terms of cumulative regret and its growth, even with the knowledge of horizon provided to UCB-Rev. Comparisons of FMB with Median Elimination Algorithm (MEA) [Even-Dar et al. (2006)] which was also shown to achieve $O(n)$ sample complexity is shown in Figure 6.2.2. Here we compare the performances of the two algorithms for incurring the same

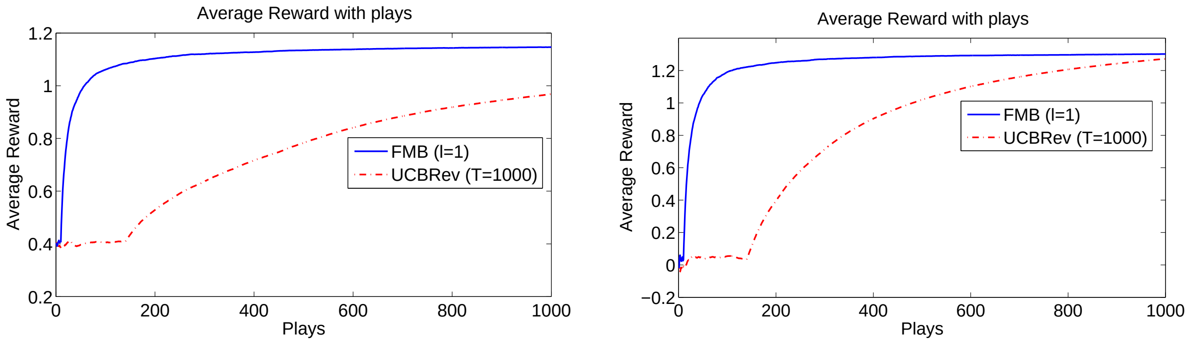


Figure 6.2.1: Comparison of FMB with UCB-Rev [Auer & Ortner (2010)] on Bernoulli and Gaussian rewards, respectively

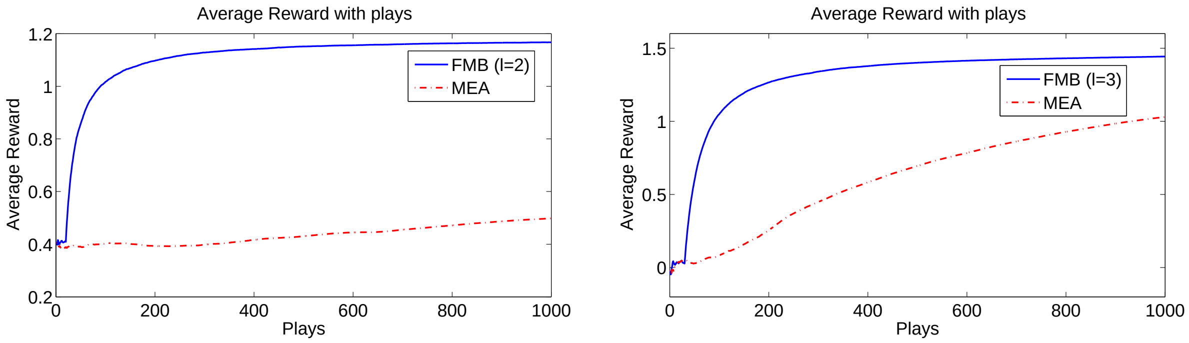


Figure 6.2.2: Comparison of FMB with Median Elimination Algorithm (MEA) [Even-Dar et al. (2006)] on Bernoulli and Gaussian rewards, respectively

sample complexity. The parameters for MEA's performances depicted ($\epsilon = 0.95, \delta = 0.95$) performed best with respect to regret among 34 different uniformly changing instances tested, while no parameter optimization for FMB was performed. To achieve $O(n)$ guarantees at $\epsilon = 0.95, \delta = 0.95$, it was observed that $l = 2$ and $l = 3$ were respectively required as arm-picks at the start of FMB for the Bernoulli and Gaussian experiments. Though this may not be a fair comparison as appropriate values of l were computed empirically to attain the ϵ, δ confidences, we observe relatively very low values of l are sufficient to ensure $O(n)$ sample complexity.

We conclude that the proposed class of algorithms, in addition to substantially reducing regrets while learning, seem to perform well with respect to sample complexity as well.

Chapter 7

Conclusions & Future Work

The proposed class of algorithms are the first to use fractional moments in bandit literature to the best of our knowledge. Specifically, the class is shown to possess algorithms that provide PAC guarantees with $O(n)$ complexity in finding an ϵ-optimal arm in addition to algorithms incurring the theoretical lowest regret of $O(\log(t))$. Experimental results support this, showing the algorithm achieves substantially lower regrets not only when compared with parameter-optimized ϵ-greedy and SoftMax methods but also with state-of-the art algorithms like MEA [Even-Dar et al. (2006)] (when compared in achieving the same sample complexity) and UCB-Rev [Auer & Ortner (2010)]. Minimizing regret has been a crucial factor in various applications. For instance Agarwal et al. (2008) describe relevance to content publishing systems that select articles to serve hundreds of millions of user visits per day. In this regard, FMB performs 20 times faster (empirically) compared to UCB-Rev. In addition to performance improvements, FMB provides a neat way of controlling sample complexity and regret with the help of a single parameter. To the best of our knowledge, this is the first work to introduce control in sample complexity and regret while learning bandits.

