Thompson Sampling for Dynamic Multi-armed Bandits

Article · December 2011

DOI: 10.1109/ICMLA.2011.144

CITATIONS
12

READS
1,566

3 authors:

Neha Gupta
Munich University of Applied Sciences
7 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Ole-Christoffer Granmo
Universitetet i Agder
169 PUBLICATIONS **1,088** CITATIONS
SEE PROFILE

Ashok Agrawala
University of Maryland, College Park
302 PUBLICATIONS **7,698** CITATIONS
SEE PROFILE

2011 10th International Conference on Machine Learning and Applications

Thompson Sampling for Dynamic Multi-Armed Bandits

Neha Gupta
Department of Computer Science
University of Maryland
College Park 20742
Email: neha@cs.umd.edu

Ole-Christoffer Granmo
Department of ICT
University of Agder
Norway
Email: ole.granmo@uia.no

Ashok Agrawala
Department of Computer Science
University of Maryland
College Park 20742
Email: agrawala@cs.umd.edu

***Abstract*—The importance of multi-armed bandit (MAB) problems is on the rise due to their recent application in a large variety of areas such as online advertising, news article selection, wireless networks, and medicinal trials, to name a few. The most common assumption made when solving such MAB problems is that the unknown reward probability θ^k of each bandit arm k is fixed. However, this assumption rarely holds in practice simply because real-life problems often involve underlying processes that are dynamically evolving. In this paper, we model problems where reward probabilities θ^k are *drifting*, and introduce a new method called *Dynamic Thompson Sampling (DTS)* that facilitates Order Statistics based Thompson Sampling for these dynamically evolving MABs. The DTS algorithm adapts its success probability estimates, $\hat{\theta}^k$, faster than traditional Thompson Sampling schemes and thus leads to improved performance in terms of lower regret. Extensive experiments demonstrate that DTS outperforms current state-of-the-art approaches, namely pure Thompson Sampling, UCB-Normal and UCB_f, for the case of dynamic reward probabilities. Furthermore, this performance advantage increases persistently with the number of bandit arms.**

I. Introduction

The multi-armed bandit (MAB) problem forms a classical arena for the conflict between exploration and exploitation, well-known in reinforcement learning. Essentially, a decision maker iteratively pulls the arms of the MAB, one arm at a time, with each arm pull having a chance of releasing a reward, specified as the arm's reward probability θ^k. The goal of the decision maker is to maximize the total number of rewards obtained without knowing the reward probabilities. Although seemingly a simplistic problem, solution strategies are important because of their wide applicability in a myriad of areas.

Thompson Sampling based solution strategies have recently been established as top performers for MABs with Bernoulli distributed rewards [1]. Such strategies gradually move from exploration to exploitation, converging towards only selecting the optimal arm, simply by pulling the available arms with frequencies that are proportional to their probabilities of being optimal. This behavior is ideal when the reward probabilities of the bandit arms are fixed. However, in cases where the reward probabilities are dynamically evolving, referred to as *Dynamic Bandits*, one would instead prefer schemes that explore and track potential reward probability changes. Apart from the Kalman filter based scheme proposed in [2], the latter problem area is largely unexplored when it comes to Thompson Sampling. Another important obstacle in solving the problem is due to the fact that we cannot sample noisy instances of θ^k directly, as done in [2]. Instead, we must rely on samples obtained from Bernoulli trials with *reward probability* θ^k, which renders the problem unique.

In this paper, we introduce a novel strategy — Dynamic Thompson Sampling. Order Statistics based Thompson Sampling is used for arm selection, but the reward probabilities θ^k are tracked using an exponential filtering technique, allowing adaptive exploration. In brief, we explicitly model changing θ^k's as an integrated part of the Thompson Sampling, considering changes in reward probability to follow a Brownian motion – one of the most well-known stationary stochastic processes, extensively applied in many fields, including modeling of stock markets and commodity pricing in economics.

