A Note on the Advantage of Context in Thompson Sampling

Michael Byrd and Ross Darrow

July 28, 2020

Abstract

Personalization has become a focal point of modern revenue management. However, it is often the case that minimal data is available to appropriately make suggestions tailored to each customer. This has led to many products making use of reinforcement learning based algorithms to explore sets of offerings to find the best suggestions to improve conversion and revenue. Arguably the most popular of these algorithms are built on the foundation of the multi-arm bandit framework, which has shown great success across a variety of use cases. A general multi-arm bandit algorithm aims to tradeoff adaptively exploring available, but under observed, recommendations, with the current known best offering. While much success has been achieved with these relatively understandable procedures, much of the airline industry is losing out on better personalized offers by ignoring the context of the transaction, as is the case in the traditional multi-arm bandit setup. Here, we explore a popular exploration heuristic, Thompson sampling, and note implementation details for multi-arm and contextual bandit variants. While the contextual bandit requires greater computational and technical complexity to include contextual features in the decision process, we illustrate the value it brings by the improvement in overall expected

Keywords— bandit algorithms, online learning, e-commerce

1 Introduction

Real-time, model-driven insights become increasingly important for maintaining a competitive edge in the marketplace. Many upcoming applications of revenue management require or will be enhanced by real-time experimentation. Some examples:

1. **Ancillary pricing**. While common to initialize ancillary prices with customer surveys and conjoint analysis (Green, Krieger, & Wind, 2001), after initial prices are set it's often useful to do live experiments to further optimize the prices. Even well-designed experiments may not accurately capture the the true willingness-to-pay of an item, and, worse, can not adapt to changing behaviors over time without further studies.

2. **Personalized offers**. Recently, under mild assumptions, it was estimated that profits could be improved over 10% by using personalized pricing (Dubé & Misra, 2019). For an introduction to offer management for airlines, see (Vinod, Ratliff, & Jayaram, 2018).

3. **Online shopping and booking**. Airlines and online travel agencies want to experiment with different display approaches to improve conversion.

Offline analysis, such as traditional A/B/n testing, has shown to be too slow to adapt to the current rapidly changing marketplace. The main criticisms being that an A/B/n test is expected to run for a predetermined amount of time while also uniformly allocating traffic allocation regardless of recently observed findings. To address these problems, reinforcement learning algorithms have been introduced to quickly learn toward an optimal solution in real-time. Many problems fall into the bandit paradigm, which balances between recommending new offerings with the current known-best offering. The most well-known of these approaches is the multi-arm bandit (Whittle, 1980). The travel industry has begun making extensive use of multi-arm bandit algorithms to address many web-scale problems.

While the multi-arm bandit works well, the traditional algorithms ignore the available contextual data. Incorporating each unique context into business decisions has been shown to help adaptively address similar issues with minimal data. For example, a well-known empirical experiment with the contextual bandit illustrated substantial gains in performance for personalizing news articles (Li, Chu, Langford, & Schapire, 2010). To illustrate the advantage of using context, we explore a multi-arm bandit algorithm employed by many in the travel industry, Thompson sampling, and show how recent advancements extended it to handle context.

1.1 Overview

In Section 2 we present the foundation for bandit algorithms. This includes reviewing Thompson sampling. Section 3 then further inspects a contextual bandit algorithm using Thompson sampling. Here, we review the algorithm and inspect improvements to ease computation. In Section 4 we implement a simulation study to investigate the potential performance improvements for the contextual algorithm over bandit algorithms that ignore the context. Finally Section 5 offers final thoughts and future improvements.

