CS 221 Project Final Report: Exploring Bandit Algorithms

David Chen
Stanford University
dchen11@stanford.edu

***Abstract*—This report outlines an ongoing project on exploring bandit algorithms within the classic stochastic multi-armed bandit framework. The project's goal is to provide a comprehensive comparison of algorithms through reproduction of existing empirical analysis of regret and runtime, qualitative observations of behavior, and theoretical comparison of proven regret bounds. To date, a simulation framework has been developed and populated with algorithms such as random, ε-greedy, explore-then-commit, Bayes-UCB, Thompson sampling, and information-directed sampling, with a set of empirical results conducted in various independent and linear bandit settings.**

I. Introduction

In the broad context of online decision-making under uncertainty, the class of problems known as multi-armed bandits (MABs) has provided a rich environment for insightful theoretical analysis and applicable algorithms. Multi-armed bandit problems have been studied in a wide variety of fields, ranging from computer science and statistics, to operations research and economics. Although closely related to the more general setting of reinforcement learning, the study of MABs typically has a strong focus on the classic dilemma of exploration-exploitation. Still, fundamental insights from theoretical developments have been further developed for more complex reinforcement learning settings, and simple yet effective bandit algorithms have been deployed for many practical use-cases in the industry with great success.

A. Project goals and final progress

The primary focus of this course project is to provide a comprehensive comparison of a collection of algorithms for the classic stochastic multi-armed bandit setting. Namely, I do not focus on settings such as those of contextual or adversarial bandits.

I have implemented a functioning framework for simulation of various bandit algorithms, and reproduced a subset of the simulation results presented in Russo and Van Roy [1].

Algorithms implemented include ε-greedy, explore-then-commit (ETC), Bayes-UCB, and Thompson sampling (TS), and variance-based information-directed sampling (V-IDS). These algorithms are tested in the context of various independent settings, such as Bernoulli, Gaussian, and Poisson bandits, as well as a linear Gaussian setting.

I have also included a short selection of important prerequisite knowledge and key theoretical results related to the above algorithms.

B. Brief overview of the multi-armed bandit problem.

I examine the stochastic multi-armed bandit problem. At each time period, an agent is allowed to choose an action (an arm) to execute, and subsequently observes a random outcome, often in the form of a scalar reward. Outcomes are associated with the specific arm, and can either be *independent or dependent* with respect to the other arms. The true distribution of the outcomes is unknown, and thus exploration of arms is necessary in order to gather knowledge about rewards. In this project, I restrict analysis to settings with stationary outcome distributions over time, as well as restricting the class of eligible actions to be fixed finite sets.

The objective of MAB problems is to maximize the average cumulative reward over time. Thus, a central issue that arises is that of *exploration-exploitation*, where a tradeoff is necessary in order to discover actions associated with higher rewards, while still leveraging high reward actions over the time horizon. In general, the time horizon can be infinite, but I only analyze and implement problems in a finite-time setting for this project.

A key difference between MABs and the more general reinforcement learning framework is the lack of "state". In the setting of Markov decision processes, outcomes are associated with a changing state as well as the selected action, whereas in the restricted bandit setting, any given action is assumed to produce i.i.d. outcomes when chosen in different time periods.

Typical theoretical analysis of MABs often involves the notion of *regret*, which intuitively is the expected difference in the sum of rewards between a strategy that chooses the optimal action at every round, and the actual strategy. There is also the notion of *per-period regret*, which is specific to a single round. Upper and lower bounds on regret are of interest for various algorithms, and much of the literature is dedicated to deriving and improving these bounds.

II. Related Works

The first formulation of the multi-armed bandit problem is most commonly attributed to a paper from Robbins in 1952 [2]. Since then, numerous techniques and settings have appeared in the literature. The introduction of "upper confidence bound" strategies as an approach to more efficient exploration appeared in Lai and Robbins [3].

Many early approaches were more aligned with the frequentist perspective, and extensions of the idea of upper confidence bounds resulted in algorithms such as UCB1 [4], which also proved upper bounds on the cumulative regret

that scaled logarithmically with time. Over time, analysis for the Bayesian approach also gained popularity, such as the Bayes-UCB approach introduced by Kaufmann et al. [5].

Around the same time, an approach known as Thompson sampling started gaining recognition in the context of MABs. Thompson sampling itself pre-dated the formal bandit definition, first introduced by Thompson in 1933 [6]. In the last couple of decades, theoretical and experimental analysis demonstrated competitive performance in the context of bandits [7].

Eventually, this culminated in an elegant approach to deriving upper bounds on regret for Thompson sampling using concepts from information theory. Russo and Van Roy introduced the concept of the *information ratio*, which was used to prove general bounds that depended on the entropy of the prior distribution of the optimal action [8].

The information ratio turned out to be quite useful beyond a one-time analysis of Thompson sampling; Russo and Van Roy developed a novel algorithm that explicitly minimized the information ratio during the decision-making process, and provided theoretical and experimental results that demonstrated its superiority over Thompson sampling in various settings [1]. It is this paper that I take inspiration in terms of reproduction of simulation results.

