2021 CLASS FOR DISSERTATIONS SUBMITTED TO IITM

A Project Report

submitted by

ARABHI SUBHASH, CS17B005

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

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

June 2021

THESIS CERTIFICATE

This is to certify that the thesis entitled **Encoder Selection Problem**, submitted by **Arabhi Subhash, CS17B005**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor 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.

Arun Rajkumar
Research Guide
Professor
Dept. of CSE
IIT Madras, 600036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to thank **Prof. Lakshmi Narasimhan**, Dept. of Electrical Engineering, IIT Palakkad for guiding me throughout the project. I would also like to thank **Utsav Dey** for his help.

ABSTRACT

KEYWORDS: Encoder, Channel, Rate, Tolerance, Bandits, System, Linear Programming, Markov

Every data transfer system (like storage devices, networks etc) have some kind of encoding for data reliability. There is a trade-off between memory efficiency and reliability in encoders. In this project, I am trying to solve the problem of encoder selection based on history and get the best of the trade-off using different learning algorithms.

Brief : Using concepts from **Linear Bandits** to choose among different coding algorithms to increase efficiency in data transfer systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
RTFM	Read the Fine Manual
OPLB	Optimistic Pessimistic Linear Bandits
SLB	Stochastic Linear Bandits

NOTATION

en	Number of Encoders
ch	Number of Channels
P	Channel Selection probability matrix
E	Error Probability matrix
R	Rate vector
tol	Tolerance on Probability of error
lr	Learning Rate
p	Estimate of P
Reg	Regret
$\lambda, \delta, \gamma, \alpha, w$	Different parameters used along the course

CHAPTER 1

INTRODUCTION

The problem we are trying to address is encoder selection. Consider a data transfer or storage system, where our data is encoded and transferred through a set of different channels. We have different encoders and it is usually the case that more reliable encoder (i.e. less error probability) have less data-transfer rate. Our problem here is to select encoders so as to get the best rate within some error tolerance. We are solving for a general case where the channels are selected with a hidden probabilities but we are aware of error probabilities of each encoder-channel pair and rate of transfer associated with each encoder. Idea is to come up with a efficient (computationally less expensive) online algorithm which solves our use case so that it can be deployed into real time systems.

In this thesis we are trying to formulate our problem as stochastic linear bandit problem. The algorithm we are trying to use is a modified version of Optimistic-Pessimistic Linear Bandits as presented in Pacchiano *et al.* [2020]. We are able to see the convergence of average rates keeping the error below tolerance with the proposed algorithm. We are also addressing the regret to some extent. Apart from that we are proposing two other algorithms which produce better rate with 5% lenience in tolerance and address the problem with Markov channel probabilities.

1.1 Problem Formulation

We adopt the following notion. Let system has en encoders to choose from and ch channels of transfer. Let R be a vector of size $en \times 1$ having rates of each encoder and E be a matrix of size $en \times ch$ where E_{ij} is error probability corresponding to ith encoder and jth channel. P is the background probability with which we are selecting channels, for a general case it's of size $ch \times 1$ and tol is the error tolerance. T is the total number of rounds the algorithm runs. $||x||_2$, $||x||_1$ denotes l2 and l1 norms of vector x.

The setting is, in round t we provide the system with x_t of size $en \times 1$ which is a probability distribution from which system samples an encoder e, in the background, system samples a channel c, using distribution P. Now Bernoulli sampling is done with probability E_{ec} and we get 1 or 0 (whether the transfer is successful or not) as output b_t. Using this output we need to make better estimate x (i.e. x_{t+1}). Here reward is $r_t = R \cdot x_t$ and cost is $c_t = 1 - b_t$

1.2 Algorithm

To apply the below algorithm our system must have a safe probability distribution x_0. It acts as a base to which we project our vectors and try to estimate cost parameters. Pertaining to our problem, x_0 is vector for which irrespective of channel probabilities the error is below tolerance.

