Online Multinomial Classification via Thompson Sampling under Multinomial Logit Model

Dinghuai Zhang, Jiaqi Zhang, Feng Zhu
Peking University
{1600013525, 1600010617, 1600010643}@pku.edu.cn

Abstract

Thompson Sampling [21] is a traditional but powerful meta-algorithm framework for online decision problems, where there exists a balance between exploration and exploitation. Under Thompson sampling framework, we develop two kinds of algorithms, approximation methods and Pólya-Gamma sampling methods, to solve online multi-classification problem with multinomial logit assumption. We perform systematically test on both simulated and real datasets to demonstrate the effectiveness of our proposed algorithms.

1 Introduction

In this report, we will investigate Bayesian methods in online multi-classification under multinomial logit model. Different from canonical multi-classification tasks in machine learning or statistical learning community, we consider a setting where data arrives in a sequential manner and need to be predicted *online*. The decision maker need to update his decision methods with accumulated information. The ultimate goal is to maximize the total number of correct predictions.

This setting could be embedded into some practical environment, e.g., online advertisement recommendation system. Each time a customer arrives, the system needs to recommend an advertisement. If the customer clicks on the ad, we get a reward 1, otherwise 0. Then we update our recommendation method. The goal is to maximize the total clicks during a certain period.

In this decision making process, the fundamental difficulty lies in the exploitation-exploration balance: in order to maximize cumulative reward, agents need to trade-off what is expected to be best at the moment, (i.e., exploitation), with potentially sub-optimal exploratory actions [20]. Solving this problem in an efficient way is a significant challenge. In this report, we turn to Thompson Sampling [23] for a solution, which combines exploitation and exploration in an implicit way.

1.1 Related Literature and Our Contributions

In the following paragraphs, we will briefly summarize the literature that is related to our problem. We then describe our contribution in this report.

Multinomial Classification Multinomial or multiclass classification is the problem of classifying instances into one of three or more classes. Though some classification algorithms are by nature binary algorithms, they can be turned into multinomial classifiers by a variety of strategies. There are various models, including neural networks[15], decision trees[19], support vector machines[8], etc. One common and powerful model is the multinomial logit model. It lies in the family of generalized linear models, and can extensively characterize a wide range of classification problems. One may ask whether the model has strong generalization ability as neural networks, but in this report, we will discuss various bayesian methods only based on logit model.

Preprint. Under review.

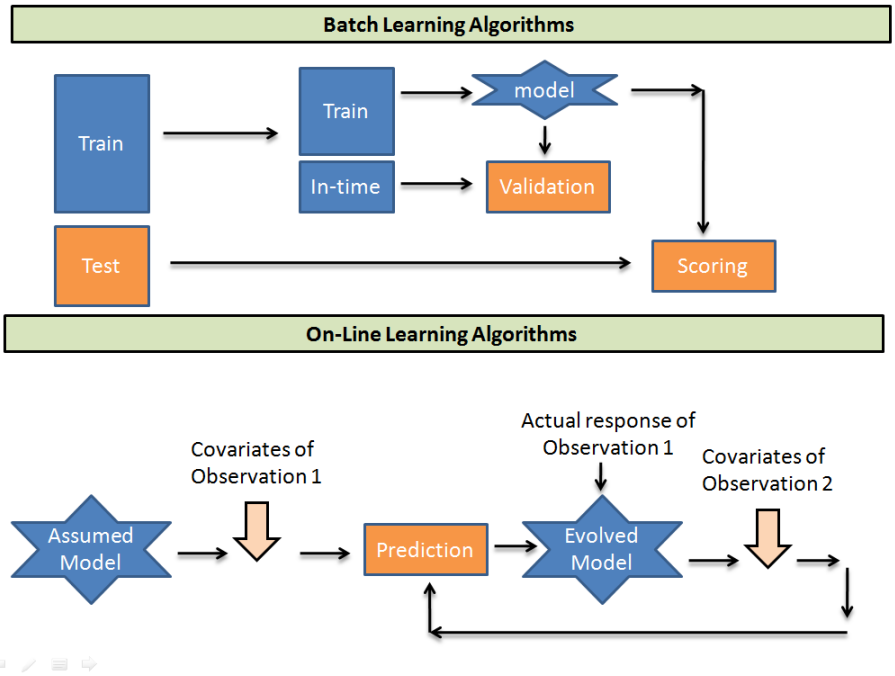

Figure 1: Comparison between traditional learning and online learning, image taken from this blog

Contextual Bandits Our problem can be treated as a special case of contextual bandits. Contextual Bandits have been extensively studied in the literature, and we are not going to carry out detailed explanation. Readers could refer to [24] for a comprehensive review. However, literature often focus on the *linear* stochastic case and the algorithms are based on "upper confidence bound" [16, 3]. Another mainstream is adversarial contextual bandits [6, 22], where everything are fixed in advance and there are no randomness. Our problem belongs to stochastic bandits, but have more complex structure than linear one.

Thompson Sampling Thompson sampling (TS) [23], also known as *posterior sampling*, is an algorithm for online decision problems where actions are taken sequentially in a manner that must balance between exploiting what is known to maximize immediate performance and investing to accumulate new information that may improve future performance [21]. It has aroused increasing attention in recent years because of its surprising performance in practice. For TS in contextual bandits, still linear cases is the mainstream [4]. As for logistic model, literature commonly focus on *binary* response, see, e.g., [7]. Even if the model is multinomial logistic, the solution method is not Bayesian [2].

Our Contribution

- *New setting* We combine online multinomial classification with multinomial logit models, which has been largely ignored in literature. Moreover, our definition of full feedback and semi feedback emphasizes the influence under different information gain.
- *New methods* We combine and extend the existing approximation methods to semi-feedback setting. The extension and modification is straightforward, easy to implement, and achieve good performance. We also extend the PG-TS method from binary response model to polychotomous responses model as well as design efficient algorithm in implementation. In the semi feedback setting, we derive from scratch a new formulation of approximation methods as well as a data augmentation method using Pólya-Gamma latent variables.
- *New experiments* We systematically conduct numerical experiments, both on simulated and real dataset. We test the performance of various methods and display their difference. We find that whatever the setting and dataset, most of the methods achieve relatively satisfactory results.

1.2 Organization

This report is organized as follows. In Section 2, we introduce the settings of our target problem. In Section 3, we introduce our practical algorithms, including Thompson Sampling, Laplace Approximation, Ensemble Sampling and Pólya-Gamma Latent Variables Strategy. We display the experimental results in Section 4 and summarize our report in Section 5.