Finally, there are multiple other resources that have gathered results and techniques across the field of bandits as a whole, including extensions such as contextual, adversarial, and many other related settings [9], [10]. Some background and theoretical results are inspired by the contents of these comprehensive texts from Slivkins [10], and Lattimore and Szepesvári [9].

III. Methodology

A. Problem formulation

We work with a probability space $(\Omega, \mathbb{F}, \mathbb{P})$, and all random variables are defined with respect to this space, including the random variables that model prior uncertainty as described commonly in the Bayesian formulation.

The agent chooses actions $(A_t)_{t\in\mathbb{N}}$ from a finite set $\mathcal{A}$, and subsequently observes the outcomes $\left(Y_{t,A_t}\right)_{t\in\mathbb{N}}$, where each $Y_{t,a} \in \mathcal{Y}$. We assume, according to the Bayesian perspective, that there is a random element θ that describes the true distribution of outcomes, such that conditioned on θ, the sequence $(Y_t)_{t\in\mathbb{N}} = \left((Y_t, a)_{a\in\mathcal{A}}\right)_{t\in\mathbb{N}}$ is independent and identically distributed.

Furthermore, the agent observes a reward associated with the outcome. In many cases, the reward and outcome are equivalent, but generally, reward can be a known function $R : \mathcal{Y} \to \mathbb{R}$. For convenience, we can denote $R_{t,a} := R\big(Y_{t,a}\big)$.

Once we have the notion of reward, we can define the optimal action(s) to be A^* such that $A^* \in \arg\max_{a\in\mathcal{A}} \mathbb{E}\big[R_{t,a} \,\big|\, \theta\big]$. Building on top of this, we can finally define the T-period *regret* of a strategy of choosing actions π to be:

$$\text{Regret}(T, \pi) = \sum_{t=1}^{T} \Big(R_{t,A^*} - R_{t,A_t}\Big),$$

where the sequence of actions is understood to be chosen by π. We can take an expectation on both sides, with respect to randomness in the choice of actions, outcomes, and over the prior distribution of θ, which leaves us with the *expected regret*.

In general, $\pi = (\pi_t)_{t\in N}$ is understood to be a sequence of functions that take in the history $\mathcal{H}_t = \big(A_1, Y_{1,A_1}, ..., A_{t-1}, Y_{t-1,A_{t-1}}\big)$, and outputs a probability distribution over the set of actions $\mathcal{A}$.

The history is important for the Bayesian formulation, where commonly, a posterior distribution is updated as more data is collected. For example, an estimate of the parameter θ of an unknown Bernoulli distribution corresponds to a conjugate prior Beta distribution, which has parameters that are simple to update given incoming reward observations.

Further concepts which are useful to mention are fundamental concepts in information theory.

The *Shannon entropy* of a discrete random variable is defined as follows:

$$H(X) = \sum_{x\in\mathcal{X}} \mathbb{P}(X = x) \log \frac{1}{\mathbb{P}(X = x)}.$$

The *Kullback-Leibler divergence* $D_{\text{KL}}(P \,\|\, Q)$ between two probability measures P and Q (given P is absolutely continuous with respect to Q) is:

$$D_{\text{KL}(P\,\|\,Q)} = \int \log\left(\frac{dP}{dQ}\right) dP.$$

The *mutual information* $I(X;Y)$ with respect to random variables X and Y can be expressed as an expectation over X involving the KL-divergence:

$$\begin{aligned} &I(X;Y) \\ &= \sum_{x\in X} \mathbb{P}(X = x) D_{\text{KL}}(\mathbb{P}(Y \in \cdot \mid X = x) \,\|\, \mathbb{P}(Y \in \cdot)). \end{aligned}$$

Finally, I describe the *information ratio* as first introduced in [8] and utilized in IDS [1]. Let $\Delta_t(a)$ denote the expected regret of an action a at timestep t. Letting A^* denote the optimal action[1] , we can define $\Delta_t(a)$ as the following:

$$\Delta_t(a) := \mathbb{E}\big[R_{t,A^*} - R_{t,a} \,\big|\, \mathcal{H}_t\big].$$

We can also define the *information gain* of an action to be:

$$g_t(a) := I_t\big(A^*; Y_{t,a}\big).$$

The information ratio for a given action is then $\frac{\Delta_t(a)^2}{g_t(a)}$.

B. Dataset (bandit simulation)

All relevant "data" is generated online (ad-hoc) during simulations. Outcomes associated with specific distributions are generated randomly using existing libraries, such as numpy.random and scipy.stats.

For example, in a Bernoulli bandit instance, the initial parameters θ_k across K arms are generated independently

[1]Note that the optimal action is unknown, thus we represent it with a random variable.

from a continuous uniform distribution on $[0, 1]$ using np.random.uniform. The outcomes/rewards are generated according to another random function such as np.random.rand. All subsequent rewards are then generated as needed during simulation.