$$x_0 \cdot (E \cdot P) <= tol$$

$$MAX(x_0 \cdot E) <= tol$$

We can obtain x_0 it by solving system of linear inequalities -

$$E^T[c] \cdot x <= tol \ \forall c \in \{0, 1...ch - 1\}$$

We have seen algorithm to be working even if these equations have no proper solution, if one can provide a x_0 with trail and error that has a average cost over large number of trails that is less than tolerance. Let c_0 be average cost of x_0 and $eo = x_0/\,||x_0||_2$. We define projection into safe space as $x^{o,\perp} = x - x^o$ where $x^o = (x \cdot e0)e0$. The variables R, S, L defined in page 3 of Pacchiano *et al.* [2020] satisfy the assumptions when all equal to 1 in our problem. The x, probability distribution is in general terms equal to π, so along the text you might see using these variables interchangeably to make better sense of equations.

Algorithm 1 Encoder selection using modified OPLB algorithm

Require: confidence δ, regularization λ, constant α
  for $t = 1, 2...T$ **do**
    Run a system cycle with x_t and generate c_t
    Using the values $c_0, c_1...c_t$ construct $\widehat{\mu}_t^{o,\perp}$, an estimate of P in safe space
    Generate an estimated cost function $\widetilde{c_x}$ using $\widehat{\mu}_t^{o,\perp}$
    Solve linear program that maximizes $R \cdot x$ with $\widetilde{c_{x,t}} < tol$ to get x_{t+1}
  end for

Calculation of $\widehat{\mu}_t^{o,\perp}$:

$$I_{V_{o,\perp}} = I - (1/\,||x_0||_2^2)x_0x_0^T$$

$$\Sigma_t^{o,\perp} = \lambda I_{V_{o,\perp}} + \sum_{s=1}^{t} x_s^{o,\perp}(x_s^{o,\perp})^T$$

$$c_t^{o,\perp} = c_t - (x_t \cdot e_0)c_0/\,||x_0||_2$$

$$\widehat{\mu}_t^{o,\perp} = (\Sigma_t^{o,\perp})^{-1} \sum_{s=1}^{t} c_s^{o,\perp} x_s^{o,\perp}$$

Calculation of cost function $\widetilde{c_{x,t}}$:

$$\beta_t = R\sqrt{en\log(\frac{1+(t-1)L^2/\lambda}{\delta})} + \sqrt{\lambda}S$$

$$\widetilde{c_{x,t}} = \frac{(x^o \cdot e_0)c_0}{||x_0||} + (x^{o,\perp} \cdot \widehat{\mu}_t^{o,\perp}) + \alpha\beta_t((O \cdot x^{o,\perp}) - \gamma)/\sqrt{en}$$

In the equations, I represent identity matrix of size $en \times en$ and O is $en \times 1$ vector of all ones. We found algorithm to be working it's best for $\gamma = \frac{1}{2*en}$. The calculation of β_t and $\widehat{\mu}_t^{o,\perp}$ is taken from Pacchiano *et al.* [2020]. The equation of $\widehat{\mu}_t^{o,\perp}$ is approximate version of l_2 regularized least squares estimate of P in safe sub space. These formulations allows us to use the bounds proved in Abbasi-yadkori *et al.* [2010].

1.3 Regret Analysis

In any general linear bandit setup we have reward as $(x \cdot \theta)$, where improvement in estimation of theta is reflected as improvement in regret. In our case we know that $\theta = R$ and better estimation of P is in-turn seen as improvement in x. So we were not able to prove regret conclusively but able to address it to some extent using below calculations. This also makes us drop the t subscript in optimal x, x_{t*} and just call it x_*. We know regret for our problem formulation is $Reg(T) = \sum_{t=1}^{T} reg_t$ where $reg_t = (R \cdot x_*) - (R \cdot x_t)$. We can break the problem into 2 cases where $\widetilde{c_{x_*,t}} \leq tol$ i.e. x_* is within x_t space and otherwise.