2 Our Setting

Consider an online multinomial classification problem as follows. There are K possible categories (arms) in total. For time $t = 1, 2, ..., T$

- Observe a context(feature vector) x_t.
- Make a decision $k_t \in \mathcal{A}_t = \{0, ..., K-1\}$ about which class it belongs to based on x_t.
- Receive a reward r_t, with possibly some additional information y_t from the environment.
- Update our algorithm.

Our goal is to maximize the *cumulative reward* $\sum_{t=1}^{T} r_t$.

In this report, we will mainly consider the multinomial logit model

$$\mathbb{P}(\text{True label} = k|x_t) = \frac{\exp(\theta_k^T x_t)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_t)},$$

where $\theta_0 = 0$. The reward is defined as

$$r_t = \begin{cases} 1, & \text{Prediction is right.} \\ 0, & \text{Prediction is wrong.} \end{cases} \tag{1}$$

2.1 Full Feedback

In this setting, $y_t = (0, ..., 1, ..., 0)$ represents the true label. The likelihood can be written as

$$p(y_t|\theta, x_t, k_t) = \Pi_{k=0}^{K-1} \left(\frac{\exp(\theta_k^T x_t)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_t)} \right)^{y_{k,t}}.$$

The posterior can be written as

$$p_t(\theta) \propto p_0(\theta) \Pi_{\tau=1}^{t} \Pi_{k=0}^{K-1} \left(\frac{\exp(\theta_k^T x_\tau)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_\tau)} \right)^{y_{k,\tau}}.$$

2.2 Semi Feedback

In this setting, $y_t = r_t$ represents whether we predict the label correctly. The likelihood can be written as

$$p(y_t|\theta, x_t, k_t) = \left(\frac{\exp(\theta_{k_t}^T x_t)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_t)} \right)^{y_t} \left(1 - \frac{\exp(\theta_{k_t}^T x_t)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_t)} \right)^{1-y_t},$$

where $k_t \in \{0, ..., K-1\}$ is our action in period t. The posterior can be written as

$$p_t(\theta) \propto p_0(\theta) \Pi_{\tau=1}^{t} \left(\frac{\exp(\theta_{k_\tau}^T x_\tau)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_\tau)} \right)^{y_\tau} \left(1 - \frac{\exp(\theta_{k_\tau}^T x_\tau)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_\tau)} \right)^{1-y_\tau}.$$

Note that when $K = 2$, i.e., there are only 2 categories, and semi feedback setting is equivalent to full feedback one. When $K > 2$, semi feedback provides less information, especially when our prediction is wrong. This impose challenges for designing algorithms.

3 Algorithms

3.1 Thompson Sampling

In 3.1.1, we will briefly introduce and discuss the general TS. In 3.1.2, we will embed TS into our settings and present an algorithm framework.

3.1.1 General TS

Thompson Sampling is a classic algorithm. In each round (or time step), it draws a sample and chooses an action greedily under the optimal policy for the sample. A reward is then observed after the action is taken. The posterior distribution over models is then updated after that. More specific details can be seen in the 1.

Algorithm 1: General Thompson Sampling

Determine prior $p_0(\theta)$
for $t = 1, 2, ..., T$ **do**
  Receive context x_t;
  Sample θ^t according to $P(\theta|D)$;
  Select $a_t = \arg\max_a \mathbb{E}_r\left(r|x_t, a, \theta^t\right)$;
  Receive reward r_t ;
  $D = D \cup (x_t, a_t, r_t)$, which can be seen as updating posterior distribution ;
end

3.1.2 TS in our settings

We give the formal description in Algorithm 2. Note that Algorithm 2 only serves as a framework of embedding TS into our settings.

Algorithm 2: Framework for Online Multi-classification via Thompson Sampling

Determine prior $p_0(\theta)$;
for $t = 1, 2, ..., T$ **do**
  Receive context x_t;
  Sample $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_{K-1}$ from $p_{t-1}(\theta)$;
  Take action $k_t = \arg\max_{k\in\{0,...,K-1\}}\{\hat{\theta}_k^T x_t\}$ ($\hat{\theta}_0 = 0$);
  Receive r_t and collect all the available information y_t;
  Update posterior $p_t(\theta) \propto p_{t-1}(\theta)p(y_t|\theta, x_t, k_t)$;
end

There are two sub-problems that need to be dealt with in Algorithm 2, and we list them as follows.

- What is the prior $p_0(\theta)$? In this report, we will mainly consider (Multivariate) Normal Distribution. It is easy to handle and usually leads to closed form formula. More importantly, several convenient approximation methods rely on Normal Distribution.
- How to sample from/approximate the posterior? This is the most crucial problem in our algorithm design. As is shown in Section 2, the posterior does not have a simple expression and is thus hard to sample from. In 3.2 and 3.3, we will introduce approximation and sampling methods respectively. It's worth mentioning that, either we reserve the complex expression of the posterior but seek for an efficient sampling method, or we sacrifice the accuracy of posterior with a more convenient approximated posterior with easily sampling methods. Once we've managed to solve one of "Sampling" and "Approximation", then we do not need to deal with the other.

3.2 Approximation Algorithms

Common sampling methods may not be suitable. In each period, to obtain an appropriate posterior sample, we have to simulate from the beginning because each past observation must be accessed

to generate the next action. This means that the computation time required per time period grows as time progresses. In order to keep the computational burden manageable, it can be important to consider incremental variants of our approximation methods. We will only consider *incremental* update scheme with a fixed rather than growing per-period compute time.

In 3.2.1 and 3.2.2, we will introduce two approximation algorithms and present the corresponding algorithms *in full feedback setting*. In 3.2.3, we will solve the challenges of semi feedback setting through two methods.

3.2.1 Laplace Approximation

Laplace Approximation is a well-known approximation method of sampling from posterior, see, e.g., [13]. It approximates a potentially complicated posterior distribution by a Gaussian distribution. Samples from this simpler Gaussian distribution can then serve as approximate samples from the posterior distribution of interest. Chapelle and Li [7] proposed this method to approximate TS in a display advertising problem with a logistic regression model of ad-click-through rates. We now embed the method into our settings.

