ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Bernoulli Bandits
An Empirical Comparison

Ronoh K.N[1,2], Oyamo R.[1,2], Milgo E.[1,2], Drugan M.[1] and Manderick B.[1]

1- Vrije Universiteit Brussel - Computer Sciences Department - AI Lab
Pleinlaan 2 - B-1050 Brussels - Belgium
2- Moi University
P.o - Box - 3900- 30100 - Eldoret - Kenya

Abstract. An empirical comparative study is made of a sample of action selection policies on a test suite of the Bernoulli multi-armed bandit with $K = 10$, $K = 20$ and $K = 50$ arms, each for which we consider several success probabilities. For such problems the rewards are either *Success* or *Failure* with unknown success rate. Our study focusses on ϵ-greedy, *UCB1-Tuned*, Thompson sampling, the Gittin's index policy, the knowledge gradient and a new hybrid algorithm. The last two are not well-known in computer science. In this paper, we examine policy dependence on the horizon and report results which suggest that a new hybridized procedure based on Thompsons sampling improves on its regret.

1 Introduction

In this paper, we compare empirically a number of action selection policies on a special case of the stochastic multi-armed bandit (MAB) problem: the Bernoulli bandit. The bandit does not know the expectations of the reward distributions. In order to estimate them, the rewards have to be collected from all arms. Each time a new reward is obtained, the estimate of the corresponding distribution is updated and the agent becomes more confident in the new estimate. Meanwhile, the agent has to try to achieve its goal: maximising the total expected reward. The MAB, introduced by [1], is the simplest example of a sequential decision problem where the agent has to find a proper balance between exploitation and exploration. The importance of the MAB lies in the *exploitation-exploration tradeoff* inherent in sequential decision making. *Exploitation* means that the greedy arm is selected, i.e. the one with the highest observed average reward. Since this is not necessarily the optimal arm, the agent may resolve to *exploration* of a non-greedy arm to improve the estimate of its expected reward. An action selection policy tells the agent which arm to pull next. Regret is the expected loss after n time steps due to the fact that it is not always the optimal arm that is played. Maximizing the total expected reward is equivalent to minimizing the total expected regret. The expected regret incurred after n time steps is defined as $R(n) \triangleq n\mu^* - \sum_{i=1}^{n} \mu_n$, where μ^* is the largest true mean and μ_n is the true mean of the arm pulled at time step n. This can be rewritten as $R(n) = n\mu^* - \sum_{k=1}^{K} \mu_k \ \mathbb{E}(n_k)$, where $\mathbb{E}(n_k)$ is the expected number of times that arm k is played during the first n time steps. MAB-problems can be classified according to reward distributions of the arms, the time horizon which is

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

finite or infinite, and statistical analysis are usually either done according to the frequentist or the Bayesian paradigm. Theoretical studies prefer to minimize the *total expected regret* or *loss*. Since theory only gives worst case upper bounds for the this regret. The empirical performance of a policy is in most cases better than indicated by these theoretical regret bounds. Moreover, for many action selection policies there is no theoretical analysis. The rest of this paper is organised as follows: In Section 2, we give a brief review of previous empirical research. Section 3 gives the basics of the Bernoulli bandit problem. Section 4, reviews the action selection policies considered in the empirical comparison. Section 5 describes the experimental setup and the results. Finally, the conclusion is given in Section 6.

2 State of the Art of Empirical Comparison

A systematic evaluation by [2] compared popular action-selection policies used in reinforcement learning but did not include the Gittins index, the knowledge gradient and Thompson sampling. It investigated the effect of the variance of rewards on the policy's performance and optimally tuned the parameters of each one. Another paper [3] gave a preliminary study of ϵ-greedy with *softmax* and interval estimation. Unfortunately, it did not detail performance measures used, hence the difficulty in interpretation of results. This study gives an insight into the ϵ-greedy (ϵG), used successfully in reinforcement learning, the *UCB1-Tuned*, a variant of the upper confidence bound (UCB) policies that works very well in practice, the Gittins index (GI) and the knowledge gradient (KG) relative to Thompson sampling (TS), a Bayesian approach to the exploitation-exploration trade-off that recently became popular in machine learning. TS has been shown to provide the best alternative for MABs with side observations and delayed feedback. The policy is broadly applicable and easy to implement [4]. GI and KG are quite popular in operations research but are they relatively unknown in the machine learning community. The motivation for our empirical comparison is threefold: few empirical comparisons have been done so far, theoretical comparison is still limited in its scope in some cases, and an empirical understanding of the Bernoulli bandit problem is important for many applications, e.g. optimizing the click through rate [5].