C. Baseline (simple algorithms)

The baseline involves implementation of simple non-adaptive exploration algorithms for the beta-Bernoulli bandit setting. The "non-adaptive exploration" terminology is borrowed from Slivkins [10]. These include a random strategy, various ε-greedy strategies, and explore-then-commit.

The random strategy simply chooses an action uniformly at random from the set of available actions, at each time-step. This is mainly chosen to demonstrate a worst-case upper bound on regret for all subsequent algorithms.

The ε-greedy algorithms involve choosing a uniformly random action at each time period with probability ε_t, and otherwise choosing the action with the maximum point-estimate of the mean reward. Notably ε_t can vary over time, but overall this class of algorithms is still classified into the non-adaptive exploration category, given it does not change its exploration strategy based on the realized history.

Some examples of valid choices of ε_t are:

- Constant (ex. $\varepsilon_t = \varepsilon \in [0, 1)$),
- Decaying (ex. $\varepsilon_t = \varepsilon(t) = t^{-1/3}$),
- Explore-then-commit (ex. $\varepsilon_t = \varepsilon(t) = \mathbb{1}_{t<200}$).

These approaches are chosen as the baseline of algorithms that do not incorporate any additional notion of uncertainty into the exploration strategy, which leads to provably worse regret-bounds and demonstrably worse realized regret in simulation.

D. Main approach (advanced algorithms)

More interesting algorithms arise when we attempt to balance exploration-exploitation through use of optimism, probability matching, or explicitly minimizing the information ratio.

A major family of algorithms in this area are the **UCB algorithms**, which range from frequentist algorithms such as UCB1 to the Bayesian Bayes-UCB.

On the other hand, **Thompson sampling** selects an action based off of the statistical possibility that it is optimal under the posterior distributions.

Finally, I examine **information-directed sampling**, where the action is chosen to minimize an information ratio based on the expected regret and mutual information. Specifically, I implement a variant of IDS using the variance $\mathrm{Var}_t\big(\mathbb{E}_t\left[R_{t,a} \mid A^*\right]\big)$. In other words, this is the variance of the expected reward of an action over different realizations of the optimal action. When substituted into the information-ratio in place of $g_t(a)$, the variance-based information ratio provides an upper bound on the information ratio, and has been proven to satisfy the same bounds as original IDS.

These algorithms have all been shown to have improved upper-bounds for regret compared to the baseline algorithms, and have also demonstrated better performance in simulation.

E. Additional settings

While the beta-Bernoulli bandit setting provides useful insights by itself, extension of analysis to a wider variety of settings may provide a more complete picture of the capabilities of all the algorithms, as well as distinguish algorithms that are capable of taking advantage of settings where there exists a richer information structure.

To that end, I believe it will be worthwhile to implement the following additional settings:

- *Independent Gaussian*, where the reward for each arm follows a Gaussian distribution with a fixed known variance, and the mean parameters are assumed to be independent samples from a Gaussian prior.
- *Independent Poisson*, where the reward for each arm follows a Poisson distribution, and the rate parameters are assumed to be independent samples from a Gamma prior.
- *Linear Gaussian*, where actions $a \in \mathbb{R}^d$ are known d-dimensional vectors. The rewards correspond to $a^\top \theta + \epsilon_t$, where θ is unknown and drawn from a multivariate Gaussian prior. Gaussian noise is added in the form of ϵ_t with fixed and known variance.

F. Evaluation

For the independent settings, I evaluate all algorithms over 2000 simulations (trials), each running for $T = 2000$. For each trial, we calculate the cumulutative sum over time, and that sequence is then averaged over all trials.

For the linear Gaussian setting, I evaluate some algorithms over $T = 250$, given the heavy amount of computation required and some unresolved issues preventing parallel computation.

I also briefly touch upon the computational runtimes of the algorithms. Some algorithms, such as ε-greedy, only rely on a few elementary operations each iteration, while some, like IDS, involve more intensive numerical methods to approximate integrals, or MCMC based techniques to directly approximate certain expectations and probabilities.

A final point of comparison can be done theoretically through best known regret bounds in similar settings. Derivations will be provided for selected results, and comparison between algorithms as well as their empirical results will be shown.

IV. Theoretical comparison

Empirical performance of an algorithm can be complemented with theoretical results. There are a few important results one often cares about, but two of the most important are upper and lower bounds on regret.

These can vary between different settings, and often settings with more structure and more ways to gather data can lead to provably better bounds than the more general case. Indeed, it is possible to show that in settings with more feedback, such as either full or partial feedback, that an algorithm such as IDS will accumulate less regret, theoretically, than algorithms that do not take advantage of this information.

In this section, I compare the upper bounds on regret for the algorithms I implemented for simulation. For all results, I assume that rewards are bounded (by 0 and 1 for simplicity).

- *random and greedy:*

  These can be considered special suboptimal cases of ε-greedy, and have cumulative regret that scales linearly with time. See below.

