Submitted January 1, 2020; Published ??/??

Contextual Bandits in Highly Dynamic Environments

Taylor W. Killian TWKILLIAN@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
Vector Institute
Toronto, ON M5S 1A1, CANADA

Editor:

Abstract

Real world applications of contextual bandits are rarely placed in fixed or stationary environments. After a time, specified actions may lose their expected return as rewards shift in response to the executed action. This paper introduces a highly dynamic environment for contextual bandits where the reward probability of an arm decays after it is selected. The reward probabilities eventually recover their initial value after not being selected. Preliminary empirical evaluation of this environment demonstrates, with different decay and recovery rates, that Thompson sampling performs well.

Keywords: Contextual Bandits, Reward Decay, Non-stationarity, Thompson Sampling

1. Introduction and Background

Data driven decision making and computational science has transformed industries and has spurred extensive growth across many scientific disciplines. Core to this technological progress is a selection process between a set of identifiers or actions that an algorithm may suggest when provided some observation or data point. These types of problems have been addressed with a multitude of statistical techniques with many promising advances made from the area of Machine Learning (ML) (Bishop, 2006) where flexible models are trained and adapted based on only the data they are provided. Reinforcement Learning (Sutton and Barto, 2018), a subfield of ML, addresses situations where such observations of the data and subsequent decisions are made in a sequential manner from operating within a specified environment.

The Bandit Learning Framework Reinforcement Learning (RL) allows for the training of algorithms to act and learn from iterative interactions within an environment, defined by the observations made available to the algorithm as well as the objective the algorithm is trained to accomplish. The simplest RL problem is the multi-armed bandit (MAB) (Whittle, 1980; Berry and Fristedt, 1985) where a set of options or ”arms”, each with a fixed payout probability, is provided to the algorithm. The objective is to determine which arm provides highest likelihood of receiving a payout and maximize the return by only choosing that arm. A solution to a MAB problem uses the observed returns from selecting among the available arms to identify the optimal choice. This is a flexible problem framework that has been used to represent a broad variety of decision making problems (Bouneffouf and Rish, 2019).

©Fall 2019 Taylor W. Killian.

Despite the flexibility of the MAB framework, it is limited by the assumption that the nature of each arm remains constant despite underlying characteristics unique to each time an algorithm may be presented with the set of arms. Contextual information about a user or the circumstances surrounding the expected decision may adjust the utility of each arm and which one may be considered optimal. This context can be used by the algorithm to accelerate and guide the selection of the new optimal arm. This setting of the MAB problem has been designated as a Contextual bandit and has been motivated by the promise of personalizing algorithmic decision selection. Algorithms for contextual bandits have been developed for a variety of settings (Auer et al., 2002; Langford and Zhang, 2007; Agrawal and Goyal, 2013; Agarwal et al., 2014; Chapelle and Li, 2011; Chu et al., 2011; Russo et al., 2018). However most of these algorithms rely on the underlying reward probabilities to remain stationary.

Non-stationary environments This assumption is potentially problematic since the utilities underlying decisions in the real world are non-stationary. News articles to be recommended to readers are constantly produced and updated, not to mention that once an article has been read, it is unlikely that the user will want to read it again (Li et al., 2010; Luo et al., 2017). Displayed ads lose their effectiveness over time and, in time, may produce the opposite effect than they were originally designed for as the viewer may tire of observing the same ad again and again (Cao and Sun, 2019; Moriwaki et al., 2019). Even in more critical use cases such as healthcare (Bulucu, 2019; Lei et al., 2017; Varatharajah et al., 2018), the optimal treatment for a patient will not remain fixed as their health improves or degrades. Further, once optimal treatment decisions may become sub-optimal as the patient's body grows resistant to prescribed treatments (Housman et al., 2014; Schmidt and Löscher, 2005).

There is a burgeoning interest among the research community to develop extensions to popular algorithms for bandit learning problems to non-stationary environments. There is growing literature addressing environments where the payout of a MAB may change with time (Chakrabarti et al., 2009; Besbes et al., 2014; Komiyama and Qin, 2014), as a response to repeated use of any one arm (Levine et al., 2017; Seznec et al., 2018; Pike-Burke and Grunewalder, 2019; Basu et al., 2019) or as a way to model second-order effects facing the decision-making problem (Jedor et al., 2019; Russac et al., 2019). However, there are relatively few studies that utilize context to adapt within the non-stationary environment (Moriwaki et al., 2019). Another limitation of these proposed extensions of the MAB problem to non-stationary environments is that the environments change at discrete times and possibly only once over the course of their experiments.