2   Background

We consider the bandit problem with binary reward. For time step $t = 1, 2, \ldots$ suppose a recommendation is to be made from one of K possible options. Upon being received, the recommendation is either selected or not selected; we note that these concepts can easily be adapted to incorporate the profitability into the decision process. The goal is to recommend the option that has the highest chance of success. Suppose, of the K options available to recommend, that each option has p_A features. We refer to each option as an *arm*, where $\boldsymbol{a}_i \in \mathbb{R}^{p_A}$ denotes the set of features associated with arm i. Further suppose each instance has an additional p_I features, denoted $\boldsymbol{w}_t \in \mathbb{R}^{p_I}$. We use $\boldsymbol{x}_{t,i}$ to refer generally to the ith *context* at time t as the collection of features composed of $\boldsymbol{w}_t$ and $\boldsymbol{a}_i$. The contextual features include the arm features, instance features, and any function of the two, such as interactions, and is assumed to be of dimension p.

Let $y_t \in \{0, 1\}$ be the reward from the recommendation at time t. We assume that

$$\mathbb{E}(y_t | \boldsymbol{x}_{t,i}) = f(\boldsymbol{x}_{t,i}; \boldsymbol{\Theta}), \tag{1}$$

where $\boldsymbol{\Theta}$ are learnable parameters. We wish to recommend the arm

$$\alpha_t^* = \underset{i}{\operatorname{argmax}} \left\{ \mathbb{E}(y_t | \boldsymbol{x}_{t,i}) \right\}_{i=1}^K, \tag{2}$$

which corresponds to the arm with the highest expected reward; in the case of a binary reward, (2) corresponds to the arm with the highest probability of success. Let α_t denote the index corresponding to the arm selected at time t and $\boldsymbol{\Theta}_t$ be the parameters learned at time t. Define the cumulative regret at time t by

$$r_t = \sum_{j=1}^t f(x_{j,\alpha_j^*}, \boldsymbol{\Theta}) - f(x_{j,\alpha_j}, \boldsymbol{\Theta}). \tag{3}$$

We assume the parameters $\boldsymbol{\Theta}_t$ are updated as frequently as possible, preferably every iteration, though this may not always be the case in practice (Joulani, Gyorgy, & Szepesvári, 2013).

For instance, a common problem in airline offer management is offering personalized bundles to different travelers. Suppose each traveler arrives sequentially at time $t = 1, 2, \ldots$, where it is desired to offer the bundle which has the highest probability of being purchased. There are K possible bundles, each having various add-ons like offering a checked bag or allowing for refunds on the ticket. Suppose each offering is one-hot encoded to encapsulate if the bundle contains that offering, then the offerings can represent the arm's feature set. The complete context can include an indicator if the traveler is a business or leisure traveler, and all interactions of the bundle's offerings with the traveler type. Naturally, if the bundle was created sequentially by the traveler, each new suggestion to be potentially added into the bundle could incorporate the outcome of prior suggestions.

While tempting to always select the item that is estimated to give the highest expected reward, this could often lead to suboptimal results. Typically, bandit algorithms are used when historical data is not available. Hence, it is reasonable that the optimal solution has yet to have been learned due to having no data to incorporate into the estimated $\mathbf{\Theta}_t$. This has inspired the exploration-exploitation trade-off (Audibert, Munos, & Szepesvári, 2009), which aims to incorporate the uncertainty about undersampled arms when choosing the arm to recommend. While many strategies exists (Chapelle & Li, 2011) (Garivier & Cappé, 2011) (Karnin, Koren, & Somekh, 2013), the general notion is that the more an arm is sampled, the more certain we are about the outcome. Hence, arms we are confident to perform worse than other arms will begin to never be picked with time, ultimately converging to the arm that minimizes the regret per each given instance.