3 Bernoulli Bandit

In the Bernoulli bandit problem, the agent chooses among K different arms: $k = 1, \cdots, K$. When arm k is pulled either the reward 1 for *Success* is received with probability μ_k or 0 for *Failure* with probability $1 - \mu_k$. The rewards thus have a Bernoulli distribution $Ber(\mu_k)$ with *unknown* success probability μ_k. The estimated mean after n_k trials, of which s_k are successes and f_k are failures, is given by $\hat{\mu}_k = \frac{s_k}{s_k+f_k}$. The *optimal* arm $k^* = \arg\max_{1\leq k\leq K} \mu_k$ is the one with the highest *true but unknown* mean $\mu^* = \max_{1\leq k\leq K} \mu_k$. It is always assumed that the arms are sorted according to their expected rewards: $\mu_1 > \cdots > \mu_k >$

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

$\cdots > \mu_K$, i.e. the first arm is the optimal one and the last is the worst one. For each non-optimal arm, i.e. $k \neq 1$, the optimality gap is defined as $\Delta_k \triangleq \mu^* - \mu_k$ and the smallest gap $\Delta \triangleq \min_{k \neq 1} \Delta_k$ is assumed to be positive so that not more than one arm is optimal. The *greedy* arm at each time step is the arm with the highest estimated mean at that moment: $\hat{k}^* = \arg\max_{1 \leq k \leq K} \hat{\mu}_k$. The greedy arm might be different from the optimal one, especially in the beginning when too few rewards are available to have a reliable estimate of the true means.

4 Brief Description of the Action - Selection Policies

The **ϵ-Greedy** (ϵG) selects the greedy arm $\hat{k}^*$ most of the time with probability $1 - \epsilon$ (exploitation), while with a small probability ϵ it selects uniformly at random one of the K arms regardless of their estimated mean (exploration) [6]. ϵG works well in practice and is considered to be a benchmark.

Thompson Sampling (TS) relies on the presence and analysis of posterior data [7]. It maintains a prior distribution for the unknown parameters which at each time step n, when an arm has been played and a reward obtained, is updated using Bayes' rule to obtain the posterior. The arm with the probability of being the most optimal according to the current posterior distributions is then pulled. Reward samples are taken from the distribution and the best arm played according to the drawn parameters. TS can be summarized as follows [5]:

Algorithm 1 Thompson Sampling

Input: Initial number of successes $s_k = 0$, the failures $f_k = 0$, and their sum $n_k = 0$

for timestep $n = 1, \cdots, N$ **do**

1. For each $k = 1, \cdots, K$, sample r_k from the corresponding distribution $Beta(s_k, f_k)$
2. Play arm $k^* = \arg\max_k r_k$ and receive reward r
3. If $r = 1$, increment s_{k^*} else increment f_{k^*}

The **UCB1-Tuned** belongs to the class of Upper Confidence Bound (UCB) policies that compute an index to decide deterministically which arm to pull. UCB-policies are examples of *optimism in the face of uncertainty* [8]. The agent makes optimistic guesses about the expected rewards of the arms and selects the arm with the highest guess. *UCB1-Tuned* is a variant that has a finite-time regret logarithmically bound for arbitrary sets of reward distributions with bounded support. It takes the estimated variance V_k when arm k is pulled n_k into account and can be summarized as follows [6]:

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Algorithm 2 *UCB1-Tuned*

Input: Initial $r_k = 0$
for timestep $n = 1, \cdots, N$ **do**

1. Play machine k that maximizes $\bar{r}_k + \sqrt{\frac{\ln n}{n_k} \min(1/4, V_k(n_k))}$, where $\bar{r}_k$ is the estimated mean reward of arm k, and update $\bar{r}_k$ with the obtained reward r_k

We have included TSH which is a hybrid that starts as *UCB1-Tuned* which has better initial performance and continues as TS which outperforms *UCB1-Tuned* after some time. The switching time is determined empirically and increases as the number of arms increases.

The **Gittins index** ν_G of an arm depends on the number of times n_k it has been selected. GI relates the problem of finding the optimal policy to a stopping time problem [9]. It determines for each arm an index ν_G and selects at each time the arm with the highest value.

Algorithm 3 Finite Horizon Gittins for Bernoulli Bandits

Input: Initial successes $s_k = 0$, failures $f_k = 0$, and their sum $n_k = 0$.
for each time step $n = 1, \cdots, N$, **do**

1. Play each arm once and calculate its FHG-index $\nu_G(s_k, f_k)$
2. Play arm $k^* = \arg\max_k \nu_G(s_k, f_k)$ and observe corresponding reward r. In case of ties, choose one arm randomly among them.
3. If $r = 1$, increment s_{k^*} else increment f_{k^*} and recalculate index $\nu_G(s_{k^*}, f_{k^*})$ of arm k^*.