For each period, to obtain a Laplace Approximation, we need to find θ that maximizes

$$f_t(\theta) = \ln p_0(\theta) + \sum_{\tau=1}^{t} \ln p(y_\tau|\theta, x_\tau, k_\tau).$$

Suppose we have $\theta_{t-1} = \arg\max_\theta f_{t-1}(\theta)$, we want to find $\theta_t = \theta_{t-1} + \delta_t = \arg\max_\theta (f_{t-1}(\theta) + \ln p(y_t|\theta, x_t, k_\tau))$. Apply a first-order Taylor expansion, we have

$$\begin{aligned} 0 &= \nabla f_{t-1}(\theta_{t-1} + \delta_t) + \nabla \ln p(y_t|\theta_{t-1} + \delta_t, x_t, k_t) \\ &\approx \nabla f_{t-1}(\theta_{t-1}) + \nabla^2 f_{t-1}(\theta_{t-1})\delta_t + \\ &\quad \nabla \ln p(y_t|\theta_{t-1}, x_t, k_t) + \nabla^2 \ln p(y_t|\theta_{t-1}, x_t, k_t)\delta_t \\ &= \nabla^2 f_t(\theta_{t-1})\delta_t + \nabla \ln p(y_t|\theta_{t-1}, x_t, k_t) \end{aligned} \tag{2}$$

Therefore,

$$\begin{aligned} \delta_t &\approx -\left(\nabla^2 f_t(\theta_{t-1})\right)^{-1} \nabla \ln p(y_t|\theta_{t-1}, x_t, k_t) \\ &= -\left(\nabla^2 f_{t-1}(\theta_{t-1}) + \nabla^2 \ln p(y_t|\theta_{t-1}, x_t, k_t)\right)^{-1} \nabla \ln p(y_t|\theta_{t-1}, x_t, k_t). \end{aligned} \tag{3}$$

Note that when the likelihood is completely Gaussian, the update is precise, though this is not the case in our settings.

With this intuition in mind, we give Algorithm 3 formally. There are some points that need to be clarified.

- Since our model is a generalized linear model, the Hessian of the likelihood has rank one, thus the inverse of precision matrix can be done efficiently using Sherman-Morrison formula. In our implementation, we only restore and update $\Sigma_{k,t} = H_{k,t}^{-1}$.
- We model the parameters $\theta = (\theta_i)_i$ in an independent way. Otherwise, we need to compute a large Hessian Matrix that may not have rank one, which will greatly increase the computation.

3.2.2 Ensemble Sampling

This approach involves incrementally updating each of an *ensemble* of models to behave like a sample from the posterior distribution. The posterior can be interpreted as a distribution of "statistically plausible" models. With this interpretation in mind, Thomson Sampling can be thought of as randomly drawing from the range of statistically plausible models. Ensemble sampling aims to maintain, incrementally update, and sample from a finite set of such models.[17, 21]

Consider maintaining N models with parameters $\{\theta_0^n, H_0^n : n = 1, ..., N\}$, initialized with $\theta_0^n \sim p_0(\theta), H_0^n = \nabla_\theta^2 \ln p(\theta)\big|_{\theta=\theta_0^n}$. We update them according to

$$\begin{aligned} H_t^n &\leftarrow H_{t-1}^n - z_t^n \, \nabla_\theta^2 \ln p(y_t|\theta, x_t, k_t)\big|_{\theta=\theta_{t-1}^n} \\ \theta_t^n &\leftarrow \theta_{t-1}^n + z_t^n (H_t^n)^{-1} \, \nabla_\theta \ln p(y_t|\theta, x_t, k_t)|_{\theta=\theta_{t-1}^n} \end{aligned} \tag{4}$$

Algorithm 3: Incremental Update: Laplace Approximation

Determine prior density $p_0(\theta) = \Pi_{k=1}^{K-1} N(\mu_{k,0}, H_{k,0}^{-1})$;
for $t = 1, 2, ..., T$ **do**
  Receive context x_t;
  Sample $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_{K-1}$ from $p_{t-1}(\theta)$;
  Take action $k_t = \arg\max_{k \in \{0,...,K-1\}} \{\hat{\theta}_k^T x_t\}$ ($\hat{\theta}_0 = 0$);
  Receive r_t and y_t;
  for $k = 1, ..., K-1$ **do**
    $H_{k,t} \leftarrow H_{k,t-1} - \nabla_\theta^2 \ln p(y_t|\theta, x_t, k_t)\big|_{\theta=\mu_{k,t-1}}$;
    $\mu_{k,t} \leftarrow \mu_{k,t-1} + H_{k,t}^{-1} \nabla_\theta \ln p(y_t|\theta, x_t, k_t)|_{\theta=\mu_{k,t-1}}$;
  end
  $p_t(\theta) = \Pi_{k=1}^{K-1} N(\mu_{k,t}, H_{k,t}^{-1})$;
end

where $z_t^n \sim$ Poisson(1). To generate an action k_t, n is sampled uniformly from $\{1, ..., N\}$, and the action k_t is chosen to maximize $\mathbb{E}[y_t|\theta_t^n, x_t, k_t]$.

[21] gives the following interpretation on Ensemble Sampling. Each θ_t^n can be viewed as a random statistically plausible model, with randomness stemming from the initialization of θ_0^n and the random weight z_t^n placed on each observation. The variable, z_t^n, can loosely be interpreted as a number of replicas of the data sample (x_τ, y_τ) in the first t periods. Indeed, in a data set of size t, the number of replicas of a particular bootstrap data sample follows a Binomial$(t, 1/t)$ distribution, which is approximately Poisson(1) when t is large.

We embed Ensemble Sampling into our setting in Algorithm 4.

Algorithm 4: Incremental Update: Ensemble Sampling

Create N models with parameters $\{\theta_{k,0}^n, H_{k,0}^n : n = 1, ..., N; k = 1, ..., K-1\}$;
for $n = 1, ..., N$ **do**
  for $k = 1, ..., K-1$ **do**
    $\theta_{k,0}^n \sim p_0(\theta_{k,0})$;
    $H_{k,0}^n = \nabla_\theta^2 \ln p_0(\theta)\big|_{\theta=\theta_{k,0}^n}$;
  end
end
for $t = 1, ..., T$ **do**
  Receive context x_t;
  Sample uniformly in $\{1, ..., N\}$ and obtain corresponding $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_{K-1}$;
  Take action $k_t = \arg\max_{k \in \{0,...,K-1\}} \{\hat{\theta}_k^T x_t\}$ ($\hat{\theta}_0 = 0$);
  Receive r_t and y_t;
  for $n = 1, ..., N$ **do**
    $z_t^n \sim$ Poisson(1);
    for $k = 1, ..., K-1$ **do**
      $H_{k,t}^n \leftarrow H_{k,t-1}^n - z_t^n \nabla_\theta^2 \ln p(y_t|\theta, x_t, k_t)\big|_{\theta=\theta_{k,t-1}^n}$;
      $\theta_{k,t}^n \leftarrow \theta_{k,t-1}^n + z_t^n (H_{k,t}^n)^{-1} \nabla_\theta \ln p(y_t|\theta, x_t, k_t)|_{\theta=\theta_{k,t-1}^n}$;
    end
  end
end