- *ε-greedy:*

  Depending on the value of ε, the regret can vary from linear in time to sublinear. If we choose a constant $0 < \varepsilon \leq 1$, we can see how linear cumulative regret can arise. If, over a time horizon T, we take a completely random action for about εT rounds. That itself gives us an upper bound on regret, and it is clearly linear.

  However, for a value of ε that decreases over time, such as $\varepsilon = t^{-1/3} \cdot (K \log t)^{1/3}$, we can prove the following sublinear regret bound:

  $$\mathbb{E}[\text{Regret}(T, \pi_\varepsilon)] \leq O\big(T^{2/3} \cdot (K \log T)^{1/3}\big).$$

  Proof:

  Fix round t, and define the clean event for a given arm as the following:

  $$|\bar{\mu}_t(a) - \mu(a)| \leq \sqrt{\frac{2K \log t}{t\varepsilon_t}} = r_t(a),$$

  where $\bar{\mu}(a)$ is the current estimate of the mean of arm a, and $\mu(a)$ is the true mean. On average, we end up exploring any given arm around $\frac{t\varepsilon_t}{K}$ times by round t. Note we cannot apply Hoeffding's inequality immediately, given that the number of times we choose a is not fixed, and may even not be independent from the samples of a. To fix this, we can just let $v_j(a)$ be the average of the first j times we would have chosen a, regardless of t or the actual number of times we choose a.

  With this independence fix, now we can apply Hoeffding's inequality. We get the following:

  $$\forall j, \quad \mathbb{P}\big(\big|v_j(a) - \mu(a)\big| \leq r_t(a)\big) \geq 1 - \frac{2}{t^4}.$$

  We can then proceed by taking two union bounds, one over all j, and then one over all actions. Assuming the current round t is more than the number of arms K, This results in the following:

  $$\mathbb{P}(\forall a,\ |\bar{\mu}_t(a) - \mu(a)| \leq r_t(a)) \geq 1 - \frac{2}{t^2}.$$

  Let's call this union of clean events for all arms **the** clean event, and assume it for the rest of the proof. Now assume that for round t, we do not explore, and we instead exploit arm a. In the worst case, we do not choose the optimal arm a^*. Then we have the following bound on the instantaneous regret:

  $$\begin{aligned}\mu(a) + r_t(a) &\geq \bar{\mu}_t(a) > \bar{\mu}_t(a^*) \\ &\geq \mu(a^*) - r_t(a^*),\end{aligned}$$

  which we can rearrange to get:

  $$\mu(a^*) - \mu(a) < r_t(a) + r_t(a^*) = O\left(\sqrt{\frac{K \log t}{t\varepsilon_t}}\right).$$

  The probability of exploring is ε_t, and the instantaneous regret is upper bounded by 1, so therefore we have:

  $$\begin{aligned}\mathbb{E}[\text{Regret}(t, \pi_\varepsilon) \mid \text{clean}] &= \varepsilon_t + (1 - \varepsilon_t) \cdot O\left(\sqrt{\frac{K \log t}{t\varepsilon_t}}\right) \\ &\leq \varepsilon_t + \left(\sqrt{\frac{K \log t}{t\varepsilon_t}}\right) \\ &\leq O\big(T^{2/3} \cdot (K \log T)^{1/3}\big),\end{aligned}$$

  where the last inequality is a result of plugging in the value for ε_t.

  The probability of the "non-clean" event is $O(t^{-2})$, and the total possible regret in that scenario is t. So we can safely ignore the "non-clean" event. Therefore, for all t, including T, we have our result. ■

  This proof was re-derived from scratch, and was originally given as exercise 1.2 from Slivkin's book [10].

- *Bayes UCB:*

  Kaufmann et al. proved [5] in the beta-Bernoulli case that regret is upper-bounded such that:

  $$\mathbb{E}[\text{Regret}(T, \pi_{\text{TS}})] \leq \tilde{O}\big(\sqrt{KT}\big),$$

  where $\tilde{O}$ indicates that logarithmic factors are ignored.

- *Thompson sampling and IDS:*

  Thompson sampling under bandit feedback was proven in Russo and Van Roy [8] to have the following regret bound:

  $$\mathbb{E}[\text{Regret}(T, \pi_{\text{TS}})] \leq \sqrt{\frac{1}{2} K H(A^*) T}.$$

  Furthermore, in a setting such as the linear bandit problem, it was shown to have the improved bound:

  $$\mathbb{E}[\text{Regret}(T, \pi_{\text{TS}})] \leq \sqrt{\frac{1}{2} \log(K) d T}.$$

  Russo and Van Roy showed [1] that information-directed sampling shares the same regret bounds. However, as we will soon see, it often outperforms Thompson sampling in practice.

  In addition, replacing the information gain in the information ratio with the aforementioned variance of conditional expected rewards results in the same upper bounds on cumulative regret.

V. Results and Discussion

I now present a series of simulation results which serve as the main result of my exploration for this project. I first display figures displaying cumulative regret with repsect to time, and additionally, I display those same results but with both axes in log scale to further illustrate the relation of the regret with time.