II. Related Work

In their seminal work on MAB problems, Lai and Robbins proved that for certain reward distributions, such as the Bernoulli-, Poisson-, and uniform distributions, there exist an asymptotic bound on regret that only depends on the logarithm of the number of trials and the Kullback-Leibler value of each reward distribution [3]. The main idea behind the strategy following from this insight is to calculate an upper confidence index for each arm. At each trial the arm which has the maximum upper confidence value is played, thus enabling deterministic arm selection. Auer et al. [4] further proved that instead of an asymptotic logarithmic upper bound, an upper bound could be obtained in finite time, and introduced the algorithms UCB-1, UCB-2 and their variants to this end. The pioneering Gittins Index based strategy [5] performs a Bayesian look ahead at each step in order to decide which arm to play. Although allowing optimal play for discounted rewards, this technique is intractable for MAB problems in practice.

Dynamic Bandits have also been known as *Restless Bandits*. Restless Bandits were introduced by Whittle [6] and are considered to be PSPACE-hard. Guha et al. [7] introduced approximation algorithms for a special setting of the Restless Bandit problem. Auer et al. [8] introduced a version of

978-0-7695-4607-0/11 $26.00 © 2011 IEEE
DOI 10.1109/ICMLA.2011.144

Restless Bandits called *Adversarial Bandits*, but the technique suggested was designed to perform against an all powerful adversary and hence led to very loose bounds for the reward probabilities.

In this work, we look at the problem of Dynamic Bandits in which the reward probabilities of the arms follow bounded Brownian motion. In [9], the authors consider a similar scenario of Brownian bandits with reflective boundaries, assuming that a sample from the current distribution of θ^k itself is observed at each trial. Granmo et al. introduced the Order Statistics based Kalman Filter Multi-Armed Bandit Algorithm [2]. In their model, reward obtained from an arm is affected by Gaussian noise $\sim N(0, \sigma_{ob}^2)$ and an independent Gaussian perturbations $\sim N(0, \sigma_{tr}^2)$ at each trial. A key assumption in [2] is again that at each trial a noisy sample of the true reward is observed. In contrast, in our work, estimation of the reward probabilities θ^k is done by only using Bernoulli outcomes $r^k \sim Bernoulli(\theta^k)$. Our work is thus well suited for modeling of problems such as click through rate optimization in the Internet domain, where a click on a newspaper article or advertisement results in a binary reward, from which the click through rate θ^k is estimated. Also, instead of using reflective boundaries we consider absorbing and "cutoff" boundaries, which are more suited for the Internet domain.

III. PROBLEM DEFINITION

A. Constant Rewards

For the MAB problems we study here, each pull of an arm can be considered as a Bernoulli trial having the output set $\{0, 1\}$, with the probability θ^k denoting the probability of success (event $\{1\}$). The probability distribution of the number of successes S obtained in n^k Bernoulli trials is known to have a Binomial distribution, $S \sim Binomial(n^k, \theta^k)$:

$$p(S = s|\theta^k) = \binom{n^k}{s}(1-\theta^k)^{n-s}(\theta^k)^s. \quad (1)$$

This means that since the Beta distribution is a conjugate prior for the Binomial distribution [10], Bayesian estimation is a viable option for estimating θ^k. It is thus natural to use the Beta distribution to obtain a prior fully specified by the parameters (α_0, β_0):

$$p(\hat{\theta}^k; \alpha_0, \beta_0) = \frac{x^{\alpha_0-1}(1-x)^{\beta_0-1}}{B(\alpha_0, \beta_0)}. \quad (2)$$

The posterior distribution after the n^{th} trial can be defined recursively. If a success is received at the n^{th} trial, α_n and β_n are identified as follows:

$$\alpha_n = \alpha_{n-1} + 1, \beta_n = \beta_{n-1}. \quad (3)$$

Conversely, if a failure is received at the n^{th} trial, we have:

$$\alpha_n = \alpha_n, \beta_n = \beta_{n-1} + 1. \quad (4)$$

After s successes and r failures, the parameters of the posterior Beta distribution thus become $(\alpha_0 + s, \beta_0 + r)$. The mean and variance of this posterior, $\text{Beta}(\alpha_0 + s, \beta_0 + r)$, can be used to characterize θ^k:

$$\hat{\mu}_n = \frac{\alpha_n}{\alpha_n + \beta_n} \quad (5)$$

$$\hat{\sigma}_n^2 = \frac{(\alpha_n\beta_n)}{(\alpha_n + \beta_n + 1)(\alpha_n + \beta_n)^2}. \quad (6)$$

B. Pure Thompson Sampling (TS)

Thompson Sampling is a randomized algorithm that takes advantage of Bayesian estimation to reason about the reward probability θ^k associated with each arm k of a MAB, as summarized in Algorithm 1. After conducting n MAB trials, the reward probability θ^k of each arm k is estimated using a posterior distribution over possible estimates, $\text{Beta}(\alpha_n^k, \beta_n^k)$ [11], and the state of a system designed for K armed MABs can therefore be fully specified by $\{(\alpha_n^1, \beta_n^1), (\alpha_n^2, \beta_n^2), ...(\alpha_n^K, \beta_n^K)\}$.