We note that as the reward distributions are relaxed from $R_i \in \{0, r_i\}$ to continuous probability distributions, the sample complexity in (3.3.2) is further improved. To see this, consider a gradual blurring of the bernoulli distribution to a continuous distribution. The probabilities $P\{A_i > A_1\}$ would increase due to the inclusion of new reward-possibilities in the event space. So we expect even lower sample complexities (in terms of the constants involved) with continuous reward distributions. But the trade off is really in the computations involved. The algorithm presented can be implemented incrementally, but requires that the set of rewards observed till then be stored. On the other hand the algorithm simplifies computationally to a much faster approach in case of Bernoulli distributions[1], as the mean estimates can be used directly. As the cardinality of the reward set increases, so will the complexity in computation. Since most rewards are encoded manually specific to the problem at hand, we expect low cardinal reward supports where the low sample complexity achieved would greatly help without major increase in computations.

The theoretical analysis assumes greedy action selections on the quantities A_i and hence any further exploration other than the initiation phase (for instance, the exploration as performed by the algorithm pFMB) is unrealizable. Bounds for the exploratory algorithm pFMB on the sample complexity or regret would help in better understanding of the explore-exploit situation so as to whether a greedy FMB could beat an exploratory FMB. While FMB incurs a sample complexity of $O(n)$, determination of an appropriate l given ϵ and δ is another direction to pursue. In addition, note from (3.3.2) and (3.3.3) that we sample all arms with the worst l_i, where l_i is the necessary number of pulls for arm i to ensure PAC guarantees with $O(n)$. We could reduce the sample complexity further if we formulate the initiation phase with l_i pulls of arm i, which requires further theoretical footing on the determination of appropriate values for l. The method of tuning parameter β, and the estimation of β_S are aspects to pursue for better use of the algorithm in unknown environments with no knowledge of reward support. A further variant of the algorithms proposed, allowing change of β while learning could be an interesting possibility to look at. Extensions to the multi-slot framework, applications to contextual bandits and extensions to the complete Reinforcement Learning problem would be some useful avenues to pursue.

[1] Essential requirement is rather a low cardinal Reward support

Bibliography

Hoeffding W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association.* 58(301):13–30. 4

Even-Dar E., Mannor S. and Mansour Y. (2006). Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7, 1079–1105. (document), 3, 1.5, 3.3, 6.2, 6.2.2, 7

Sutton R. and Barto A.(1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge. 1.5

Dubhashi D P. and Panconesi A. (2009). Concentration of Measure for the analysis of Randomized Algorithms, Cambridge University Press, New York. 6

Kalyanakrishnan S. and Stone P. (2010). Efficient Selection of Multiple Bandit Arms: Theory and Practice. *Proceedings of the 27th International Conference on Machine Learning*, 511-518. 1.5

Berry D A. and Fristedt B. (1985). Bandit problems. Chapman and Hall Ltd. 1.5

Auer P., Cesa-Bianchi N. and Fischer P. (2002). Finite-time Analysis of the Multiarmed Bandit Problem Machine Learning, Springer Netherlands, 47, 235-256. 1.5

Holland J. (1992). Adaptation in natural and artificial systems. Cambridge: MIT Press/Bradford Books. 1.5

Lai T. and Robbins H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6, 4–22. (document), 1.5, 4, 4.2

Agrawal R. (1995). Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27, 1054–1078 1.5

Min S. and Chrysostomos L N. (1993). Signal Processing with Fractional Lower Order Moments: Stable Processes & Their Applications. *Proceedings of IEEE*, Vol.81 No.7, 986-1010. 1.5

Achim A M., Canagarajah C N. and Bull D R (2005). Complex wavelet domain image fusion based on fractional lower order moments, *8th International Conference on Information Fusion*, Vol.1 No.7, 25-28. 1.5

Xinyu M. and Nikias C L. (1996). Joint estimation of time delay and frequency delay in impulsive noise using fractional lower order statistics, *IEEE Transactions on Signal Processing*, Vol.44 No.11, 2669-2687. 1.5

Liu T. and Mendel J M. (2001). A subspace-based direction finding algorithm using fractional lower order statistics, *IEEE Transactions on Signal Processing*, Vol.49 No.8, 1605-1613. 1.5

Agarwal D., Chen B., Elango P., Motgi N., Park S., Ramakrishnan R., Roy S. and J. Zachariah. (2009). Online models for content optimization. In Advances in Neural Information Processing Systems 21, 2009. 7

Bubeck S., Munos R., and Stoltz G. (2009). Pure Exploration in Multi-Armed Bandit Problems. In 20th Intl. Conf. on Algorithmic Learning Theory (ALT), 2009. 5.2, 5.3

Auer P. and Ortner R. (2010). UCB revisited: Improved regret bounds for the stochastic multiarmed bandit problem. Periodica Mathematica Hungarica, 61(1-2):55–65, 2010. (document), 1.5, 6.2, 6.2.1, 7

Kim S. and Nelson B. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3):251–273, 2001. 1.5

Schmidt, Christian, Branke, Jürgen, and Chick, Stephen E. Integrating techniques from statistical ranking into evolutionary algorithms. In *Applications of Evolutionary Comp*utations, volume 3907 of LNCS, pp. 752–763. Springer, 2006. 1.5

Dayan, P. and Abbott, L.F. (2001) Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems (MIT Press, Cambridge MA), 2001. 1

Narayan, A. and Ravindran, B. (2011). "Fractional Moments on Bandit Problems". In the Proceedings of the Twenty Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2011), pp. 531-538. AUAI Press.

7