The objective of this paper is to introduce a highly dynamic learning environment for contextual bandits where the underlying reward probabilities for each arm may change after each round. The environment is set up and presented in Section 2, including how context is derived and presented to the learning algorithm. A preliminary empirical investigation is set up in Section 3 with results in Section 4. This paper is concluded in Section 5 with lessons learned and outlines areas of future work. Finally, an additional experimental environment is presented in the Appendix.

2. A Highly Dynamic Bandit Environment

A core limitation to the the algorithms developed for non-stationary environments presented in Section 1 is that there is limited dynamism in how the underlying reward probabilities change. Given the strong theoretical components of these papers, perhaps this was necessary to bear out the performance and computation bounds they present. I was often disappointed how far this may be from how real non-stationary environments change in time where there may be stochastic and continual changes to the environment.

My primary motivation in considering non-stationarity in contextual bandit learning problems is centered on treating human patients in critical care settings where personalized decisions must account for both degrading and improving health, not to mention complex interactions between prior treatments and current observed patient state. The best choice of treatment yesterday, or even an hour ago in some settings, may now be among the worst. In this setting, the changes to the underlying utility of each possible action is affected by the previous action.

To provide an abstracted version of this healthcare scenario, I am happy to present a highly dynamic contextual bandit environment developed in python and will be made publically available [1] where an arm of the bandit may be thought of as a separate treatment and the underlying reward probability is the likelihood of a positive outcome.

Reward Non-Stationarity Each arm k is defined by a reward probability p_k where, at any time t,

$$P(r_k(x_t) = 1) = p_k.$$

Here $r_k(x_t)$ is the true reward when selecting arm k and x_t corresponds to the context facing the selection of an arm at time t. In the non-stationary setting, our reward probabilities may also be time-dependent. Central to this environment is the rate at which rewards decay and recover after an arm has or has not been chosen by the algorithm. That is, at time $t+1$, the reward probability $p_{k,t+1}$ has the form

$$p_{k,t+1} = \begin{cases} \max\left(\ (1-\delta_{decay}) * p_{k,t}\ ,\ 0\right), & \text{if } k = \hat{\pi}(x_t) \\ \min\left(\ (1+\delta_{recovery}) * p_{k,t}\ ,\ p_{k,0}\right), & \text{if } k \neq \hat{\pi}(x_t) \end{cases} \tag{1}$$

where δ_{decay} and $\delta_{recovery}$ are the decay and recovery rates respectively, $p_{k,t}$ was the reward probability of arm k at the previous time step, $p_{k,0}$ corresponds to the base reward probabilities of each arm and $\hat{\pi}(x_t)$ corresponds to the action selection policy when provided context x_t.

Arm selection Rewards are assumed to be linearly associated with the provided context. That is, it is assumed that there exists some underlying parametrization β of the reward function $r_k(x_t)$ such that

$$r_k(x_t) = \beta_{k,0} + \beta_{k,1}^{\intercal} x_t + \epsilon = \beta_k^{\intercal}\,[1,\ x_t] + \epsilon$$

for each arm k, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Thus, in the bandit setting, the action selection policy can be formulated as follows,

[1] Code available at `https://github.com/twkillian/nonstationary_contextual_bandits`

$$\hat{\pi}(x_t) = \arg\max_k \ \sigma(r_k(x_t))$$

where $\sigma(\cdot)$ corresponds to the sigmoid function.