For arm selection at each trial, one sample $\hat{\theta}^k$ is drawn for each arm k from the random variable $\hat{\Theta}_n^k$, $\hat{\Theta}_n^k \sim \text{Beta}(\alpha_n^k, \beta_n^k), k = 1, 2, 3, ..., K$, and the arm obtaining the largest sample value is played. The above means that the probability of arm k being played is $P(\hat{\theta}^k > \hat{\theta}^1 \wedge \hat{\theta}^k > \hat{\theta}^2 \wedge \hat{\theta}^k > \hat{\theta}^3 ... \hat{\theta}^k > \hat{\theta}^K)$, however, the beauty of Thompson Sampling is that there is no need to explicitly compute this value. Formal convergence proofs for this method have been discussed in [1], [12].

Algorithm 1 Thompson Sampling (TS)

Initialize α_0^k=2, $\beta_0^k = 2$.
loop
  Sample reward probability estimate $\hat{\theta}^k$ randomly from $\text{Beta}(\alpha_{n-1}^k, \beta_{n-1}^k)$ for $k \in \{1, \ldots, K\}$.
  Arrange the samples in decreasing order.
  Select the arm A s.t. $\hat{\theta}^A = max_k\{\hat{\theta}^1, \ldots, \hat{\theta}^K\}$.
  Pull arm A and receive reward r_n.
  Obtain α_n^A and β_n^A: $\alpha_n^A = \alpha_{n-1}^A + r_n$; $\beta_n^A = \beta_{n-1}^A + (1 - r_n)$.
end loop

C. UCB Algorithm

The UCB-1 [4] algorithm computes an Upper Confidence Bound (UCB) for each arm k: $E[\hat{\theta}_n^k] + \sqrt{\frac{2\ln N}{n^k}}$. Here $E[\hat{\theta}_n^k]$ is the average reward obtained from arm k when the number of times arm k has been played is n^k and N is the overall number of trials so far. In this algorithm, the arm which produces the largest UCB is played at each trial, and the UCBs are updated accordingly.

UCB-Normal is a modification of the UCB-1 algorithm for the case of Gaussian rewards. The bounds used in UCB-Normal are $E[\hat{\theta}_n^k] + \sqrt{16 \cdot \frac{q^k - n^k(\hat{\theta}_n^k)^2}{n^k - 1} \frac{\ln(N-1)}{n^k}}$ where q^k is the sum of the square of the reward of arm k.

The UCB_f algorithm [9] is a more general form of the $\text{UCB} - 1$ algorithm that also incorporates Brownian motion with reflecting boundaries. In brief, the bound from

UCB-1 is extendend with an additional bound component: $\sigma^k\sqrt{8N\log\ N}$, where σ^k is the volatility of arm k.

D. Dynamically Changing Reward Probabilities

The key assumption made in most MAB algorithms is that the reward probabilities remain constant. In practice, it is rare to have constant reward probabilities, and the algorithm that we propose here explicitly takes into account changing reward probabilities.

Brownian motion is a simple stochastic process in which the value of a random variable at step n is the sum of its value at time $n-1$ and a Gaussian noise term $\sim N(0,\sigma^2)$. In this paper, we consider the time varying reward probability θ_n to follow a simple Brownian motion in the range $[0,1]$:

$$\theta_n = \theta_{n-1} + \nu_n, \nu_n \sim N(0,\sigma^2) \tag{7}$$

As θ is a probability, it must remain within [0, 1] — consequently, we need to bound the Brownian motion of the reward probabilities. We define two types of boundary properties:

- Cutoff Boundary: The reward probability is bounded between $[0,1]$ and once it reaches a boundary it remains there until the next outcome moves it out of the boundary.