2.1 Thompson Sampling for Bandit Problems

Many exploration policies exist in practice, but we focus on Thompson sampling (Chapelle & Li, 2011) (Agrawal & Goyal, 2012). Thompson sampling puts a prior distribution on $\mathbf{\Theta}$, and then aims to iteratively update the posterior distribution with each newly observed example. Access to the posterior distribution quantifies the uncertainty of the estimate of $\mathbf{\Theta}$ at time t. When making a recommendation, Thompson sampling algorithms generate $\mathbf{\Theta}_t^*$ from the current posterior distribution and use it in calculating the probability of success for each arm. This process incorporates the uncertainty of the estimate into the decision making to allow better exploration. As time progresses, the estimate of $\mathbf{\Theta}$ will become more certain and the posterior distribution will contract around $\mathbf{\Theta}$, which leads to $\mathbf{\Theta}_t^*$ often being close to $\mathbf{\Theta}$. Thompson sampling has been shown to efficiently trade-off exploring the set of available arms with offering arms that more probable to be successful (Kaufmann, Korda, & Munos, 2012). There has been considerable success achieved by Thompson sampling (Ferreira, Simchi-Levi, & Wang, 2018), and it has empirically outperformed other multi-arm bandit approaches (Chapelle & Li, 2011).

Formally, we assume that $\mathbf{\Theta} \sim \pi(\mathbf{\Theta})$ and are interested in iteratively computing

$$\pi(\mathbf{\Theta}|\boldsymbol{X},\boldsymbol{y}) = \frac{\pi(\mathbf{\Theta})\pi(\boldsymbol{X}_t,\boldsymbol{y}_t|\mathbf{\Theta})}{\pi(\boldsymbol{X}_t,\boldsymbol{y}_t)}, \tag{4}$$

where $\boldsymbol{X}_t = (\boldsymbol{x}_{1,\alpha_1},\ldots,\boldsymbol{x}_{t,\alpha_t})^T$ and $\boldsymbol{y}_t = (y_1,\ldots,y_t)^T$ denote the historical contexts for the arms recommended and their outcomes, respectively. The denominator is often referred to the normalizing constant, where

$$\pi(\boldsymbol{X},\boldsymbol{y}) = \int_{\mathbf{\Theta}} \pi(\boldsymbol{X}_t,\boldsymbol{y}_t|\mathbf{\Theta})\pi(\mathbf{\Theta})d\mathbf{\Theta}. \tag{5}$$

Often (5) can be intractable, however $\pi(\mathbf{\Theta})$ and $\pi(\boldsymbol{X}_t,\boldsymbol{y}_t|\mathbf{\Theta})$ can be chosen to be conjugate which gives closed form updates for $\pi(\mathbf{\Theta}|\boldsymbol{X},\boldsymbol{y})$ (Gelman et al., 2013). Upon observing a

new instance at time step $t + 1$, $\boldsymbol{\Theta}^*_{t+1}$ is drawn at random from $\pi(\boldsymbol{\Theta}|\boldsymbol{X}_t, \boldsymbol{y}_t)$ and used to select the arm to be recommended as

$$\alpha_t = \underset{i}{\mathrm{argmax}} \left\{ f(\boldsymbol{x}_{t,i}; \boldsymbol{\Theta}^*_{t+1}) \right\}_{i=1}^K .$$

Finally, the posterior is recomputed with the new recommendation and the outcome as in (4).

2.2 Thompson Sampling for the Multi-Arm Bandit

To first illustrate Thompson sampling, we show it in the classic setting of the multi-arm bandit problem. The multi-arm bandit problem ignores the available context when making a recommendation at each instance. In a sense, the multi-arm bandit is exploring the marginal success rate for each available arm. A Thompson sampler for a multi-arm bandit problem with binary reward puts a prior distribution on the probability of success and incorporates the empirical successes into the posterior update. This can easily be achieved using a beta prior and binomial likelihood, which is conjugate and gives closed form updates for (4). Then, at each iteration, each arm is randomly sampled from its corresponding posterior distribution, and the arm with the highest sample is recommended.