To learn $\hat{\pi}$, the algorithm needs to infer the expected rewards from each arm by learning a regression between observed contexts and the rewards received. That is, for each arm k there will be a set of parameters $\hat{\beta}$ to be learned. To enable the use of Thompson Sampling, Bayesian Linear Regression was used to learn these parameters using data collected from each arm independently. This means that the inferred rewards for each arm follow the distribution:

$$\hat{r}_k \sim \mathcal{N}(\hat{\beta}_k^{\intercal} x_t, \Sigma_k). \tag{2}$$

Context The context (x_t) supplied to the action selection policy is made up of two components. The first component constitutes the time since each arm was selected: $(t - j_k)$ where j_k is the time arm k was last selected. The second component, inspired by Moriwaki et al. (2019), constitutes the frequency each arm has been selected with over a sliding window: $\mathcal{F}_k$. Thus, in this setting, the context contains some history that may be useful for the policy in selecting the best arm to play at the current time.

$$x_{t+1} = [\{t - j_k\}_k, \{\mathcal{F}_k\}_k]$$

The thinking behind this definition of the context is that the first component provides some intuition about the immediate past where the second component provides some longer-term information. The design of the second component is that even if a particular arm hasn't been selected recently, it may have been selected at a high frequency over the relevant past (meaning that after a set number of time steps, one may expect the reward probability to have fully recovered) which may indicate that the reward probability of that arm is lower than expected. This second component is particularly important if the decay and recovery rates are not aligned, especially if the decay rate is higher than the recovery rate.

3. Environment and Experimental Setup

Initializing the environment requires specifying the number of arms (n_{arms}), the decay and recovery rates ($\delta_{decay}, \delta_{recovery}$) as well as initial reward probabilities for each arm ($p_{k,0}$). For simplicity, I specified a mean reward probability $\bar{p}_{k,0}$ and sampled k independent values for the initial probabilities, ensuring that each was a valid probability (i.e. within $[0, 1]$):

$$p_{k,0} \sim \mathcal{N}(\bar{p}_{k,0}, \sigma_p^2)$$

To seed the regressions for learning the $\hat{\beta}_k$, a batch of data was collected with H_{batch} data points where arm selection was done randomly. After batch generation, the separate arm's $\hat{\beta}_k$ are initially trained and the context is reset. At each subsequent timestep, the received context, the arm chosen and the resulting reward is added to this batch for re-fitting the regressions online at a specified frequency, f_{refit}.

Experiments are run for a specified number of iterations (n_{iters}) where the learned action selection policy $\hat{\pi}$ is provided the current context x_t and the selected arm k is assumed to always be pulled with reward received with probability $p_{k,t}$. To aid in action selection and exploration, Thompson sampling was used. Thompson sampling is enabled through the randomness present in how the rewards are inferred using Bayesian Linear Regression. Psuedocode outlining how the environment is used in the experiments is given in Algorithm 1.

```
Input: $n_{arms}, \delta_{decay}, \delta_{recovery}, \{p_{k,0}\}_k, n_{iters}, f_{refit}, H_{batch}$
begin initialization
    randomly generate data;
    fit $\hat{r}_k$ for each arm;
end
for $n_{iters}$ do
    observe $x_t$;
    choose arm $k \sim \hat{\pi}(x_t)$;
    receive reward $r_t = \hat{r}_k(x_t)$;
    store $x_t, k,$ and $r_t$;
    update $p_{k,t}$ for each arm according to Eqt. 1;
    if $t \mod f_{refit} == 0$ then
        refit $\hat{r}_k$ for each arm;
    end
end
```

Algorithm 1: Psuedocode for contextual bandit experiments in a highly dynamic environment

4. Experimental Results

Using the approach described in the previous section and with Algorithm 1, this section will provide experimental details and present results of those experiments. To preliminarily simplify experimentation with decay and recovery rates I chose to evaluate two settings of each rate, one "slow" and one "fast". This corresponded to two experiments where the rates were equal and two where they were unequal. This was designed to test whether Thompson sampling could associate the second component $\mathcal{F}_k$ of the provided context when there were different rates of decay and recovery.

All experiments were run for 2500 iterations, with 3 arms to choose between, a "fast" rate equal to 0.25 and a "slow" rate equal to 0.05. The reward center for initializing $p_{k,0}$ was set to 0.65 with σ_p^2 set to 0.1. Inference for the separate arm's reward models was done via MCMC and NUTS over the specified model in Equation 2. MCMC was run for 1500 iterations with a single chain. As a final note, the sliding window used for the second component $\mathcal{F}_k$ was set to 50. The initial training batch was comprised of 300 examples. All experiments are run comparing Thompson Sampling with Random Sampling. The performance of each approach was evaluated using a notion of regret where the different sampling algorithms were compared to an oracle that selects the current best reward probability. That is,