All simulations are run for 2000 trials, with $K = 30$, and 2000 timesteps except when otherwise specified.

A. *Independent Beta-Bernoulli*

For Bernoulli bandits, I first present all algorithms on one graph, and then focus on only Thompson sampling, Bayes UCB, and V-IDS for further figures and settings.

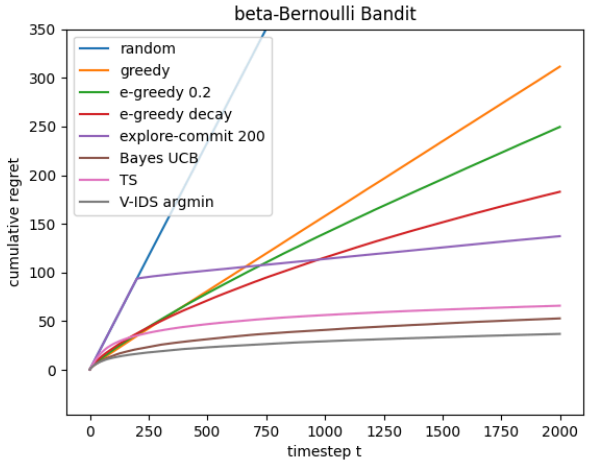

Fig. 1: Full comparison over all algorithms in the independent Bernoulli setting. For clarity, future figures omit most of the baseline algorithms.

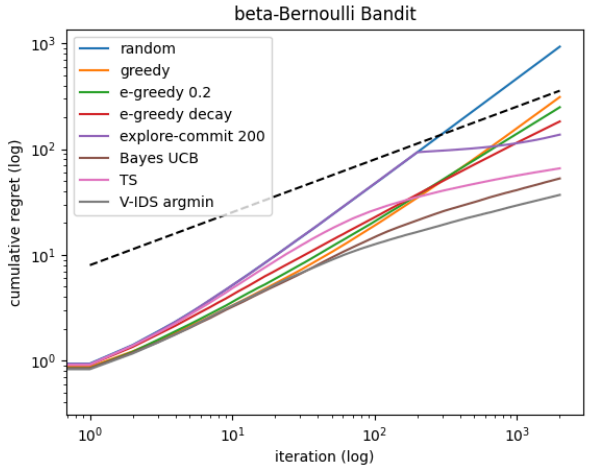

Fig. 2: Log scale view of Fig. 1. The dotted line has a slope of $\frac{1}{2}$, and represents a function that scales like square root of time.

In the log-scale figure, it is easy to compare the slopes of the cumulative regret plots with the given reference function. The algorithms with linear regret (random, greedy, constant e-greedy) have slopes of 1. The algorithms with regret that scales with square root of time (or better) have slops of $\frac{1}{2}$ or lower.

For clarity, I present just four algorithms in a separate figure:

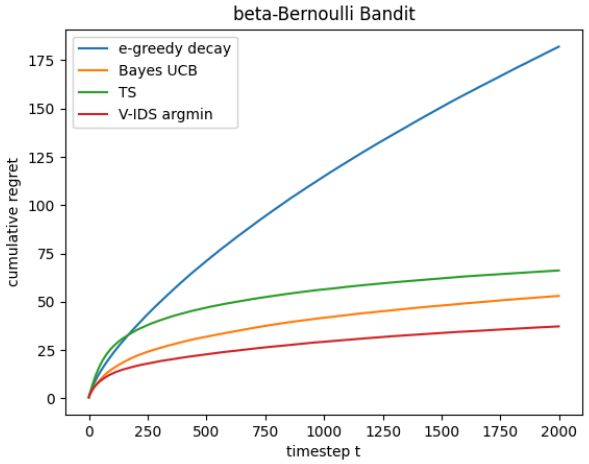

Fig. 3: Comparison of algorithms in the independent Bernoulli setting.

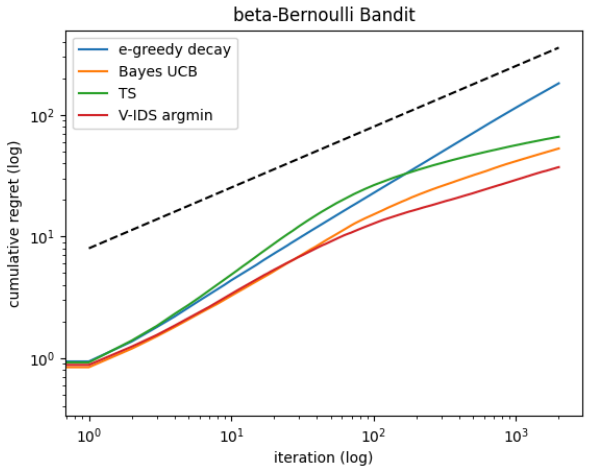

Fig. 4: Log scale view of Fig. 3.