Formally, consider unknown success probabilities for each arm $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_K)^T$, where $\theta_k \sim Beta(a, b)$ for all $k = 1, \ldots, K$. Letting $\pi(y|\theta_k) \sim Bern(\theta_k)$, then, at time t, the posterior distribution for the success of arm k is

$$\pi(\theta_k|\boldsymbol{y}_t) \sim Beta(a + s_{t,k}, b + n_{t,k} - s_{t,k}), \tag{6}$$

where $n_{t,k} = \sum_{i=1}^t \mathbf{1}(\alpha_i = k)$ is the total number of times arm k was recommended at time t and $s_{t,k} = \sum_{i=1}^t \mathbf{1}(\alpha_i = k)y_i$ is the total number of successes for arm k at time t. Then, for the instance at time step $t + 1$, possible arm success probabilities, $\boldsymbol{\Theta}^*_{t+1} = (\theta_1^*, \ldots, \theta_K^*)$, are generated according to (6). The recommended arm is selected by

$$\alpha_t^* = \underset{i}{\mathrm{argmax}} \{\theta_1^*, \ldots, \theta_K^*\}. \tag{7}$$

As time progresses, the more the posterior distribution will reflect the potential success rate for each arm. With more observations, the tighter the *Beta* posterior will contract about the true success probability. Note that each arm's success probability is completely independent from one another. Hence, if contextual information gives insight into multi sets of arms, then the regret could better be minimized by more quickly finding optimal arm traits.

3 Thompson Sampling with Context

To illustrate the benefits of including contextual information, we inspect a contextual bandit algorithm using Thompson sampling. While incorporating additional information

can help reduce the overall regret, it comes at the cost of complexity as conjugacy is not easily achieved for (4). For the sake of complexity, we assume a linear relationship between the context and success probability, such that

$$\mathbb{E}(y_t|\boldsymbol{x}_{t,i}) = h(\boldsymbol{x}_{t,i}^T\boldsymbol{\theta}) \tag{8}$$

for regression coefficients $\boldsymbol{\theta} \in \mathbb{R}^p$. The function h is assumed to be the logit function, $h(x) = 1/(1 + \exp(-x))$. Notably, any model that gives a posterior distribution, like a Bayesian neural network (Riquelme, Tucker, & Snoek, 2018), can be used, but this adds additional complexity to the update process.

With the assumed likelihood $y_t \sim Bern(h(\boldsymbol{x}_{t,i}^T\boldsymbol{\theta}))$, the choice for the prior on the regression coefficients is still needed. This, however, causes problems due to the intractability of (5) for most reasonable priors. Such issues are well known in Bayesian statistics, where heavy use of Gibbs samplers are common place. A recent advance showed a data augmentation technique can facilitate efficient sampling algorithms for (8) when $\pi(\boldsymbol{\theta}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ (Polson, Scott, & Windle, 2013). This technique was adapted by (Dumitrascu, Feng, & Engelhardt, 2018) in the context of Thompson sampling, and illustrated great improvements over other contextual bandit algorithms.

3.1 Pólya-Gamma Augmentation for Binomial Regression

Here we briefly explore the data augmentation technique that facilitates the sampling of the posterior distribution for the linear contextual bandit with binary reward. A random variable z is distributed according to a Pólya-Gamma distribution with parameters $b \in \mathbb{R}^+$ and $c \in \mathbb{R}$ if

$$z = \frac{1}{2\pi^2} \sum_{j=1}^{\infty} \frac{g_k}{(j-1/2)^2 + c^2/(4\pi^2)}, \tag{9}$$

where $g_j \sim Gamma(b, 1)$ for $j \in \mathbb{N}$; if z is a Pólya-Gamma random variable with parameters b and c, then $z \sim PG(b, c)$. A useful identity was identified by (Polson et al., 2013), showing

$$\frac{(e^\psi)^\alpha}{(1 + e^\psi)^b} = 2^{-b}e^{\kappa\psi} \int_0^\infty e^{-z\psi^2/2}p(z)dz \tag{10}$$

for $z \sim PG(b, 0)$. Naturally (10) bears resemblance to the logit function, where we can set $\psi = \boldsymbol{x}_{t,i}^T\boldsymbol{\theta}$, and was adapted to Bayesian MCMC procedures for posterior estimation.