3.2.3 Challenges and Solutions for Semi Feedback

In full feedback occasion, the log-likelihood is always *log-concave*. Thus in some sense, Laplace Approximation is sufficient to approximate the posterior. However, in semi feedback setting, the

likelihood is *log-concave* to θ_{k_t} but not necessarily to $k \neq k_t$ when $y_t = 0$. Update of Precision Matrices may violate the positive definiteness. We modify the update scheme and propose the following strategy in semi feedback setting:

- When $y_t = 1$, the log-likelihood is log-concave, and we update as usual.
- When $y_t = 0$, the log-likelihood is only guaranteed to be log-concave to θ_{k_t}, and we only update parameters relative to $\theta_{k_t}(\{\theta^n_{k_t,t}, H^n_{k_t,t} : n = 1, ..., N\})$.

Another technique for stabilizing and improving the performance is to *diagonalizing* the *Precision Matrix* (H in previous discussions) after each time we update it. In [20], the authors demonstrated through numerical experiments that such technique is quite promising, especially when the model is simple and the contexts do not have much correlation.

3.3 Sampling Method: Data Augmentation using Pòlya-Gamma Variable

In 3.3.1, we will briefly introduce the definition as well as some properties of Pólya-Gamma random variables. Then we will show how to simulate it using an efficient sampler proposed in [9]. In 3.3.2, we will formulate the data augmentation strategy using latent Pólya-Gamma variables to solve the full feedback problem in 2.1. Lastly, in 3.3.3, we will discuss some exploration for the semi feedback problem in 2.2.

3.3.1 The Pólya-Gamma Random Variable

Consider solving the logistic model using Thompson Sampling. In algorithm 2, to sample from the posterior $p_t(\theta)$, we have used Laplace and Ensemble approximations in 3.2.1, 3.2.2 to approximate the distribution. However, the distribution can also be approximated using MCMC by introducing latent variables. This scheme is often referred to as Data Augmentation. The Gibbs sampler can be analytically derived by adding Pólya-Gamma latent variables. Following from [18], we define the PG family as below.

Definition 1. *A positive random variable W has a Pólya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$, if*

$$W \overset{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + c^2/(4\pi^2)}$$

where $\overset{D}{=}$ means equality in distribution and g_k are independent variables following a Gamma distribution of $Ga(b, 1)$. Denote W as $W \sim PG(b, c)$.

By the definition and Weierstrass factorization theorem, the Laplace transform of $W \sim PG(b, 0), b > 0$ is

$$\mathbb{E}\{\exp(-Wt)|b, 0\} = \prod_{k=1}^{\infty} (1 + \frac{t}{2\pi^2(k - 1/2)^2})^{-b} = \frac{1}{\cosh^b(\sqrt{t/2})} \tag{5}$$

and the Laplace transform of $W \sim PG(b, c), b > 0, c \in \mathbb{R}$ is

$$\mathbb{E}\{\exp(-Wt)|b, c\} = \prod_{k=1}^{\infty} (\frac{1 + \frac{c^2/2}{2\pi^2(k-1/2)^2}}{1 + \frac{c^2/2+t}{2\pi^2(k-1/2)^2}})^b = \frac{\cosh^b(c/2)}{\cosh^b(\sqrt{\frac{c^2/2+t}{2}})} \tag{6}$$

Compare (5) and (6), we have

$$p(\omega|b, c) = \frac{\exp(-\frac{c^2}{2}\omega) p(\omega|b, 0)}{\mathbb{E}\{\exp(-\frac{c^2}{2}\omega)|b, 0\}} \tag{7}$$

where $p(\cdot|b, c)$ is the density of $PG(b, c)$.

Simulation From the definition, a naive approach to simulate PG variable is using the summation of Gamma variables. However, this involves the truncation of an infinite sum which can be numerical dangerous and slow to calculate. We therefore adopt a rejection sampling method proposed in [18].

This method lies on the foundation of [1]–that PG family is closely related to the Jacobi Theta and Riemann Zeta functions $J^*(1)$, as well as [9]–that an efficient accept/reject algorithm can be developed for $J^*(1)$.

The formulation and pseudo-code for sampling from $PG(1, z), z > 0$ can be found in the appendix. For $PG(n, z)$ with $n \in \mathbb{N}$, notice from (6), its Laplace transform can written as a multiplication of n Laplace transform of $PG(1, z)$, thus $PG(n, z)$ can be written as the summation of n independent $PG(1, z)$ variables, i.e. $PG(n, z) = \sum_{i=1}^{n} PG(1, z)$. Thus, sampling from $PG(n, z)$ with $n \in \mathbb{N}$ and $z > 0$ can be easily obtained.

3.3.2 Data Augmentation Strategy

Data Augmentation provides a general framework for analyzing binary and polychotomous response model ([5]). The idea is to add auxiliary variables to perform Gibbs sampling for the posterior.

For the logistic model, [5] formulated a Gibbs sampler by assuming the responses depend on a threshold defined by an underlying continuous variable. Later methods, such as in [14, 12], generally followed the same mechanism in [5]. Here, we adopt an alternative data augmentation strategy proposed in [18].

The PG strategy is based on the analytical form of the posterior distribution in the logistic model. We add will PG random variables as auxiliary variables and show that the posterior can be transform analytically into a multiple of two conditional probability, thus formulate the Gibbs sampler. This scheme differs from others in that it is exact and it only requires one layer of latent variables. Experiments in [18] also showed that it is more efficient and accurate in many datasets than all previously proposed data-augmentation schemes.

2 categories We start by the simple case where there is only $K = 2$ arms. In this case, knowing $y_{1,t}$ is equivalent to knowing y_t. Thus, in the following of this paragraph, we write y_t to represent $y_{1,t}$ and θ to represent θ_1 for simplicity.

The posterior distribution in the $t + 1$ period is

$$p_t(\theta) \propto p_0(\theta) L(\theta | D_t)$$

where D_t is the first t periods of data and $L(\theta|D_t)$ is the likelihood function.