$$\theta_n = \begin{cases} \theta_{n-1}+\nu_n & \text{if } 0 \geq \theta_{n-1}+\nu_n \leq 1 \\ 1 & \text{if } \theta_{n-1}+\nu_n > 1 \\ 0 & \text{if } \theta_{n-1}+\nu_n < 0 \end{cases}$$

- Absorbing Boundary: With absorbing boundaries θ_n remains at the boundary forever after reaching it.

$$\theta_n = \begin{cases} \theta_{n-1}+\nu_n & \text{if } \nexists i \leq n : \theta_i \geq 1 \vee \theta_i \leq 0 \\ 1 & \text{if } \exists i \leq n : \theta_i \geq 1 \\ 0 & \text{if } \exists i \leq n : \theta_i \leq 0 \end{cases}$$

The performance of MAB solution schemes can be measured in terms of $Regret$, defined as:

$$Regret = \Sigma_{n=0}^{N}(r_n^* - r_n^k).$$

Above, N is the total number of trials, r_n^* is the Bernoulli output one would receive by playing the arm with the highest θ_n^k at trial n, while r_n^k is the reward obtained after sampling the k^{th} arm as determined by the algorithm being evaluated. Note that the arm corresponding to r_n^* may change as the values of θ_n^k evolves. Hence, regret is a measure of the loss suffered by not always playing the optimal arm.

E. Dynamic Thompson Sampling Algorithm (DTS)

Unlike the algorithms for static MAB problems, the goal of the DTS algorithm proposed presently is to minimize the regret by tracking the changing values of θ_n^k as closely as possible. Note that in our model θ_n^k changes according to Eqn. 7 whether arm k is played or not. The DTS algorithm is able to track reward probabilities by replacing the update rules specified in Eqn. 3 and 4 by two set of update rules and a threshold C governing which set of update rules to use:

1) If $\alpha_{n-1}+\beta_{n-1} < C$,

$$\alpha_n = \alpha_{n-1} + r_n \tag{8}$$

$$\beta_n = \beta_{n-1} + (1-r_n) \tag{9}$$

2) If $\alpha_{n-1}+\beta_{n-1} \geq C$,

$$\alpha_n = (\alpha_{n-1}+r_n)\frac{C}{C+1} \tag{10}$$

$$\beta_n = (\beta_{n-1}+(1-r_n))\frac{C}{C+1} \tag{11}$$

Notice that the first set of update rules makes the scheme behave identical to Pure Thompson Sampling when $\alpha_{n-1}+\beta_{n-1} < C$, while the second set of update rules for $\alpha_{n-1}+\beta_{n-1} \geq C$ ensures that $\alpha_n+\beta_n$ never grows above C. I.e., for $\alpha_{n-1}+\beta_{n-1} = C$ we have:

$$\alpha_n+\beta_n = (\alpha_{n-1}+\beta_{n-1}+1)\frac{C}{C+1} \tag{12}$$

$$= (C+1)\frac{C}{C+1} \tag{13}$$

$$= C. \tag{14}$$

Also, by updating the values of α_n, β_n according to rule set 2) above, more weight will be assigned to the more recent rewards as opposed to older rewards. That is, if we continue substituting the value of α_{n-1} in Eqn. 10 above, we get

$$\alpha_n = \left((\alpha_{n-2}+r_{n-1})\frac{C}{C+1}+r_n\right)\frac{C}{C+1} \tag{15}$$

$$= \alpha_{n-2}(\frac{C}{C+1})^2 + r_{n-1}(\frac{C}{C+1})^2 + r_n\frac{C}{C+1} \tag{16}$$

whence, it becomes apparent that the weighting is exponential.

To summarize, the above strategy provides exponential weighting of the outcomes of the trials, with the more recent outcomes getting more weight. In the same manner, we could express β_n as a discounted sum of previous outputs of the Bernoulli trials. Similarly, we observe that the mean μ_n of $\mathrm{Beta}(\alpha_n,\beta_n)$ at trial n is,

$$\mu_n = \frac{\alpha_n}{\alpha_n+\beta_n} \tag{17}$$

$$= \frac{\alpha_{n-1}+r_n}{C} \times \frac{C}{C+1} \tag{18}$$

$$= \frac{\alpha_{n-1}}{C}\frac{C}{C+1} + r_n\frac{1}{C+1} \tag{19}$$

$$= \frac{C}{C+1}\frac{\alpha_{n-1}}{\alpha_{n-1}+\beta_{n-1}} + \frac{1}{C+1}r_n \tag{20}$$