B. *Independent Gaussian*

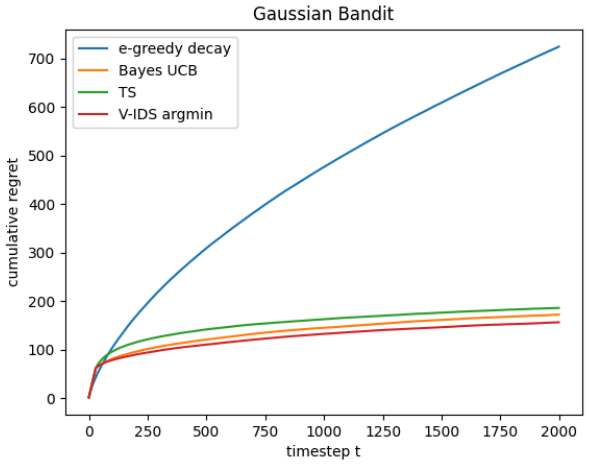

Fig. 5: Comparison in the independent Gaussian setting.

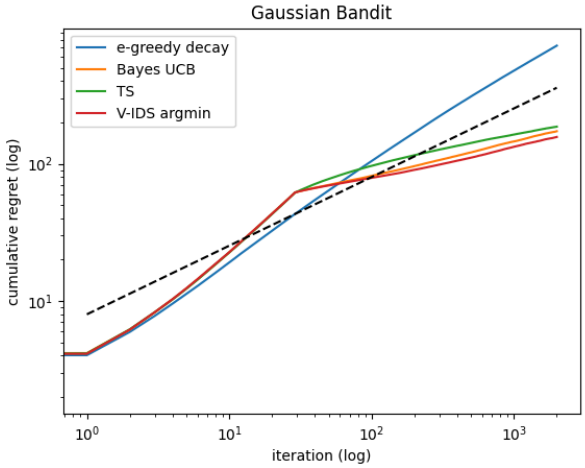

Fig. 6: Log scale view of Fig. 5. Note the exaggerated change in slope due to an initial pass over all unchosen actions.

C. Independent Gamma-Poisson

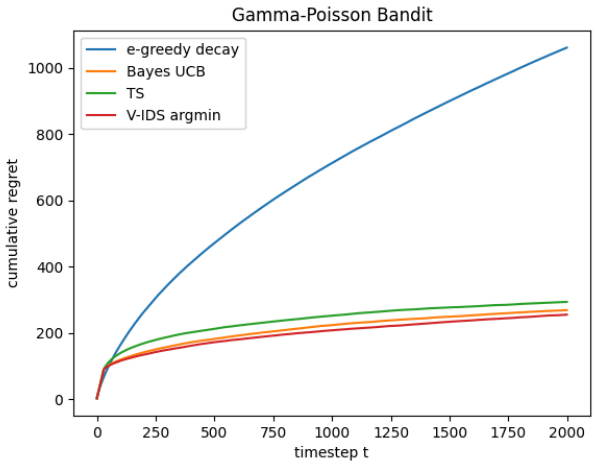

Fig. 7: Comparison in the independent Poisson setting. Note that compared to other settings, the regret observed can be higher even with $\lambda = 1$.

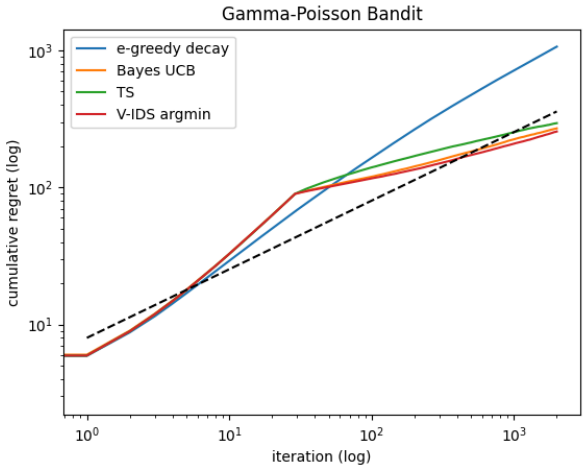

Fig. 8: Log scale view of Fig. 7.

D. Linear Gaussian

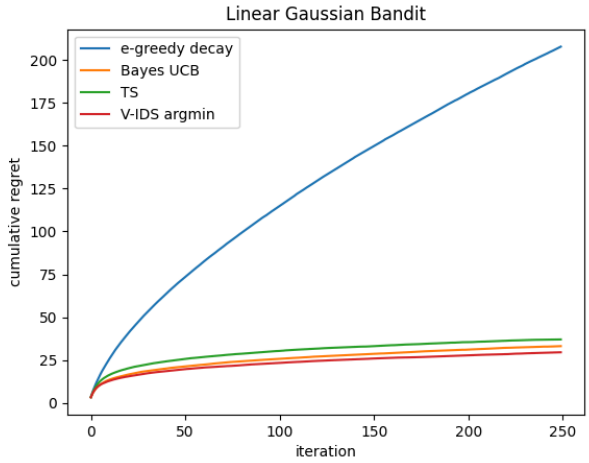

Fig. 9: Comparison in the linear bandit setting.