The t's period's contribution to the likelihood is

$$L_t(\theta) = \frac{\{\exp(x_t^T \theta)\}^{y_t}}{1 + \exp(x_t^T \theta)} \tag{8}$$

From (7), taking the integrals of both sides,

$$\int_0^\infty \exp\{-\omega_t (x_t^T \theta)^2/2\} p(\omega_t|1, 0) d\omega_t = \mathbb{E}\{\exp(-\frac{(x_t^T \theta)^2}{2} \omega_t)|b, 0\}$$

appealing to (5), the right side is

$$\mathbb{E}\{\exp(-\frac{(x_t^T \theta)^2}{2} \omega)|b, 0\} = \frac{1}{\cosh(x_t^T \theta/2)}$$

thus,

$$\begin{aligned} \exp(h_t x_t^T \theta) \int_0^\infty \exp\{-\omega_t (x_t^T \theta)^2/2\} p(\omega_t|1, 0) d\omega_t &= \frac{\exp\{-\omega_t (x_t^T \theta)^2/2\}}{\cosh(x_t^T \theta/2)} \\ &\propto \frac{\{\exp(x_t^T \theta)\}^{y_t}}{1 + \exp(x_t^T \theta)} \end{aligned}$$

Then plugging into (8), we have

$$L_t(\theta) \propto \exp(h_t x_t^T \theta) \int_0^\infty \exp\{-\omega_t (x_t^T \theta)^2/2\} p(\omega_t|1, 0) d\omega_t \tag{9}$$

where $h_t = y_t - 1/2$, $p(\omega_t|1,0)$ is the density of $PG(1,0)$.

Using the total probability formula, we write $L_t(\theta) = \int p(y_t, \omega_t|\theta)d\omega_t$, then from (9) the joint distribution is

$$p(y_t, \omega_t|\theta) = \exp(h_t x_t^T \theta) \exp\{-\omega_t (x_t^T\theta)^2/2\} p(\omega_t|1,0)$$

Thus, the conditional distribution is

$$p(\omega_t|\theta) = \sum_{y_t} p(y_t, \omega_t|\theta) \propto \exp\{-\omega_t (x_t^T\theta)^2/2\} p(\omega_t|1,0)$$

And the conditional posterior of θ is

$$\begin{aligned} p(\theta|\omega, D_t) &\propto p_0(\theta) \prod_{i=1}^{t} L_t(\theta|\omega_i) \\ &\propto p_0(\theta) \prod_{i=1}^{t} \exp\{h_i x_i^T\theta - \omega_i (x_i^T\theta)^2/2\} \\ &= p_0(\theta) \exp\{-(z - X\theta)^T \Omega (z - X\theta)/2\}, \ for\ some\ z \end{aligned}$$

Thus, if the prior is a normal distribution $N(\mu, H)$, then the Gibbs sampler for $p_t(\theta)$ is

$$\begin{aligned} \omega_i|\theta &\sim PG(1, x_i^T\theta) \\ \theta|D_t, \omega &\sim N(m_\omega, V_\omega) \end{aligned} \tag{10}$$

where $i = 1, ..., t$ and

$$\begin{aligned} V_\omega &= (X^T \Omega X + H^{-1})^{-1} \\ m_\omega &= V_\omega (X^T Y + H^{-1}\mu) \end{aligned}$$

with $\Omega = diag(\omega_1, ..., \omega_t)$ and $Y = (y_1 - 1/2, ..., y_t - 1/2)$.

K categories Following from the function in 2.1, the likelihood of θ_j on the rest of θ denoting as θ_{-j} is

$$L(\theta_j|\theta_{-j}, D_t) = \prod_{i=1}^{t} \left(\frac{e^{\eta_{ij}}}{1 + e^{\eta_{ij}}}\right)^{y_{j,i}} \left(\frac{1}{1 + e^{\eta_{ij}}}\right)^{1 - y_{j,i}}$$

where $\eta_{ij} = x_i^T \theta_j - c_{ij}$ with $c_{ij} = \log \sum_{k \neq j} \exp x_i^T \theta_k$.

This formulation resembles the binary case (8), thus incorporating PG variables we can similarly formulate the Gibbs sampler as

$$\begin{aligned} \omega_{ij}|\theta_j &\propto PG(1, x_i^T\theta_j - c_{ij}) \\ \theta_j|\Omega_j &\propto N(m_j|V_j) \end{aligned} \tag{11}$$

where the prior of θ_j is $N(\mu_j, H_j)$ and

$$\begin{aligned} V_j &= (X^T \Omega_j X + H_j)^{-1} \\ m_j &= V_j (X^T (Y_j - \Omega_j c_j) + H_j^{-1} \mu_j) \end{aligned}$$

with $\Omega_j = diag(\omega_{1j}, ..., \omega_{tj})$, c_j being the jth column of c and $Y_j = (y_{j,1} - 1/2, ..., y_{j,t} - 1/2)$. One may sample using (11) by iterating through $j = 1, 2, ..., K$ or simply sample new versions of θ_j after all ω_{ij} are sampled based on the previous θ_js.

PG-TS algorithm The PG-TS algorithm uses the Gibbs sampler in (11) to sample from the posterior in the Thompson Sampling algorithm 2.

The pseudo-code is as below.

Algorithm 5: PG-TS

Choose prior $p_0(\theta_j) = N(\mu_j, H_j)$ where $j = 1, ..., K-1$ and the number of layers of Gibbs sampler M;
for $t = 1, 2, ..., T$ **do**
  Receive context x_t;
  for $m = 1, 2, ..., M$ **do**
    for $i = 1, 2, ..., t-1$ **do**
      Sample ω_{ij} from $PG(1, x_i^T\theta_j - c_{ij})$ where $j = 1, ..., K-1$
    end
    $\Omega_j = diag(\omega_{1j}, ..., \omega_{tj})$ where $j = 1, ..., K-1$ where $j = 1, ..., K-1$;
    $Y_j = (y_{j,1} - 1/2, ..., y_{j,t} - 1/2)$ where $j = 1, ..., K-1$;
    $m_j = V_j(X^T(Y_j - \Omega_j c_j) + H_j^{-1}\mu_j)$;
    $V_j = (X^T\Omega_j X + H_j)^{-1}$ where $j = 1, ..., K-1$;
    Sample θ_j from $N(m_j, V_j)$ where $j = 1, ..., K-1$;
  end
  Take action $k_t = \arg\max_{k \in \{0,...,K-1\}}\{\theta_k^T x_t\}$ ($\theta_0 = 0$);
  Receive r_t and collect all the available information y_t;
end

In practice, one layer of Gibbs sampler is enough. Since in [5], they showed that combining data augmentation and Gibbs sampler only requires us to sample once from the conditional probabilities. Intuitively, this is because when t is large, D_t contains almost the same information as D_{t-1}, thus θ_{t-1} follows similar distribution as θ_t. Also, experiments in [11] showed the result for both $M = 1$ and $M = 100$, there appeals no significant difference in the performances.