$$\pi^{\text{oracle}}(x_t) = arg \max_k \{p_{k,t}\}_k$$

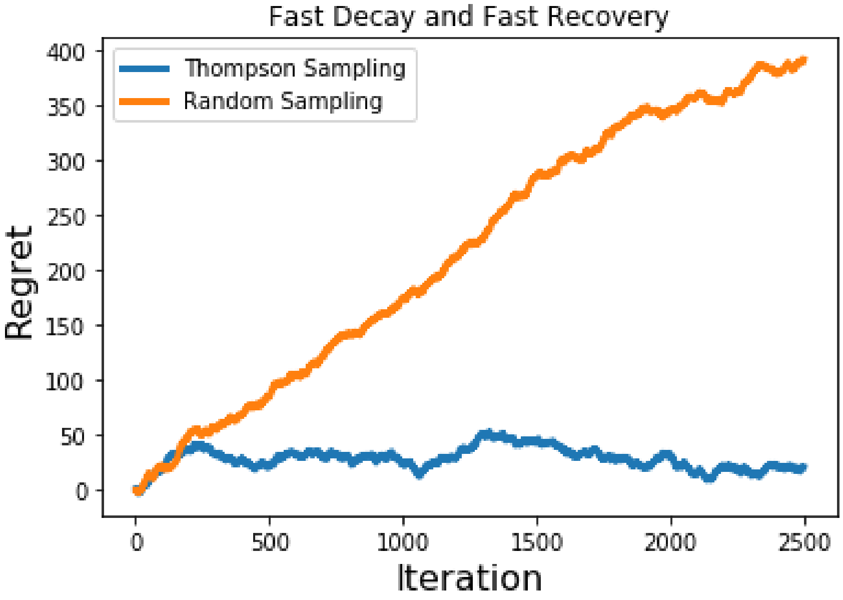

Figure 1: Comparison between Thompson and Random Sampling when there is a "fast" rate for both decay and recovery

Thus, regret can be defined as

$$R = \sum_{i}^{n_{iters}} r_{k^o}(x_i) - r_{k^{'}}(x_i),$$

where k^o corresponds to the arm selected by the oracle and $k^{'}$ corresponds to the arm selected by $\hat{\pi}$ when provide the context x_t. The goal is to minimize regret over the duration of the experiment. Now, there are some peculiarities with this definition of regret as well as how rewards are realized. There are instances over the course of an experiment where the arm chosen by $\hat{\pi}$ provides reward when that chosen by the oracle does not. This results in"negative" regret where the learned action selection policy performs "better" than the oracle. While this is certainly possible through the way I've defined the regret, in practice it seldomly occurs. In the best case (as is shown in Fig. 1) the cumulative regret appears to oscillate as the learned action selection policy often matches the oracle's suggested arm but due to the nature of how the reward is administered, it may or may not be reflected in how the regret is measured.

Experimental results from the four settings of the decay and recovery rates are presented in Figures 1-2. Aside from some odd behavior when the decay of reward probabilities far outpaces the recovery rate (see Fig. 2, something that will be discussed below), the experiments demonstrate that naive Thompson Sampling performs better than Random Sampling. This shouldn't be surprising but these presented results represent a nice stepping off point for more sophisticated algorithm development that can seek to reliably improve beyond