$$= \Delta\cdot\mu_{n-1} + (1-\Delta)r_n \tag{21}$$

where $\Delta = \frac{C}{C+1}$. Clearly, this approach yields *exponential filtering* of r_n [13]. Observe finally that the variance σ_n^2 of $\mathrm{Beta}(\alpha_n,\beta_n)$ is bounded as follows:

$$0 \leq \sigma_n^2 \leq \frac{1}{4(C+1)}. \tag{22}$$

This is the case because the variance of $\mathrm{Beta}(\alpha_n,\beta_n)$ is

$$\sigma_n^2 = \frac{(\alpha_n\beta_n)}{(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)^2}.$$

and because the product of α_n and β_n is maximized when $\alpha_n = \beta_n = C/2$ and minimized when either α_n or β_n approaches 0.

Algorithm 2 Dynamic Thompson Sampling (DTS)

loop
  Sample reward probability estimate $\hat{\theta}^k$ randomly from $\text{Beta}(\alpha^k_{n-1}, \beta^k_{n-1})$ for $k \in \{1, \ldots, K\}$.
  Arrange the samples in decreasing order.
  Select the arm A s.t. $\hat{\theta}^A = max_k\{\hat{\theta}^1, \ldots, \hat{\theta}^K\}$.
  Pull arm A and receive reward r_n.
  if $\alpha^A_{n-1} + \beta^A_{n-1} < C$ **then**
    $\alpha^A_n = (\alpha^A_{n-1} + r_n)$, $\beta^A_n = \beta^A_{n-1} + (1 - r_n)$.
  else
    $\alpha^A_n = (\alpha^A_{n-1} + r_n)\frac{C}{C+1}$,
    $\beta^A_n = (\beta^A_{n-1} + (1 - r_n))\frac{C}{C+1}$.
  end if
end loop

The DTS algorithm introduced in this paper is based on the above two sets of update rules and is specified in Algorithm 2 for the K-armed bandit case, where the motion of the corresponding reward probabilities $(\theta^1, \theta^2, \theta^3, .., \theta^K)$ is Brownian. The algorithm starts by initializing the priors α^k_0=2, $\beta^k_0 = 2$ for all the arms, and then proceeds by gradually updating the α^ks and β^ks as penalties and rewards are received. Because of the exponential weighting of rewards, drifting reward probabilities are tracked, which in turn leads to a better performance as will be shown presently.

IV. Experiments

In this section, we primarily evaluate the performance of the DTS algorithm by comparing it with UCB_f, TS and UCB-Normal. Even though we have performed a large number of experiments using a wide range of reward distributions, we here only report the most important and relevant ones due to limited space. We report the regret obtained as the measure of performance of the different algorithms. As DTS is a randomized algorithm, the regret becomes a random variable. The expected value of the regret is estimated by repeating each experiment 400 times.

A. Varying value of standard deviation σ

To get an insight into the Brownian motion of the reward probability θ, we performed experiments in which we simulated the dynamics of θ for different values of standard deviation. In Fig. 1, we show a sample plot of the curves for 4 values of $\sigma = \{0.05, 0.01, 0.005, 0.001\}$ starting with $\theta_0 = 0.5$. The curve with standard deviation $\sigma = 0.05$ is cutting across the boundaries 0 and 1 very often and accurate learning seems unrealistic in this situation. The other graphs with standard deviations $\sigma = \{0.01, 0.005, 0.001\}$ are more stable and seem more appropriate to model realistic learning problems.

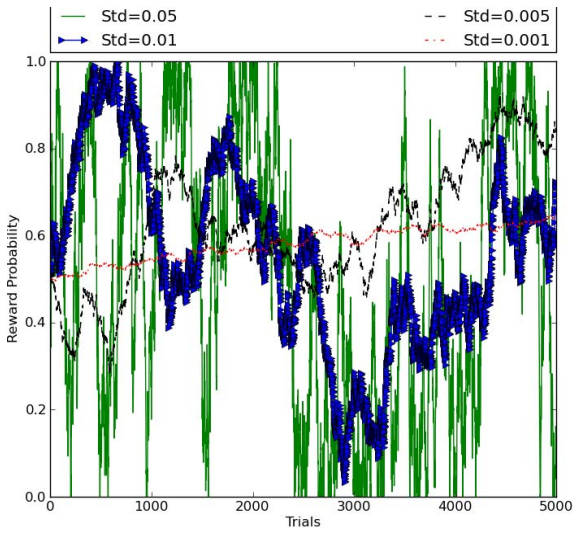

Fig. 1. Typical variations of the reward probability θ for different values of standard deviations. $\theta_0 = 0.5$ in all cases.