Case 1 : $\widetilde{c_{x_*,t}} \leq tol$

As we are solving linear program with maximum $R \cdot x_t$ and $\widetilde{c_{x_t,t}} < tol$. We can see that if $\widetilde{c_{x_*,t}} \leq tol$ the we can be sure that

$$R \cdot x_t \geq R \cdot x_*$$

$$(R \cdot x_*) - (R \cdot x_t) \leq 0$$

$$reg_t \leq 0$$

Case 2 : $\widetilde{c_{x_*,t}} > tol$

In this case we like to use the formulation done in page 13 lemma 4 of Pacchiano *et al.* [2020] and do a similar one for our problem. We wish to find

$$\widetilde{x_t} = \eta_t x_* + (1 - \eta_t)x_0 \tag{1.1}$$

which is with in safe space. We want η_t to express the extent to which x_* belongs to the safe space. This implies η_t is maximum or $\widetilde{c_{\widetilde{x_t},t}} = tol$. Substituting $\widetilde{x_t}$ in $\widetilde{c_{x,t}}$ and using $\gamma' = \gamma/\sqrt{en}$ we get

$$\widetilde{c_{\widetilde{x_t},t}} = tol = \frac{(1 - \eta_t)(x_0 \cdot e_0)c_0}{||x_0||_2} + \eta_t \widetilde{c_{x_*,t}} - (1 - \eta_t)\alpha\gamma'$$

Solving for η_t gives

$$\eta_t = \frac{tol - c_0 + \alpha\beta_t\gamma'}{\widetilde{c_{x_*,t}} - c_0 + \alpha\beta_t\gamma'} \tag{1.2}$$

Now we have to obtain bound $\widetilde{c_{x_*,t}}$. We know that $(x_* \cdot \mu_*) \leq tol$ where $\mu_* = P$, denoted like that to better understand equations. This gives

$$(((x_* \cdot e_0)\frac{x_0}{||x_0||_2}) \cdot \mu_*) + (x_*^{o,\perp} \cdot \mu_*) \leq tol$$

$$\frac{(x_*^o \cdot e_0)c_0}{\|x_0\|_2} + (x_*^{o,\perp} \cdot \mu_*) \leq tol \tag{1.3}$$

We know that

$$\widetilde{c_{x_*,t}} = \frac{(x_*^o \cdot e_0)c_0}{\|x_0\|_2} + (x_*^{o,\perp} \cdot \widehat{\mu_t}^{o,\perp}) + \alpha\beta_t((O \cdot x_*^{o,\perp}) - \gamma)/\sqrt{en}$$

Using (1.3)

$$\widetilde{c_{x_*,t}} \leq tol - (x_*^{o,\perp} \cdot \mu_*) + (x_*^{o,\perp} \cdot \widehat{\mu_t}^{o,\perp})\alpha\beta_t((O \cdot x_*^{o,\perp}) - \gamma)/\sqrt{en}$$

$$\widetilde{c_{x_*,t}} \leq tol - \left\|\widehat{\mu_t}^{o,\perp} - \mu_*\right\|_2 \left\|x_*^{o,\perp}\right\|_2 + \alpha\beta_t((O \cdot x_*^{o,\perp}) - \gamma)/\sqrt{en}$$

As we have used the same formulation of $\widehat{\mu_t}^{o,\perp}$ proposed in Abbasi-yadkori *et al.* [2010], we can use the confidence set proved there - $\left\|\widehat{\mu_t}^{o,\perp} - \mu_*\right\|_2 \leq \beta_t$ with a probability $1 - \delta$ (confidence δ)

$$\widetilde{c_{x_*,t}} \leq tol + \beta_t \left\|x_*^{o,\perp}\right\| + \alpha\beta_t(O \cdot x_*^{o,\perp})/\sqrt{en} - \alpha\beta_t\gamma/\sqrt{en}$$