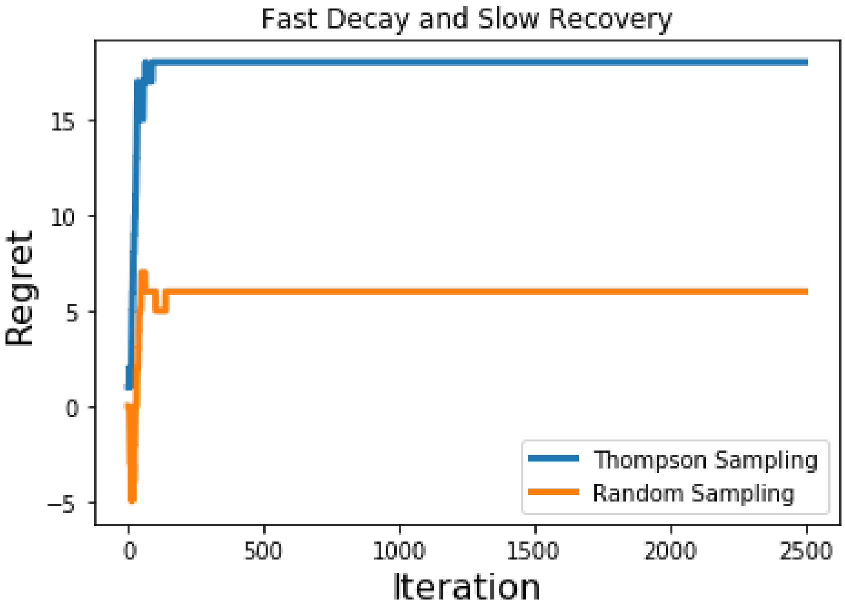

Figure 2: Comparison between Thompson and Random Sampling when there is a "fast" rate for decay and but "slow" recovery rate

Thompson sampling, especially in settings where there is a slow recovery rate (Figures 2 and 4).

In settings where there is a "fast" recovery rate (Figs. 1 and 3), the effect of a decaying reward probability is quickly mitigated if that arm is not immediately chosen again. In these settings, an optimal selection strategy would be to alternate between the top two arms, taking advantage of the recovery rate returning a decayed reward probability close to its initial setting. The "fast" recovery rate also covers for occasional sub-optimal actions where an optimal arm may be played too frequently. In the times where this arm is not played, it's reward probability recovers quicker than the sustained effects of playing the arm too frequently.

However, when there is a "fast" decay rate, sub-optimal action selection can destabilize the environment such that it becomes difficult to recover suitable reward probabilities for any arm. This is no more apparent than when the environment has a "slow" recovery rate (Fig. 2). In this setting, the aggressive decay causes both the arm selection of both the oracle and $\hat{\pi}$ to choose arms that have a low likelihood of providing any reward. This type of outcome is exacerbated when there are a small number of arms to select from as all arms are quickly driven to have small reward probabilities.

Additional considerations that were striking in their affect on the outcome was how tightly interwoven Thompson sampling was with the quality of the reward models $\hat{r}_k$ trained for each arm. Early on in my development for this project, the regressions would over fit to the initially drawn training batch and would become fixed in the reward estimates they would provide no matter the context. I found that running MCMC for fewer samples as well

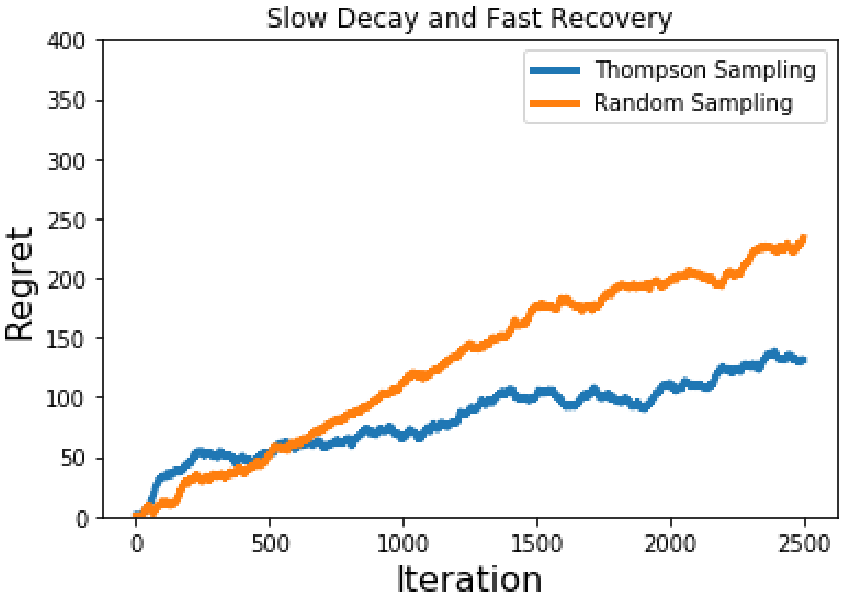

Figure 3: Comparison between Thompson and Random Sampling when there is a "slow" rate for decay and but "fast" recovery rate