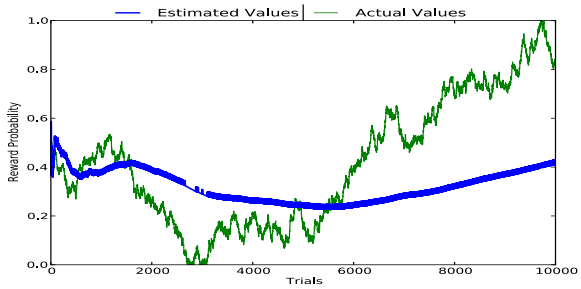

Fig. 2. Plot shows the estimated and actual values of θ for the case of a single arm. Estimated values are calculated based on TS algorithm.

B. Estimation vs. Actual

We perform these experiments to show how closely the estimated values of $\hat{\theta}$ are to the actual value of θ for the case of TS and DTS algorithms for a single arm. The two graphs, Fig. 2 and Fig. 3, show the results for the estimated and actual values of θ. We see that the DTS algorithm provides a muche more accurate estimate of θ based on its exponential filtering, when compared to the TS algorithm.

C. Tuning parameter C for DTS algorithm

Fig. 4 shows a plot of the root mean square error (RMSE) obtained for different values of C and standard deviation σ for 10, 000 trials in the DTS and TS algorithm for a single arm. RMSE is measured as :

$$RMSE = \sqrt{\frac{\Sigma_{n=1}^{N}(\theta_n - \hat{\theta}_n)^2}{N}} \tag{23}$$

Note here that RMSE values averaged over 400 runs are reported in the graph. In this experiment, we take two different values of $\theta = \{0.8, 0.5\}$ and choose the standard deviation

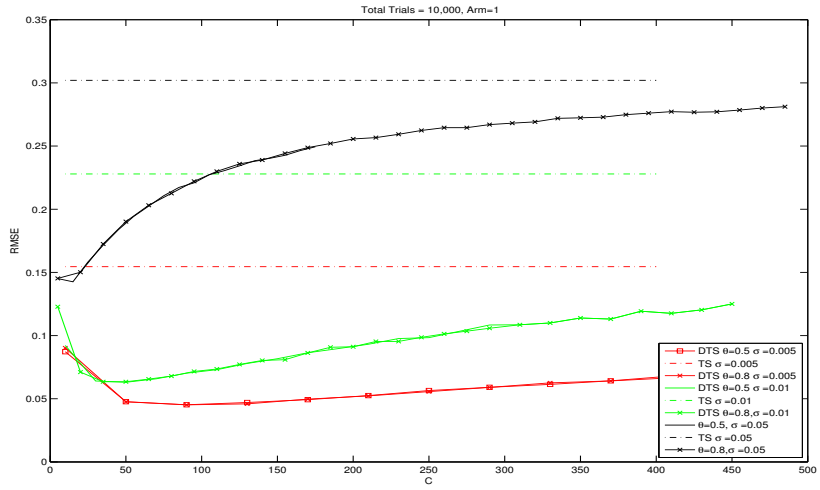

Fig. 4. Plots for RMSE for two different values of θ, 3 different values of standard deviation σ and with/without the exponential filtering for θ

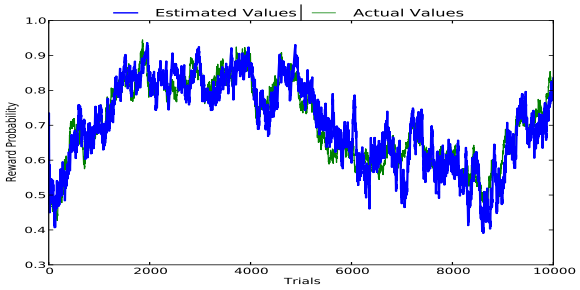

Fig. 3. Plot shows the estimated and actual values of θ_t for the case of a single arm. Estimated values are calculated based on DTS algorithm.

in the set $\{0.005, 0.01, 0.05\}$. We notice that the graphs for different values of θ, but same standard deviation, are overlapping. We also observe that the value at which the RMSE is minimum drops with increasing σ. This is because higher values of σ leads to more dynamic arm probabilities, hence a shorter reward history is required for estimating θ.