We know that $(O \cdot x) \leq \|x\|_1 \leq \sqrt{d}\,\|x\|_2$

$$\widetilde{c_{x_*,t}} \leq tol + (1 + \alpha)\beta_t \left\|x_*^{o,\perp}\right\|_2 - \alpha\beta_t\gamma' \tag{1.4}$$

Since $\widetilde{x_t}$ is with in the safe space, the linear programming step in algorithm assures that $R \cdot x_t \geq R \cdot \widetilde{x_t}$

$$reg_t = (R \cdot x_*) - (R \cdot x_t) \leq (R \cdot x_*) - R \cdot \widetilde{x_t}$$

Using (1.1)

$$reg_t = (R \cdot (x_* - x_0))(1 - \eta_t)$$

Using (1.2)

$$reg_t = (R \cdot (x_* - x_0))(1 - \frac{tol - c_0 + \alpha\beta_t\gamma'}{\widetilde{c_{x_*,t}} - c_0 + \alpha\beta_t\gamma'})$$

Using (1.4)

$$reg_t \leq (R \cdot (x_* - x_0))(1 - \frac{tol - c_0 + \alpha\beta_t\gamma'}{tol + (1+\alpha)\beta_t \left\| x_*^{o,\perp} \right\|_2 - c_0})$$

$$reg_t \leq (R \cdot (x_* - x_0))((1+\alpha) \left\| x_*^{o,\perp} \right\|_2 - \alpha\gamma') \frac{\beta_t}{(tol - c_0) + (1+\alpha)\beta_t \left\| x_*^{o,\perp} \right\|_2} \tag{1.5}$$

From the result (1.5) we can see that regret is less than 1 fraction with β_t but we know $\beta_t \sim c_1\sqrt{\log(t)} + c_2$ - this $\sqrt{\log(t)}$ dependence makes its value almost constant for $10^5 to 10^7$ iterations which leaves us with linear regret. We know that linear regret is weak, repeated sub-optimal action pick can do that but we should note that in our problem this is a case and the other one have a negative value. Below are some more algorithms with better results whose regret is out of scope because these involve manipulations at implementation level to get better output.

1.4 Some More Algorithms

1.4.1 Approximate Algorithm with Adaptive Tolerance

Algorithm 2

Require: confidence δ, regularization λ, constant $\alpha \sim 0.001$
  for $t = 1, 2...T$ **do**
    Run a system cycle with x_t and generate c_t
    Using the values $c_0, c_1...c_t$ construct $\widehat{\mu}_t^{o,\perp}$, an estimate of P in safe space
    Calculate new estimated cost function $\widetilde{c_x}^*$ using $\widehat{\mu}_t^{o,\perp}$
    Adaptive tolerance $tol_a = 2 * tol - (\sum_{s=1}^{t} c_s)/t$
    Solve linear program that maximizes $R \cdot x$ with $\widetilde{c_{x,t}}^* < tol_a$ to get x_{t+1}
  end for

The main changes from the previous algorithm is the adaptive tolerance, new cost function $\widetilde{c_{x,t}}^*$. We also observed the best α value for this function 0.001, which makes sense because we are performing no estimates on reward side and Pacchiano *et al.* [2020] puts $\alpha \geq 1$ to constrain regret caused by this estimate.

$$\widetilde{c_{x,t}}^* = \frac{(x \cdot e_0)c_0}{||x_0||} + (x^{o,\perp} \cdot \widehat{\mu}_t^{o,\perp}) + \alpha\beta_t(\left\|x_{t-1}^{o,\perp}\right\|_2)$$

1.4.2 Problem with Markov Channel Probabilities

Algorithm 3

Require: Weight Parameter w
  for $t = 1, 2...T$ **do**
    for $s = 1, 2...S$ **do**
      Run the system with x_t and get c_{t_s}
    end for
    Train HMM with as $p_t^{st} = p_{t-1}^{ss}, p_t^{em} = [1 - x_t, x_t]$ and $c_{t_1}...c_{t_S}$ as output
    Obtain *temp*, transition probabilities from training
    Calculate p_t^{tr} from *temp* and p_{t-1}^{tr} using w
    Calculate p_t^{ss} from p_t^{tr}
    Calculate new cost function $c'_{x,t} = (x \cdot (E \times (p_t^{ss})^T))$
    Solve linear program that maximizes $R \cdot x$ with $c'_{x,t} < tol$ to get x_{t+1}
  end for