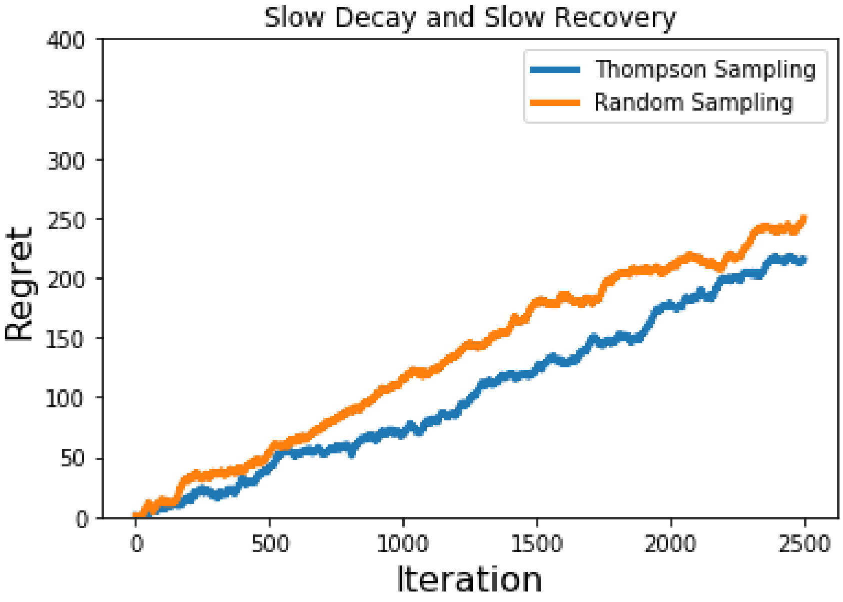

Figure 4: Comparison between Thompson and Random Sampling when there is a "slow" rate for both decay and recovery

as initializing the regression parameters with higher variance were the primary solutions to over fitting and deterministic behavior from the inferred reward models. Another side of this coin is that some arms may not be sampled often enough to provide sufficient data to reliably train a model on. This is a concern when considering scaling to experiments with more arms which may require a larger initial training batch.

5. Conclusion and Future Work

This project report presents a new, highly dynamic, contextual bandit environment for non-stationary settings. Preliminary research shows that a naive application of Thompson sampling in such an environment performs well yet leaves significant room for algorithmic improvement as there are considerable gains to be made in settings where there is a "slow" recovery rate of the reward probabilities that have decayed as each arm is selected.

There are additional directions of future work that I believe could add additional texture to this environment. One direction would be to include a sense of user personalization where different users have individual decay and recovery rates. The context would need to include some form of identifying the user type and their intrinsic rates. Along a similar line of thinking would be to assign different decay and recovery rates to each arm. Another aspect of future work would be to re-think the immediacy of decay and recovery of each arm's reward probabilities. Perhaps different delays between arm selection and decay, or requiring multiple timesteps after an arm has been selected before beginning its recovery toward its initial setting.

An important set of lines of future work is to complete the empirical analysis of this environment by running experiments with more arms and a broader and more varied array of decay and recovery rates. Another aspect of improving the presented experiments would be in reporting the mean and standard deviation over multiple runs of each experiment to show how reliable the presented algorithmic performance.

Ultimately, the motivation behind this project was to investigate contextual bandit settings that were closer to real world settings. In particular, settings that I may encounter in my research while collaborating with medical researchers affiliated with area hospitals. While I do not feel as though the environment I've presented is perfectly representative of any real world, I do feel that it is a step closer. I'm proud of this and am excited to incorporate what I've learned into my future research.

Acknowledgments

It goes without saying that Prof. Joseph J. Williams is deserving of my gratitude in the formation and follow through of this project. He was encouraging and supportive when there appeared to be prior work that had already done what I originally set out to accomplish. I would also like to thank Prof. Audrey Durand who helped to shape this project and offered her expertise in modeling with contextual bandits to help poke holes in and shore up the technical foundation of what I set out to accomplish with this project. Finally, I would be remiss to not acknowledge my classmates Carson McLean, Arnold Yeung and Marina Tawfik who were great to bounce ideas off and receive support from.

Appendix A: An additional direction...