With some manipulation it can be shown that

$$\pi(\boldsymbol{\theta}|\boldsymbol{X}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) \propto \pi(\boldsymbol{\theta}) \prod_{i=1}^{t} \exp(\frac{z_i}{2}(\boldsymbol{x}_{i,\alpha_i} - \kappa_i/z_i)^2) \tag{11}$$

for $\kappa_i = y_i - 1/2$ and $PG(1, 0)$ latent variables $\boldsymbol{z}_t = (z_1, \ldots, z_t)^T$. When $\pi(\boldsymbol{\theta}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, then (11) is proportional to a Normal distribution.

Further, it can be shown that the full-conditional distribution for the latent variables $\boldsymbol{\theta}$ and $\boldsymbol{z}_t$ can be found in closed form

$$\boldsymbol{\theta}|\boldsymbol{X}_t, \boldsymbol{y}_t, \boldsymbol{z}_t \sim N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{12}$$

$$z_i|\boldsymbol{X}_t, \boldsymbol{y}_t, \boldsymbol{\theta} \sim PG(1, \boldsymbol{x}_{i,\alpha_i}^T \boldsymbol{\theta}), \tag{13}$$

where $\boldsymbol{\Sigma}^* = (\boldsymbol{X}_t^T \boldsymbol{Z}_t \boldsymbol{X}_t + \boldsymbol{\Sigma}^{-1})^{-1}$, $\boldsymbol{Z} = diag(\boldsymbol{z}_t)$, $\boldsymbol{\mu}^* = \boldsymbol{\Sigma}^*(\boldsymbol{X}_t \boldsymbol{\kappa} + \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu})$, and $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_t)^T$ By iteratively updating between (12) and (13), the posterior distributions can be estimated with help from quick accept-reject sampling strategies (Makalic & Schmidt, n.d.).

3.2   Pólya-Gamma Augmentation for Thompson Sampling

The Pólya-Gamma distribution augmentation was incorporated into a Thompson sampler by (Dumitrascu et al., 2018) in a similar fashion to the presentation in (Polson et al., 2013) for a Gibbs sampler to estimate binomial regression. Upon receiving each instance, the Pólya-Gamma Thompson sampler (PT-TS) approximates the posterior distribution of the regression coefficients, $\boldsymbol{\theta}$, by drawing M iterative samples of (12) and (13). This allows incorporating the most recent data into the shape of the posterior, and progressively contracts the posterior distribution about the true regression coefficients. The Mth draw of $\boldsymbol{\theta}$ is used as $\boldsymbol{\theta}^*$ and used in calculating (8) for each arm, which then suggests the arm with largest expected success rate.

The guidance in (Dumitrascu et al., 2018) advocates for using $M = 100$, giving a large burn-in time to better traverse the posterior distribution. Having large burn-in times is ideal, but difficult to compute in practice. This is further complicated by the difficult operations in (12) and (13), like matrix inversion, that are computationally difficult and must be performed in each iteration of the burn-in for each instance. The use of $M = 1$ was found by (Dumitrascu et al., 2018) to give competitive results. This result suggests that the algorithm can quickly move toward ideal regions of the posterior.

As t grows, the posterior distribution will shrink toward the $\boldsymbol{\theta}$, giving little mass to be traversed. This, of course, is ideal, as we desired to find the true underlying distribution; however, the incremental value of each data point will decrease as the uncertainty, as measured by $\boldsymbol{\Sigma}^*$, increasingly shrinks in magnitude. We further propose a rolling window over the data used, taking only the $min(L, t)$ most recent observations and outcomes when estimating the posterior at time t. This (1) reduces the amount of computation and (2) allows for gradual adaption to underlying changes in the underlying process. We find in our experiments that, for a reasonable value of L, there is minimal difference in the observed regret between using all data or a rolling window.