In this algorithm we try to estimate transition probabilities of the hidden Markov chain using error probabilities and system output. Let $p_t^{ss}, p_t^{tr}, p_{st}^{st}, p_t^{em}$ be the steady state, transition, start and emission probability estimates of P at time t. We formulate cost function using steady-state probabilities of the HMM model we trained. We use same encoder distribution S times, to get better estimates. We have to make sure S is greater than size of P i.e. $S \geq ch \times ch$. Selecting encoder from best x in every round in turn affect p estimates. This can be avoided by using weighted mean with parameter w in finding p_t^{tr} with the equlation - $p_t^{tr} = \frac{w*(t-1)*tr_{t-1}+p_t^{tr}}{w*t-w+1}$. Algorithm seen to be working best when w is 2 or 3. To still improve performace and get better transition probability estimates, we can use ϵ-greedy pick of x and adaptive tolerance etc.

1.5 Results

In the obtained plots we are using $\delta = .1$, $\lambda = .1$, and $\alpha = .1, .001$ for algorithms 1 and 2 respectively. This setup is reasonable and works for most real-time systems. We are able to see rate convergence in all the three algorithms. The red line indicate tolerance and optimal rate for error and rate plots respectively

As discussed in previous section algorithm 1 tend to select safe action, we can see from 1.1, after initial exploration it's error always lie below tolerance. Algorithm 2 on the other hand use adaptive tolerance to get better rates, the trade-off is lenient error tolerance and rate fluctuations. For Markov setup we compared algorithm 3 and 2. The faster convergence and slightly better rates in 1.2 depict proper use of markov knowledege.