3.3.3 Explorations for Semi Feedback

In the semi feedback problem, the likelihood of θ is in 3.2.3. Similar to the above case, we have the likelihood of θ_j on the rest of θ denoting as θ_{-j} as

$$L(\theta_j|\theta_{-j}, D_t) = \prod_{i=1}^{t}(\frac{e^{\eta_{ij}}}{1+e^{\eta_{ij}}})^{\hat{y}_{j,i}}(\frac{1}{1+e^{\eta_{ij}}})^{1-\hat{y}_{j,i}}$$

where $\hat{y}_{j,i} = y_i 1\{k_i = j\}$ with k_i being the label assigned to x_i by the machine and y_i being the regret of the ith period, and η is defined the same as in the K categories case above.

Thus the Gibbs sampler has the formulation as in (11) expect that Y is substituted with $\hat{Y}$.

However, as appealing as this formulation is, a general problem with semi feedback problem still exists. That is when the number of arms is large, the period that predicts the wrong label doesn't provide much information to the machine, in the PG case it doesn't change the Gibbs sampler very much. This makes the probability of predicting the right label in the next period stay almost the same, thus makes the results highly rely on the choice of initial value. The performance is prone to be highly related with the prior distribution.

4 Experiments

In this section, we will examine Thompson Sampling with different approximation/sampling methods through numerical experiments. In 4.1, we will carry out experiments on simulated data. In 4.2, we will experiment on real data for classification.

4.1 Simulated Data

In this subsection, we will carry out experiments on simulated data with K arms, d features and T periods (The three parameters vary according to different settings). Each arm k corresponds to a

vector $\theta_k \in \mathbb{R}^{d\times 1}$ sampled from $\mathcal{N}(0, 5I_d)$. The true label of a context $x \in \mathbb{R}^{d\times 1}$ sampled from $\mathcal{N}(0, 10I_d)$ is obtained by $\mathbb{P}(\text{True label} = k|x) = \frac{\exp(\theta_k^T x)}{\sum_{t=0}^{K-1}\exp(\theta_t^T x)}$, which is same as (1).

Our first measurement in time t is *cumulative regret*, i.e., the total number of mis-classification before t. The second measurement is *average regret*, obtained by averaging the total number of mis-classification in $(t-100, t](t \geqslant 100)$.

In the following 4.1.1, we will compare all the related methods in Section 3.

4.1.1 Gathering all Methods

We test these methods mentioned above under two different settings. In setting 1, we set $K = 2$, $d = 10$, $T = 5000$, while in setting 2, we set $K = 5$, $d = 10$, $T = 1000$. All results are averaged 25 times. For full feedback, the results are in Figure 2 and Figure 3. For semi feedback, the results are in Figure 4. We do not show semi feed back results for simulation 1 because when $K = 2$, full and semi are actually the same as analyzed before. "-pre" means that the method is combined with diagnolizing the preicison matrices.

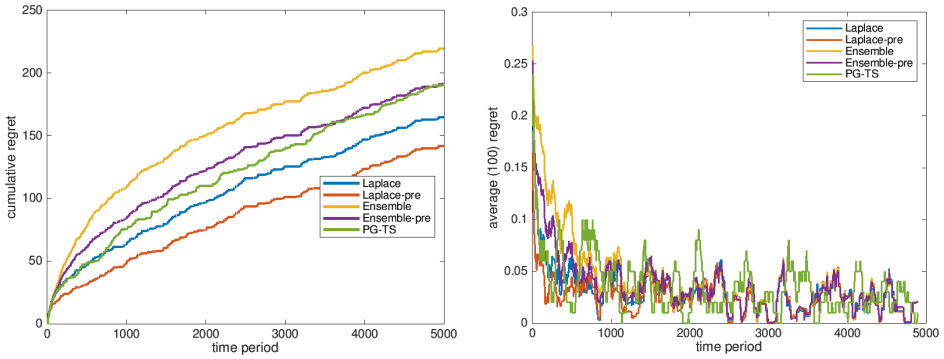

Figure 2: Full Feedback under Simulation Setting 1

We can see that, in full feedback setting, Laplace Approximation performs better than Ensemble Sampling and PG-TS. Especially, diagnolizing precision matrices could indeed improve performance. What's more, in semi feedback setting, the performance of PG-TS drop quite a bit while other methods remain approximately the same. This may due to the fact that for semi feedback setting, PG sampling is not very suitable for approximating the distribution.

4.2 Real Data

4.2.1 Shuttle

The Shuttle Statlog Dataset [10] provides the value of d = 9 indicators during a space shuttle flight, and the goal is to predict the state of the radiator subsystem of the shuttle. There are k = 7 possible states, and if the agent selects the right state, then reward 1 is generated. Otherwise, the agent obtains no reward (r = 0). The most interesting aspect of the dataset is that one action is the optimal one in 80% of the cases, and some algorithms may commit to this action instead of further exploring.

4.2.2 Covertype

The Covertype Dataset [10] classifies the cover type of northern Colorado forest areas in k = 7 classes, based on d = 54 features, including elevation, slope, aspect, and soil type. Again, the agent obtains reward 1 if the correct class is selected, and 0 otherwise.

4.2.3 Gathering All Methods

We test the methods mentioned above for two different datasets, Shuttle and Covertype, and show their performance in the following figures.