We next present an empirical evaluation of the different MAB algorithms using tuned values of the model parameters.

D. Varying Standard Deviation

In the first experiment to evaluate the performance of different MAB strategies, we vary the standard deviation σ of θ. We consider a total of 10 arms, $\theta_{opt} = 0.6$, and all the other 9 arms are generated from Uniform distribution $\mathrm{U}(0.6, 0)$. Fig. 5, 6 show the regret obtained by using different standard deviations for the SR method for 10, 000 trials for the case of cutoff and absorbing boundaries. The different values of the standard deviation are {0.001, 0.005, 0.008, 0.01, 0.02}. We see that the DTS algorithm shows the least regret as compared to other MAB strategies for both the cases of absorbing as well as cutoff boundaries.

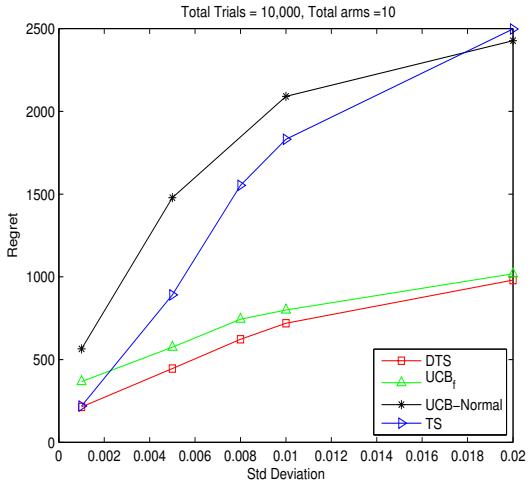

Fig. 5. Plots of Regret comparing DTS with UCB_f, UCB-Normal and TS algorithms for the case of Cutoff Boundaries

E. Changing the number of arms

We perform this experiment to show the effect of increasing the number of arms on the regret obtained for the case of Brownian bandits. We set $\theta_{max} = 0.6$ and initially randomly generate a set of 9 arms with reward probabilities in interval $(0.6, 0)$ using Uniform distribution, and add four arms from the same set $\mathrm{U}(0.6, 0)$ for a total of 10K trials. We use $\sigma = 0.005$ as standard deviation for the DTS algorithm. As shown in Fig. 7 and 8, the DTS algorithm performs much better than the UCB_f, Thompson Sampling and UCB-Normal algorithm. The difference between UCB_f and DTS algorithm grows as the number of arms increase which shows that the UCB_f algorithm does not scale with the number of arms for either

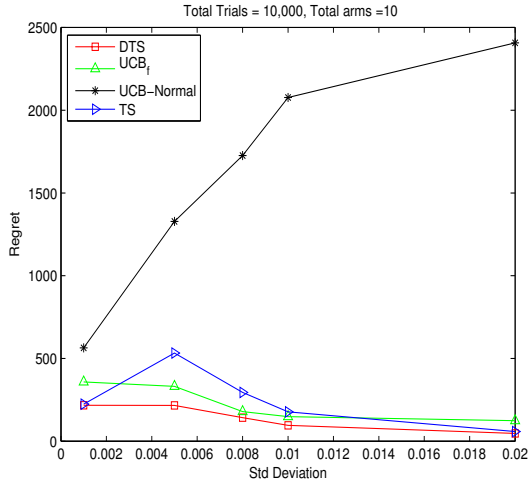

Fig. 6. Plots of Regret comparing DTS with UCB_f, UCB-Normal and TS algorithms for the case of Absorbing Boundaries