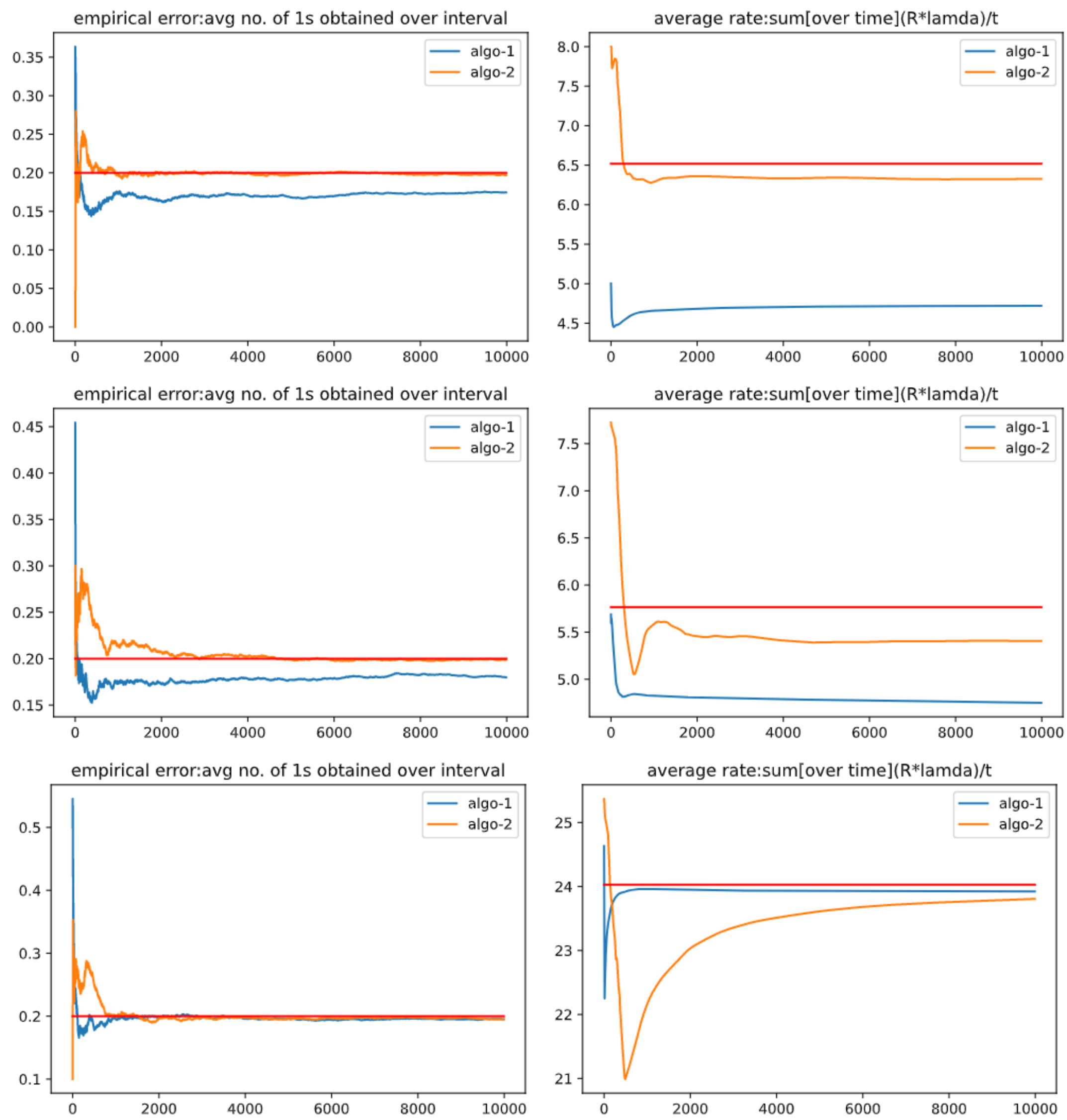

Figure 1.1: Plots of empirical error(left) and average reward(right) of sample test-beds with (en,ch, tol) values equal to (6,8,.2), (10,12,.25), (15,20,.2) respectively when algorithms 1 and 2 are used