In this appendix, I want to briefly outline another environment that I initially built as part of my project (and is included in the accompanying github repo[2]). I was fixated with the idea that contextual bandit problems had to incorporate user preferences or features. As such, I was struggling to find a way to account for users in a simpler version of the environment presented in Section 2 without specifying a separate reward function for each user. What I resorted to, after discussing this project with Audrey Durand–professor at Laval University, contextual bandit expert, and friend–was a resource management problem where the goal is to provide a user with a preferred item while keeping in mind how much of that item there is.

A.1: An Environment for Resource Management

The resource management setting that I had designed concerned a collection of items that one would like to present to users to purchase. Each user had a preference over the items that corresponded to the probability they were to purchase each item if it were presented to them. The objective was for the algorithm to determine which item to present to the user while also keeping in mind whether there were any of that item remaining to be sold. Ideally, if a user's most preferred item was out of stock the algorithm would suggest items that were the next most preferred in their intrinsic ranking.

User Context One of the biggest challenges with this environment was the apparent non-linear association between user preference, resource availability and eventual reward. It was unclear how to construct a context that provided enough information, without being fully transparent, that some algorithm would be able to interpret correctly. At first, I stratified the interval between 0 and 1 into bins based on the user's most preferred item. Within that bin I sampled a numerical value centered on the midpoint of the bin. The rest of the initial context was a binary identifier whether there were any available items of each type.

Bayesian Neural Networks With the apparent non-linear relationship between context and reward, I determined to use a Bayesian Neural Network (BNN) as an inferred reward function for each arm. Having worked with BNNs previously, I should've been more cautious as they are extremely temperamental and require extensive hyperparameter search to get to work properly. After some time of trying (and failing) to get a very simple BNN to not overfit or otherwise revert to deterministic behavior (where a single arm was defaulted to no matter the context), I decided to abandon this environment and punt it ahead for potentially another area for future work.

[2] `https://github.com/twkillian/nonstationary_contextual_bandits`

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. In *Advances in Neural Information Processing Systems*, pages 4785–4794, 2019.

Donald A Berry and Bert Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5:71–87, 1985.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pages 199–207, 2014.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Djallel Bouneffouf and Irina Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*, 2019.

Cem Bulucu. *Personalizing treatments via contextual multi-armed bandits by identifying relevance*. PhD thesis, Bilkent University, 2019.

Junyu Cao and Wei Sun. Dynamic learning of sequential choice bandit problem under marketing fatigue. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Advances in neural information processing systems*, pages 273–280, 2009.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

Genevieve Housman, Shannon Byler, Sarah Heerboth, Karolina Lapinska, Mckenna Longacre, Nicole Snyder, and Sibaji Sarkar. Drug resistance in cancer: an overview. *Cancers*, 6(3):1769–1792, 2014.

Matthieu Jedor, Vianney Perchet, and Jonathan Louedec. Categorized bandits. In *Advances in Neural Information Processing Systems*, pages 14399–14409, 2019.

Junpei Komiyama and Tao Qin. Time-decaying bandits for non-stationary systems. In *International Conference on Web and Internet Economics*, pages 460–466. Springer, 2014.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 817–824. Citeseer, 2007.

Huitian Lei, Ambuj Tewari, and Susan A Murphy. An actor-critic contextual bandit algorithm for personalized mobile health interventions. *arXiv preprint arXiv:1706.09090*, 2017.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, pages 3074–3083, 2017.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. *arXiv preprint arXiv:1708.01799*, 2017.

Daisuke Moriwaki, Komei Fujita, Shota Yasui, and Takahiro Hoshino. A contextual bandit algorithm for ad creative under ad fatigue. *arXiv preprint arXiv:1908.08936*, 2019.

Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. In *Advances in Neural Information Processing Systems*, pages 14122–14131, 2019.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026, 2019.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Dieter Schmidt and Wolfgang Löscher. Drug resistance in epilepsy: putative neurobiologic and clinical mechanisms. *Epilepsia*, 46(6):858–877, 2005.

Julien Seznec, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko. Rotting bandits are no harder than stochastic ones. *arXiv preprint arXiv:1811.11043*, 2018.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

Yogatheesan Varatharajah, Brent Berry, Sanmi Koyejo, and Ravishankar Iyer. A contextual-bandit-based approach for informed decision-making in clinical trials. *arXiv preprint arXiv:1809.00258*, 2018.

Peter Whittle. Multi-armed bandits and the gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):143–149, 1980.