4 Simulation

We implement a small simulation study to illustrate the gain in using PG-TS over Thompson sampling in the MAB setting. Each study is conducted assuming a linear relationship between the arm features and the interaction of the instance and arm features as in (8); note we do not include the instance features as they provide no information to the preference of an arm by themselves. For a horizon of $T = 1000$ steps, we observe randomly generated instances and recommend an arm for a random sampler, MAB Thompson sampler, and variants of the PG-TS sampler. Each instance is generated at random, where $\boldsymbol{w}_t \sim N(\mathbf{0}_{p_I}, diag(\mathbf{1}_{p_I}))$ for $t = 1, \ldots, T$. We generate the arm feature values $a_i \sim N(\mathbf{0}_{p_A}, diag(\mathbf{1}_{p_A}))$ for $i = 1, \ldots, K$ and regression coefficients $\boldsymbol{\theta} \sim N(\mathbf{0}_p, diag(\mathbf{1}_p))$ at time $t = 1$, which persist for the entire time horizon. We consider two settings:

1. $K = 10$ arms with 2 instance features and 2 arm features,

2. $K = 100$ arms with 2 instance features and 2 arm features.

With 2 instance features and 2 arm features, then the model has a total of $p = 6$ features. Each setting inspected is replicated 25 times.

The MAB Thompson sampler is set with hyperparameters $a = 1$ and $b = 1$. For the PG-TS prior we assume $\boldsymbol{\theta} \sim N(\mathbf{0}_p, diag(\mathbf{1}_p))$ for each version; we note that, while we assume a somewhat strong prior, the Gibbs sampler is valid under vague priors if unwilling to make such an assumption (Choi, Román, et al., 2017). We inspect PG-TS with a rolling window and without a rolling window, where, for the rolling window, we set $L = 100$. Additionally we inspect the difference in changing the number of Gibbs sampler iterations at each instance, with $M = 25$ and $M = 1$. We permute each of the rolling window and Gibbs sampler iteration settings, giving 4 variants of PG-TS. The performance is measured by the average cumulative regret, as calculated in (3), at each time $t = 1, \ldots, T$ for each method; additionally, we show the 5% and 95% percentiles for the cumulative regret at each time.

We plot the results of Setting 1 and Setting 2 in Figure 1. We opt to display the results for $M = 25$, but note that the $M = 1$ results were about 2.5 worse than the displayed PG-TS results; however, even with $M = 1$, both PG-TS outperformed the MAB Thompson sampler. The performance improvement from incorporating context is quite stark, where both the rolling window and standard PG-TS procedures upper percentile bound is much less than the MAB algorithm. Additionally, while the standard PG-TS does perform better than the rolling window version, the difference is relatively small. The rolling window then seems to reduce the amount of computation at a minimal cost to the overall model.
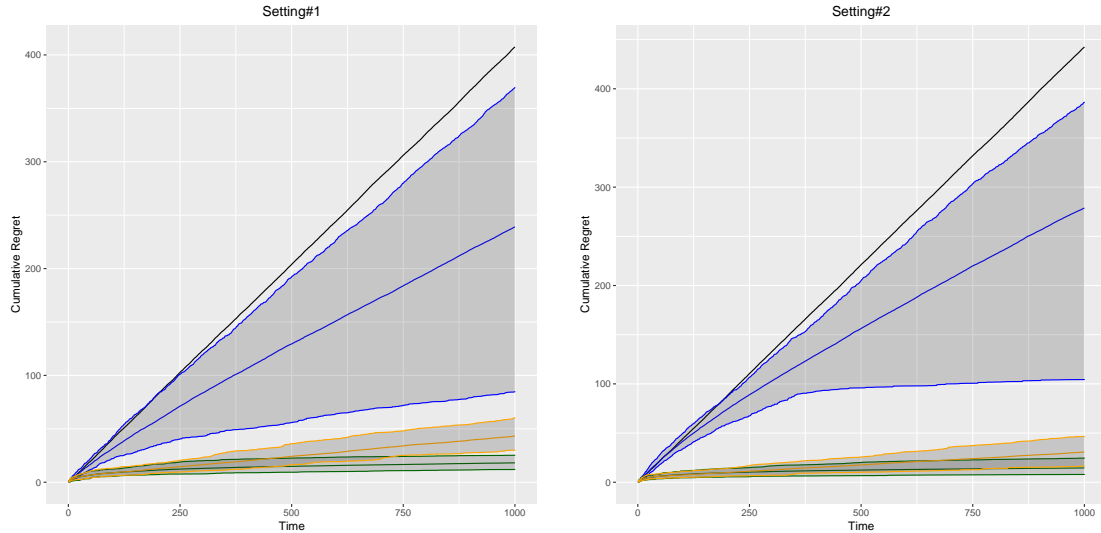
Figure 1: The average cumulative regret with the 5% and 95% percentile bands for the MAB Thompson sampler and PG-TS with and without a rolling window for $M = 25$. Both settings use $p = 6$ total context features with $K = 10$ and $K = 100$ arms.