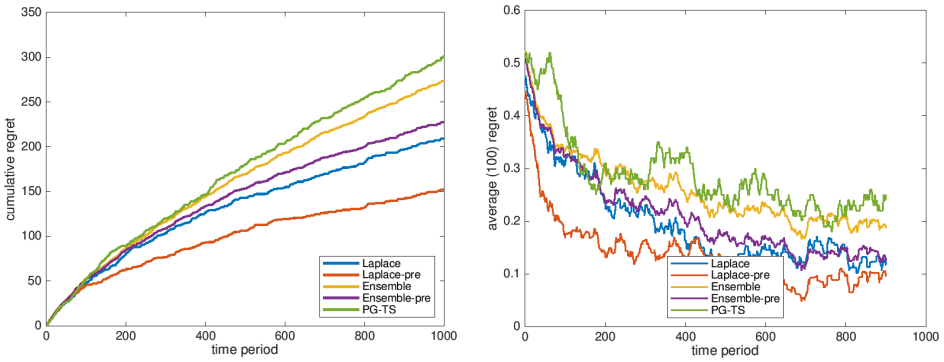

Figure 3: Full Feedback under Simulation Setting 2

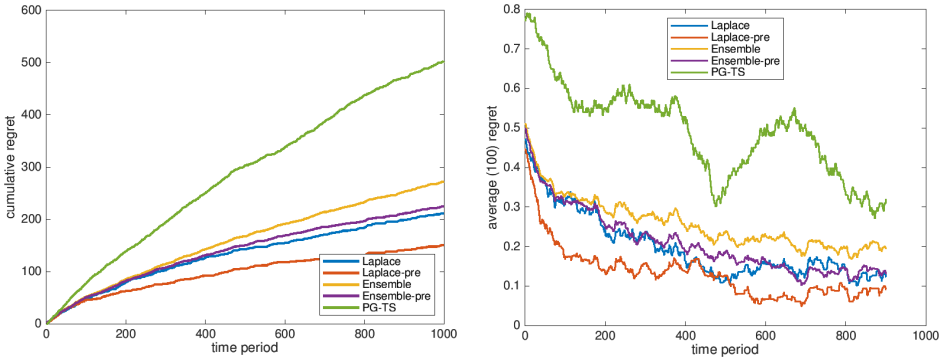

Figure 4: Semi Feedback under Simulation Setting 2

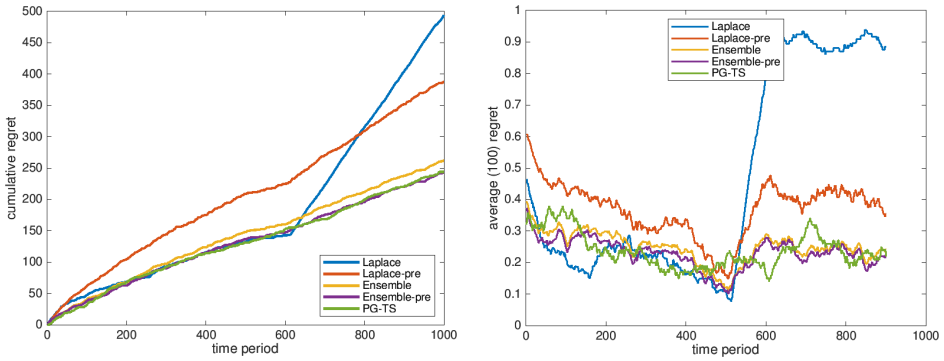

Figure 5: Full Feedback with Shuttle Dataset

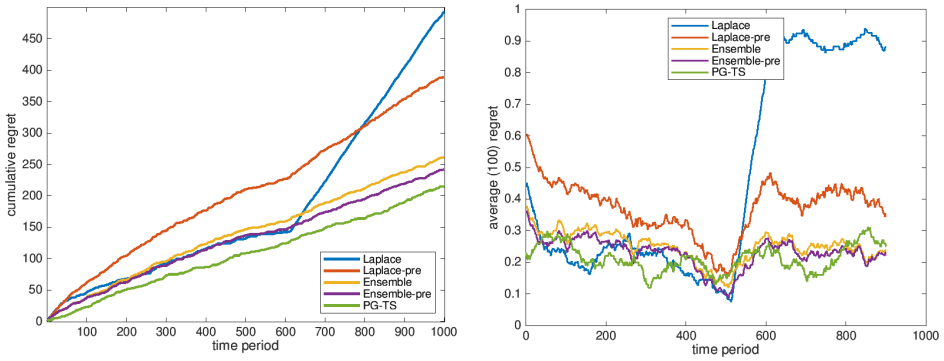

Figure 6: Semi Feedback with Shuttle Dataset

For full feedback Shuttle dataset, Figure 5 shows that PG-TS performs better than other methods. Notice that Shuttle dataset can be seen as a seven class classification problem, thus all the methods achieve non trivial solutions. As for semi feedback setting, algorithms have similar behaviours and PG-TS still get the top performance. Furthermore, Ensemble-pre achieves lower regret than ensemble method. This may result from the rationality of diagonalization assumption.

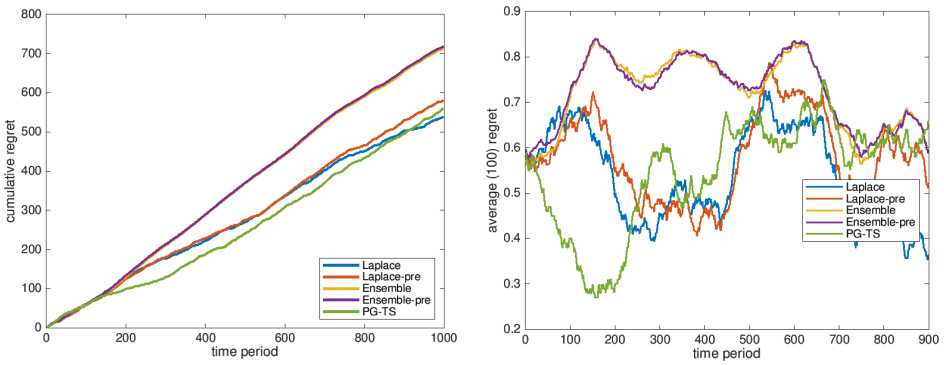

Figure 7: Full Feedback with Covertype Dataset

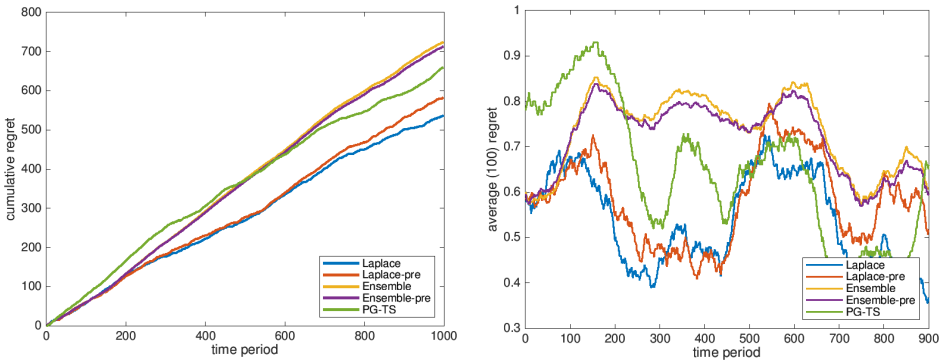

Figure 8: Semi Feedback with Covertype Dataset

For Covertype dataset, all these methods don't have a promising performance under both full and semi feedback setting, which can be seen from Figure 7 and Figure 8. All of them classify contexts mistakenly for more than half time steps. This may result from that the Covertype dataset doesn't follow our multinomial assumption. Though, among these methods, PG-TS also achieves the best performance as it does in Shuttle dataset, which demonstrates the superiority of exact inference. On the other hand, as in the Shuttle dataset, PG-TS lose its dominating place in semi feedback setting.