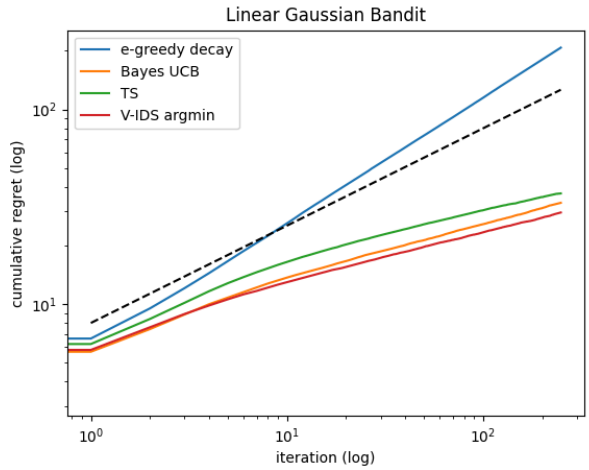

Fig. 10: Log scale view of Fig. 9.

VI. Error Analysis

A. Note on computation time

The linear bandit setting was much more compuationally demanding, given the introduction of a dimension d, and the need to invert matrices during the posterior update.

It is especially demanding for V-IDS implemented with MCMC sampling, since there is a need to generate M samples from a multivariate gaussian distribution.

For this reason, the time horizon for calculation was limited to just 250 steps. Even with this limitation, it seems to be the case that V-IDS is marginally better than Thompson sampling and Bayes UCB.

Compared to the original paper [1], For the computationally heavy setting with $K = 100$ and $d = 30$, I was able to acheive 0.015 seconds per decision for V-IDS, and around 0.00005 seconds (about 50 µs) per decision for Thompson sampling.

For the other settings, I was able to run simulations in parallel on 16 cores (MacBook Pro M4 Max), completing full sets of simulations in under 10 minutes per run. However, for linear bandits, I ran into issues with parallelization and was unable to debug it in time.

B. V-IDS v.s. V-IDS with argmin

In the work introducing IDS [1], it was mentioned that any algorithm that has "nearly" minimal information ratio still satisfies strong regret bounds.

During implementation of V-IDS, I noticed that when optimizing for a policy that produces a probability distribution between two actions, that it was often the case that one action was assigned much higher probability. I changed the action selection mechanism to use a simple argmin over information ratio, and saw that the behavior between V-IDS and V-IDS with argmin was nearly the same.

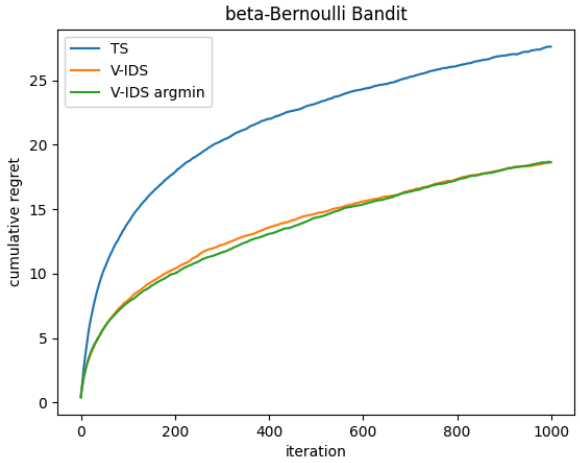

Fig. 11: V-IDS using typical IDSAction selection compared against V-IDS using a simple minimization over information ratios for each arm.

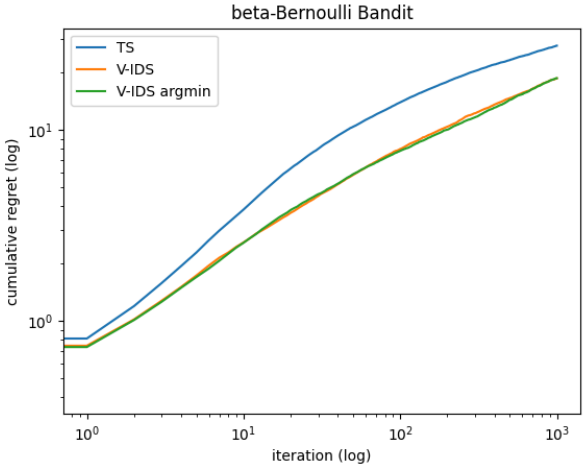

Fig. 12: Log scale of Fig. 11.

This suggests that at least for the beta-Bernoulli setting in simulation, the performance of the two policies are nearly identical, while the argmin operation takes a fraction of the time that the full IDSAction optimization would take.

C. scipy v.s. CVXPY

When selecting an optimization library for IDSAction, I originally intended on using CVXPY. However, given my inexperience with convex optimization, I kept on running into issues formulating the problem correctly. I ended up using scipy.optimize.minimize_scalar instead.