The **Knowledge Gradient** (KG) can be adapted to handle cases where the rewards of the arms are correlated [10]. The policy selects an arm according to: $k_{KG} = \arg\max_{1 \leq k \leq K} \hat{\mu}_k + (N - n)\nu_{KG}(k)$ where $\nu_{KG}(k)$ is the knowledge gradient index of arm k. KG adopts the procedure that follows:

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Algorithm 4 Finite Horizon Knowledge Gradient for Bernoulli Bandits

Input: Initial $s_k = 0, f_k = 0$ and $n_k = s_k + f_k$
for $k = 0, 1, \cdots, K$ and $n = 1, \cdots, N$ **do**

1. Calculate the KG-index $\nu^{(k)}_{KG}(s_k, f_k)$
2. Play arm $k^* = \arg\max_k \nu_{KG}(s_k, f_k)$ and observe corresponding reward r. In case of ties, choose one arm randomly among them.
3. If $r = 1$, increment s_{k^*} else increment f_{k^*} and recalculate index $\nu_{KG}(s_{k^*}, f_{k^*})$ of arm k^*.

5 Empirical Analysis

The number of arms in our test suite are $K = 10$, $K = 20$ and $K = 50$. The horizon ranges from $N = 500$ to $N = 10,000$. We compare the cumulative regret of the policies discussed in Section 4 as the horizon N and the number of arms K vary. TSH performs relatively well for $K = 10$ arms with $N > 100$. With $K = 20$, TSH performs relatively better than the other strategies when $N > 180$. GI and ϵG greedy surprisingly have notably lesser regret for $N \leq 500$. TSH and TS perform best for $N > 500$. When N is fixed to 5000 and K is varied, it is indeed worth noting that TSH improves in relative performance. TS, consistent with the results in [11], shows the best overall performance.

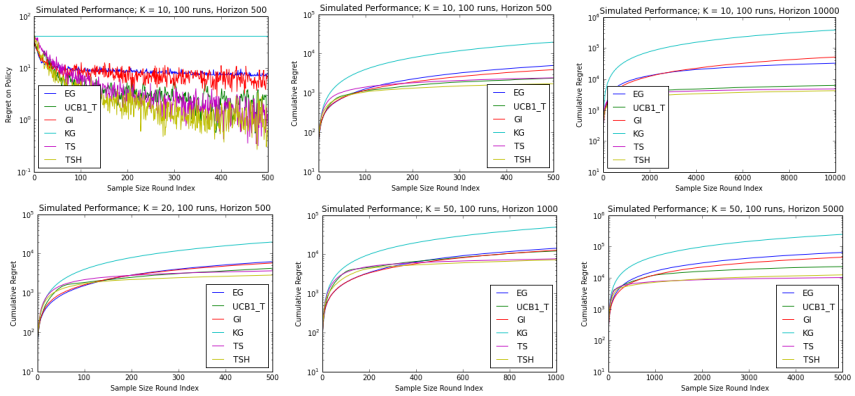

Figure 1: Figure showing the horizon N for $K = 10, K = 20$ and $K = 50$. For more information, see the text.

6 Conclusions

The results provide a clue to an important question: At what stage does one begin to use TS? To achieve minimal cumulative regret, the agent can exploit

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

the theoretical guarantees of the *UCB1-Tuned* before using TS. Further empirical studies should ascertain the time needed to initialize deterministically. The possibility of a policy incorporating the features of UCB and TS to minimize cumulative regret needs to be investigated. The analysis was done for an environment for which the success rate was fixed - empirical tests need to be done for scenarios where the rates change according to some stochastic process.

References

[1] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[2] Volodymyr Kuleshov and Doina Precup. Algorithms for the multi armed bandit problem. *Journal of Machine Learning*, 1, PP. 1 - 48, 2000.

[3] Joannes Vermorel and Mehryar Mohri. Multi armed bandit algorithms and empirical evaluation. *In European Conference on Machine Learning, Springer*, PP. 437-448, 2005.

[4] Shie Manor Aditya Gopalan and Yishay Mansour. Thompsons sampling for complex online problems. *Proceedings of the 31st International Conference on Machine Learning, Beijing, China*, W and CP Vol. 32, 2014.

[5] Olivier Chapelle and Lihong Li. An empirical evaluation of thompsons sampling. *Yahoo! Research,Santa Clara Canada, CA*, NIPS 2011: 2249-2257, 2011.

[6] C. Nicolo P. Auer and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning, Kluwer Academic Publishers.*, Vol.. 47: 235-256, 2002.

[7] Thompsons W.R. On the likelihood that one unknown probability exceeds another in view of evidence of two samples. *Biometrika*, 01/1933, 25: 285-294, 1933.

[8] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[9] K. Glazebrook J. C. Gittins and R. Weber. *Multi-Armed Bandit Allocation Indices*. J. Wiley And Sons, Series Wiley-Interscience Series In Systems And Optimization, New York, 2011.

[10] W. B. Powell and I. O. Ryzhov. *Optimal Learning*. John Wiley and Sons, Canada, 2012.

[11] Shipra Agrawal and Navil Goyan. Analysis of thompsons sampling for the multi-armed bandit problem. *JMLR: Workshop and Conference Proceedings*, 23:39.1- 39.26, 2012.