5 Conclusions

In this report, we delve into the online multi-classification problem under multinomial logit model. Under Thompson Sampling framework, we mainly develop two ways of solutions: gaussian approximation methods and PG-TS methods. We systematically test these methods with both simulated and real datasets, through which we demonstrate the power of our proposed methods. On the simulated datasets, the approximation methods perform better than PG-TS, since they capture the structure of the underlying model with high quality. On the real datsets, PG-TS has better performance, for it relies on the analytic form of the posterior and the data-augmentation scheme guarantees high accuracy of sampling.

Contribution of each member The part of approximation methods is developed by Feng Zhu and the part of Pólya-Gamma Thompson Sampling is developed by Jiaqi Zhang, while other investigation and experiments part is developed by Dinghuai Zhang.

References

[1] Biane , Philippe , Author , Pitman , Jim , Yor , and Marc . Probability laws related to the jacobi theta and riemann zeta functions, and brownian excursions. *Bulletin of the American Mathematical Society*, 01 2001.

[2] Alekh Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *International Conference on Machine Learning*, pages 1220–1228, 2013.

[3] 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.

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

[5] J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–79, 1993.

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

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

[8] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[9] Luc Devroye. On exact simulation algorithms for some distributions related to jacobi theta functions. *Statistics & Probability Letters*, 79(21):2251 – 2259, 2009.

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[11] Bianca Dumitrascu, Karen Feng, and Barbara Engelhardt. Pg-ts: Improved thompson sampling for logistic contextual bandits. 05 2018.

[12] Sylvia Frühwirth-Schnatter and Rudolf Frühwirth. *Data Augmentation and MCMC for Binary and Multinomial Logit Models*, pages 111–132. 01 2010.

[13] Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[14] Chris C. Holmes and Leonhard Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Anal.*, 1(1):145–168, 03 2006.

[15] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[16] 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.

[17] Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In *Advances in Neural Information Processing Systems*, pages 3258–3266, 2017.

[18] Scott Polson and Windle. Bayesian inference for logistic models using po lya-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

[19] J. R. Quinlan. Induction of decision trees. *MACH. LEARN*, 1:81–106, 1986.

[20] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

[21] 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.

[22] Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert Schapire. Efficient algorithms for adversarial contextual learning. In *International Conference on Machine Learning*, pages 2159–2168, 2016.

[23] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[24] Li Zhou. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.

Appendix

A Sampling from Pólya-Gamma distribution $PG(1, z)$

Define the Jacobi distribution $J^*(1)$ as

$$\mathbb{E}\{e^{tJ^*(b)}\} = \cosh^{-b}(\sqrt{t/2})$$

and its exponentially tilted distribution $J^*(1, z)$ as

$$f(x|z) = \cosh(z)e^{-xz^2/2}f(x)$$

where $f(x)$ is the density of $J^*(1)$.

From (6), we have the relation between $PG(1, z)$ and $J^*(1, z)$

$$PG(1, z) = \frac{1}{4}J^*(1, z/2)$$

An efficient sampler for $J^*(1)$ using accept/reject algorithm is developed in [9]. [18] extended this method to sampling from $J^*(1, z)$ therefore $PG(1, z)$.

Suppose we want to sample from the density f, the accept/reject algorithm is

1 Sample X from a density g

2 Draw U from uniform distribution $\mathcal{U}(0, cg(X))$ where $\|f/g\| \leq c$

3 Accept X iff $U \leq f(X)$

Define the piece-wise coefficient function as

$$a_n(x|z) = \begin{cases} (n+1/2)\pi(\frac{2}{\pi x})^{3/2}\cosh(z)\exp\{-\frac{z^2x}{2} - \frac{2(n+1/2)^2}{x}\} & 0 < x \leq t \\ (n+1/2)\pi\cosh(z)\exp\{-\frac{z^2x}{2} - \frac{(n+1/2)^2\pi^2 x}{2}\} & x > t \end{cases}$$

[9] showed that if t is near 0.64, then

$$f(x|z) = \sum_{n=1}^{\infty}(-1)^n a_n(x|z)$$

Notice that $a_n(x|z)$ is decreasing with n. Thus let $S_N(x) = \sum_{n=1}^{N}(-1)^n a_n(x|z)$ be the partial sum of f, then step 3 of the accept/reject algorithm is accepting X if $U \leq S_i(X)$ for some odd i and rejecting if $U > S_i(X)$ for some even i. The partial sum can be calculated iteratively. Also, it is nature to let the function $cg(X)$ in step 2 to be $S_0(X)$.

We choose the function g in step 1 to be a mixture of an inverse-Gaussian and an exponential

$$X \sim \begin{cases} IG(|z|^{-1}, 1)1(0 < X \leq t) & with\ probability\ p/(p+q) \\ Exp(-z^2/2 + \pi^2/8)1(X > t) & with\ probability\ q/(p+q) \end{cases}$$

where $p = \int_0^t cg(x)dx$ and $q = \int_t^{\infty} cg(x)dx$.

The pseudo-code for this accept/reject algorithm is as below.

Algorithm 6: Sample from $PG(1, z)$

Input $z > 0$, $t = 0.64$;
Define $a_n(x|z)$ as above, IG is the inverse-Gaussian distribution, Exp is the exponential distribution;
$p \leftarrow \frac{\pi}{\pi^2/4+z^2} \exp\{-(\pi^2/8 + z^2/2)t\}$ and $q \leftarrow 2\exp(-z)IG(t|1/z, 1.0)$;
repeat
  Draw V from $\mathcal{U}(0, 1)$;
  if $V < p/(p + q)$ **then**
    repeat
      Draw X from $IG(|z|^{-1}, 1)$
    until $X \leq t$;
  else
    repeat
      Draw X from $Exp(-z^2/2 + \pi^2/8)$
    until $X > t$;
  end
  Calculate $S = a_0(X)$ and draw U from $\mathcal{U}(0, S)$;
  $n \leftarrow 0$;
  repeat
    $n \leftarrow 1$;
    if *n odd* **then**
      $S \leftarrow S - a_n(X)$;
      return $X/4$ if $U < S$;
    else
      $S \leftarrow S + a_n(X)$;
      break if $U < S$;
    end
  until *FALSE*;
until *FALSE*;

[9] showed a lower bound of the acceptance rate of 0.61. Thus this is an rather efficient algorithm. The analysis of acceptance rate can be found in [18]. We implement this sampling method in MATLAB.