5   Conclusion

The need for adaptive exploration of online retail continues to grow. While many have addressed this issue with multi-arm bandit algorithms with great success, there is still room for improvement over simply estimating marginal success rates. Here we have explored a recent contextual extension of the well known Thompson sampler, called Pólya-Gamma Thompson Sampling, and empirically illustrated the potential gains in terms of minimizing the regret of the recommendations. While improvements to the regret are realized, it does come at the cost of more computation. These computations can be somewhat mitigated by the rolling window we imposed. Alternatively, computation could be reduced by computing updates in batches, rather than each iteration, or assuming the posterior covariance of the regression coefficients to be diagonal, simplifying the inverse operation. Regardless, as computational platforms continue to illustrate low-cost scalability, the improvement to the recommendation platform seems deserving of consideration for production systems.

References

Agrawal, S., & Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory* (pp. 39–1).

Audibert, J.-Y., Munos, R., & Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, *410*(19), 1876–1902.

Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems* (pp. 2249–2257).

Choi, H. M., Román, J. C., et al. (2017). Analysis of polya-gamma gibbs sampler for bayesian logistic analysis of variance. *Electronic Journal of Statistics*, *11*(1), 326–337.

Dubé, J.-P., & Misra, S. (2019). Personalized pricing and customer welfare. *Chicago Booth School of Business Working Paper*.

Dumitrascu, B., Feng, K., & Engelhardt, B. (2018). Pg-ts: Improved thompson sampling for logistic contextual bandits. In *Advances in neural information processing systems* (pp. 4624–4633).

Ferreira, K. J., Simchi-Levi, D., & Wang, H. (2018). Online network revenue management using thompson sampling. *Operations research*, *66*(6), 1586–1602.

Garivier, A., & Cappé, O. (2011). The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory* (pp. 359–376).

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Green, P. E., Krieger, A. M., & Wind, Y. (2001). Thirty years of conjoint analysis: Reflections and prospects. *Interfaces*, *31*(3_supplement), S56–S73.

Joulani, P., Gyorgy, A., & Szepesvári, C. (2013). Online learning under delayed feedback. In *International conference on machine learning* (pp. 1453–1461).

Karnin, Z., Koren, T., & Somekh, O. (2013). Almost optimal exploration in multi-armed bandits. In *International conference on machine learning* (pp. 1238–1246).

Kaufmann, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory* (pp. 199–213).

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 661–670).

Makalic, E., & Schmidt, D. (n.d.). *High-dimensional Bayesian regularised regression with the bayesreg package*. (arXiv:1611.06649v3)

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, *108*(504), 1339–1349.

Riquelme, C., Tucker, G., & Snoek, J. (2018). Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*.

Vinod, B., Ratliff, R., & Jayaram, V. (2018). An approach to offer management: maximizing sales with fare products and ancillaries. *Journal of Revenue and Pricing Management*, *17*(2), 91–101.

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