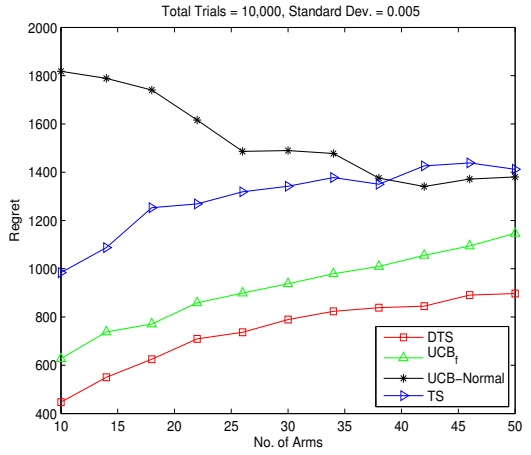

Fig. 7. Plots of Regret comparing DTS with UCB_f, UCB-Normal and TS algorithms for the case of Cutoff Boundaries

absorbing and cutoff boundaries. We do not show the results of UCB-Normal algorithm in Fig. 8 as it consistently shows poor results for the case of absorbing boundaries also.

V. Conclusion

In this paper, we presented the *Dynamic Thompson Sampling (DTS)* algorithm. DTS builds upon the Order Statistics based Thompson Sampling framework by extending the framework with exponential filtering capability. The purpose is to allow dynamically changing reward probabilities to be tracked over time. The experimental results and analysis presented in this paper show that the DTS algorithm significantly outperforms current state-of-art methods such as UCB_f, Thompson Sampling and UCB-Normal for the case of dynamic reward probabilities possessing bounded Brownian motion. We also observe an increasing performance improvement as the number of arms increases, which demonstrates the usefulness of our proposed algorithm in large-scale MAB problems. The DTS strategy can be further extended to include variations such as mortal bandits, hierarchical bandits, as well as strategies for identifying the k-best arms by introducing immunity from elimination. We are also working on proving the theoretical bounds of the DTS algorithm.

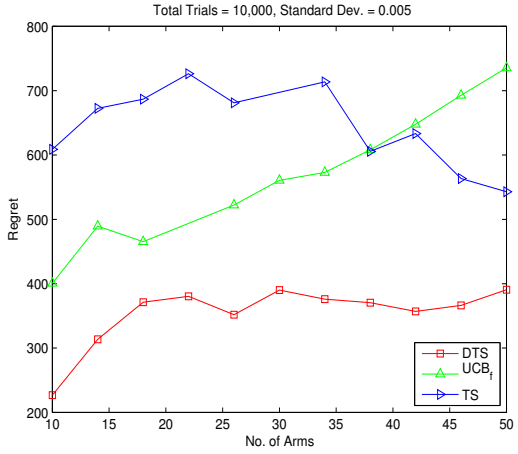

Fig. 8. Plots of Regret comparing DTS with UCB_f, UCB-Normal and TS algorithms for the case of Absorbing Boundaries

References

[1] O.-C. Granmo, "Solving two-armed bernoulli bandit problems using a bayesian learning automaton," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 3, pp. 207–234, 2010.

[2] O.-C. Granmo and S. Berg, "Solving non-stationary bandit problems by random sampling from sibling kalman filters," in *Twenty Third International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA-AIE 2010)*, 2010.

[3] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive bandit rules," *Advances in Applied Mathemetics*, 1985.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite time analysis of multi-armed bandit problem," *Machine Learning*, vol. 27, no. 2-3, pp. 235–256, 2002.

[5] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of Royal Statistical Society. Series B*, vol. 41, no. 2, pp. 148–177, 1979.

[6] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of Applied Probability*, vol. 25, pp. pp. 287–298, 1988. [Online]. Available: http://www.jstor.org/stable/3214163

[7] S. Guha, K. Munagala, and P. Shi, "Approximation algorithms for restless bandit problems," *CoRR*, vol. abs/0711.3861, 2007.

[8] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *36th Annual Symposium on Foundations of Computer Science*. IEEE, 1995.

[9] A. Slivkins and E. Upfal, "Adapting to changing environment: the brownian restless bandits," in *COLT*, 2008.

[10] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and its applications*. New York: Marcer Dekker Inc., 2004.

[11] J. Wyatt, "Exploration and inference in learning from reinforcement," *Ph.D. thesis, University of Edinburgh*, 1997.

[12] B. C. May, N. Korda, A. Lee, and D. S. Leslie, "Optimistic bayesian sampling in contextual-bandit problems," *Submitted to the Annals of Applied Probability*.

[13] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting Methods and Applications*, 3rd ed. John Wiley and Sons, Inc., 1998, chapter 4, pp. 135–179.