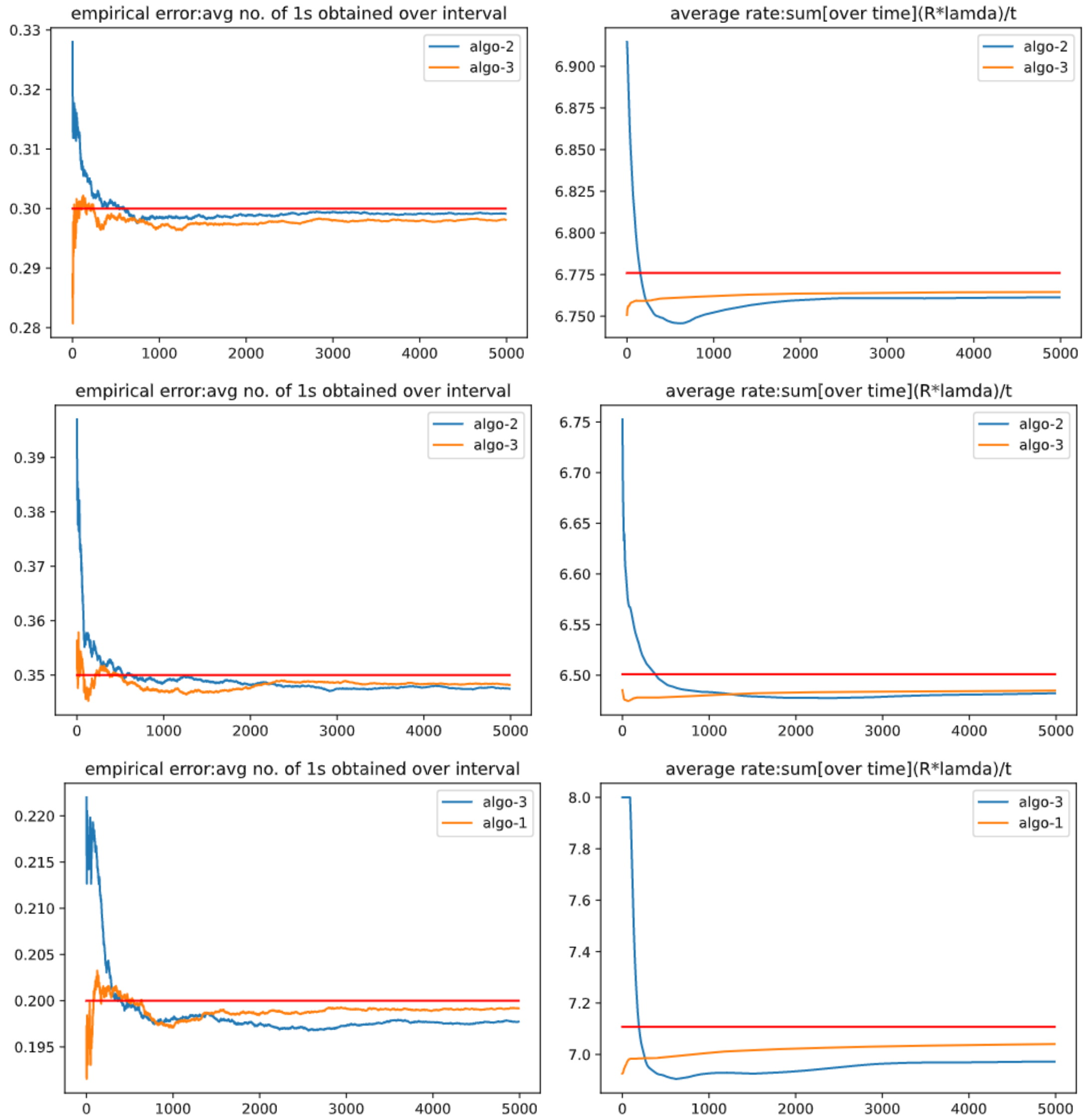

Figure 1.2: Plots of empirical error(left) and average reward(right) of sample test-beds with Markov channel probabilities and (en,ch,tol) values equal to (3,3,.3), (4,5,.35), (6,8,.2) respectively when algorithm 3 is used

1.6 Conclusion and Future Work

We see that algorithms 1 and 2 can be deployed based on the use case like whether we want a safe/reliable system or the one with better rates. These algorithms were producing decent results but theoretically we weren't able to prove their convergence. We showed that there is bound on regret at each round in Regret Analysis section but that doesn't address the convergence seen. We can see from 1.1 that within reasonable number of rounds the gap between r_t and r_* is decresing. We might include case:1 probability to get better bound. Let this probability be $1 - \epsilon$, then the regret bound we obtained will be of the form ϵ times regret in the case:2. If we were able to show that this ϵ a decreasing function of time then that can be used to bound regret. This is the direction we see the future work going.

APPENDIX A

I

LIST OF PAPERS BASED ON THESIS

1. Aldo Pacchiano and Mohammad Ghavamzadeh and Peter Bartlett and Heinrich Jiang, Stochastic Bandits with Linear Constraints, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR 130:2827-2835*, (2020).

2. Yasin Abbasi-yadkori and Dávid Pál and Csaba Szepesvári, Improved Algorithms for Linear Stochastic Bandits, *Advances in Neural Information Processing Systems 24 (NIPS)*, (2011).

REFERENCES

Abbasi-yadkori, Y., **D. Pál**, and **C. Szepesvári** (2010). Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems 24 (NIPS 2011)*. URL `https://papers.nips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html`.

Pacchiano, A., **M. Ghavamzadeh**, **P. Bartlett**, and **H. Jiang** (2020). Stochastic bandits with linear constraints. *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR 130:2827-2835, 2021*. URL `https://arxiv.org/abs/2006.10185`.