Online discussion suggests that CVXPY may be more efficient in some cases, and for that reason, I believe that my implementation of IDSAction was not the most efficient.

In the future, I may attempt to implement my own optimization algorithm here, given the relatively simple formulation and solutions.

VII. Future Work

There is much work to do, and I am excited to continue my exploration and development over the summer!

A. Codebase

Given the time constraints, the codebase itself is in a dismal state, and is not up to my standards. There are a number of TODOs that I wish to improve upon for the code organization, as well as other improvements such as typing, command-line arguments, result and figure saving, etc.

B. Further implementation

As mentioned in the error analysis section, I wish to implement optimization on my own, as well as implement it using CVXPY in order to compare performance in the IDSAction step. I also wish to further improve my implementation of my parallelization, in order to continually improve upon the simulation performance.

C. Theoretical exploration

Personally, I wish to fully immerse myself in relavent bandit literature over the summer. This includes reviewing notes on probability theory, statistics, information theory, bandit textbooks, as well as important papers. For example, one paper I could take a closer look at is "Aligning AI Agents via Information-Directed Sampling" from Jeon and Van Roy [11], where there is a noticable gap between the proven theoretical results and empirical simulation results.

VIII. Ethical Considerations

Bandits present an ethical and societal risk given their tendency to pillage and loot innocent villagers and passerbys, often through violent means. As seen in popular media, a bandit lives a life of crime, similar to that of a pirate [12].

On the other hand, the ethical and societal risks of multi-armed bandits are not so obvious. One example from my personal experience is with the relationship of MABs and related algorithms to those of recommendation and ranking systems present in the industry. Prior to Stanford, I spent time on the Facebook notifications team as well as the Facebook feed team, where I firsthand witnessed the poten-

tial of these algorithms for maximizing user engagement. In this case, maximizing reward was analgous to encouraging certain behaviors that increased topline metrics, such as sending certain notifications at specific times in order to get users to enter the app. However, this one-dimensional view of "reward" leaves out many potentially harmful side-effects. Some prominent issues that can be exacerbated by this sort of optimization in the notification space are: social media addiction, notification quality degradation, notification "blindness", loss of user trust, etc.

Potential mitigation strategies for these issues can involve modifying the objective of the algorithms to explicitly include metrics that measure the above risks.

Other strategies can involve setting up other algorithms or systems to constrain and double-check the actions output by the algorithms in question.

Finally, another potentially effective measure would be policy changes or education on the industry itself, whether through widespread legislation or through shifts in individual company culture, to prioritize user experience and wellbeing over pure metric shifts.

IX. Code and Project Summary Video

Code is linked on my github (github.com/DavidJGChen/bandit-exploration), and video is linked on my YouTube (youtu.be/0YxQ1Wj4P_I).

Acknowledgements

I would like to thank Hong Jun Jeon for encouraging me to explore a bandit related project, even when I wasn't confident in myself. His detailed feedback as I worked my way through papers was immensely useful, and I hope to work on my own theoretical work one day.

References

[1] D. Russo and B. Van Roy, "Learning to Optimize via Information-Directed Sampling," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Curran Associates, Inc., 2014, p. . [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/90720a2fcc41f9332e6a1558da327089-Paper.pdf

[2] H. Robbins, "Some aspects of the sequential design of experiments," 1952.

[3] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985, doi: https://doi.org/10.1016/0196-8858(85)90002-8.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem," *Mach. Learn.*, vol. 47, no. 2–3, pp. 235–256, May 2002, doi: 10.1023/A:1013689704352.

[5] E. Kaufmann, O. Cappe, and A. Garivier, "On Bayesian Upper Confidence Bounds for Bandit Problems," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, N. D. Lawrence and M. Girolami, Eds., in Proceedings of Machine Learning Research, vol. 22. La Palma, Canary Islands: PMLR, 2012, pp. 592–600. [Online]. Available: https://proceedings.mlr.press/v22/kaufmann12.html

[6] W. R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933, Accessed: May 22, 2025. [Online]. Available: http://www.jstor.org/stable/2332286

[7] O. Chapelle and L. Li, "An Empirical Evaluation of Thompson Sampling," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., Curran Associates, Inc., 2011, p. . [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf

[8] D. Russo and B. V. Roy, "An Information-Theoretic Analysis of Thompson Sampling," *Journal of Machine Learning Research*, vol. 17, no. 68, pp. 1–30, 2016, [Online]. Available: http://jmlr.org/papers/v17/14-087.html

[9] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020. [Online]. Available: https://books.google.com/books?id=bydXzAEACAAJ

[10] A. Slivkins, "Introduction to Multi-Armed Bandits." [Online]. Available: https://arxiv.org/abs/1904.07272

[11] H. J. Jeon and B. V. Roy, "Aligning AI Agents via Information-Directed Sampling." [Online]. Available: https://arxiv.org/abs/2410.14807

[12] E. Oda and M. Sakai, "One Piece - Luffy's Past! The Red-Haired Shanks Appears." Toei Animation, [Television broadcast], 1999.