Online Multi-classification via Thompson Sampling

Dinghuai Zhang, Jiaqi Zhang, Feng Zhu

May 2019

Table of contents

Table of Contents

Problem: General Framework

Consider an online multi-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* $\sum_{t=1}^{T} r_t$.

Solution Idea

Basic Idea: Multinomial Logistic Regression + Thompson Sampling in Contextual Bandits

- Multinomial Logistic: $\mathbb{P}(\text{True label} = k | x_t) = \frac{\exp(\theta_k^T x_t)}{\sum_{l=0}^{K-1} \exp(\theta_l^T x_t)}$, where $\theta_0 = 0$.
- Thompson Sampling in Contextual Bandits

Bernoulli Bandit: A motivating example

- The arm set $\mathcal{A}_t$ remains unchanged for all t.
- The context is the same for all t.

Problem Statement:

- T periods
- In each period t, choose one action k_t out of K actions
- If we choose the k th action, we get a reward $r_t \sim \text{Bernoulli}(\theta_k)$
- $(\theta_1, \ldots, \theta_K)$ are unknown but fixed over time
- target is to maximize the cumulative reward $\sum_{t=1}^{T} r_t$

Bernoulli bandit

- Need a way to update our the knowledge we have about all θ_k.
- Thompson sampling chooses a Bayesian way.

Bernoulli bandit

Model the prior for θ_k:

$$p\left(\theta_k\right) \sim \text{Beta}(\alpha_k, \beta_k) \propto \theta_k^{\alpha_k - 1}\left(1 - \theta_k\right)^{\beta_k - 1}$$

If we choose $k_t = k$ at time t, the bernoulli likehood is:

$$p(r_t|\theta_k) = \theta_k^{r_t}\left(1 - \theta_k\right)^{1 - r_t}$$

Update of posterior:

$$\text{posterior} \propto p\left(\theta_k\right) p(r_t|\theta_k) \propto \theta_k^{(\alpha_k + r_t) - 1}\left(1 - \theta_k\right)^{(\beta_k + 1 - r_t) - 1}$$

which is:

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } k_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } k_t = k \end{cases} \tag{1}$$

Algorithms

Algorithm 1 BernGreedy(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #estimate model:
  for $k = 1, \ldots, K$ **do**
    $\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)$
  end for

  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$
end for

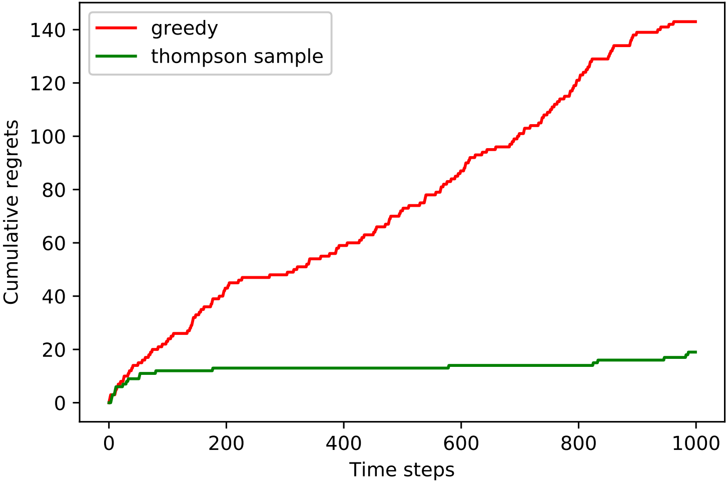
Algorithm 2 BernThompson(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #sample model:
  for $k = 1, \ldots, K$ **do**
    Sample $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)$
  end for

  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$
end for

Results



greedy
thompson sample
Cumulative regrets
Time steps
0
20
40
60
80
100
120
140
0
200
400
600
800
1000

General Thompson Sampling

Algorithm 1 Thompson sampling

$D = \emptyset$
for $t = 1$ to T **do**
  Receive context x_t
  Draw $\theta^t \sim P(\theta|D)$
  Select $a_t = \arg\max_a \mathbb{E}_r \left(r|x_t, a, \theta^t\right)$
  Observe reward r_t
  Update posterior $P(\theta|D)$ with $D = D \cup (x_t, a_t, r_t)$
end for

Table of Contents

General Algorithm in our setting

Algorithm 1: Framework for Online Multi-classification via Thompson Sampling

```
Determine prior $p_0(\theta)$ *(What is the prior?)*;
**for** $t = 1, 2, ..., T$ **do**
    Receive context $x_t$;
    Sample $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_{K-1}$ from $p_{t-1}(\theta)$ *(How to sample?)*;
    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 $y_t$ *(Full feedback or semi feedback?)*;
    Update posterior $p_t(\theta) \propto p_{t-1}(\theta) p(y_t | \theta, x_t, k_t)$ *(How to update?)*;
**end**
```

Prior

- In this talk, we will mainly consider (Multivariate) Normal Distribution.
- (Multivariate) Normal Distribution is easy to handle and usually leads to closed form formula.
- More importantly, several convenient approximation methods rely on Normal Distribution.

Feedback

- We mainly consider two different settings of feedback. In both settings, $r_t = \begin{cases} 1, & \text{Prediction is right.} \\ 0, & \text{Prediction is wrong.} \end{cases}$
  - Full Feedback: $y_t = (0, ..., 1, ..., 0)$ represents the true label.
  - Semi Feedback: $y_t = r_t$
- Note that when $K = 2$, i.e., there are only 2 categories, semi feedback is also full feedback.
- When $K > 2$, semi feedback provides less information, especially when our prediction is wrong.
- We will use y_t instead of r_t in the following.

Posterior

- Full feedback: $p(y_t|\theta, x_t, k_t) = \Pi_{k=0}^{K-1}\left(\frac{\exp(\theta_k^T x_t)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_t)}\right)^{y_{k,t}}$. In this occasion, $p_t(\theta) \propto p_0(\theta)\Pi_{\tau=1}^{t}\Pi_{k=0}^{K-1}\left(\frac{\exp(\theta_k^T x_\tau)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_\tau)}\right)^{y_{k,\tau}}$
- Semi Feedback:
$p(y_t|\theta, x_t, k_t) = \left(\frac{\exp(\theta_{k_t}^T x_t)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_t)}\right)^{y_t}\left(1 - \frac{\exp(\theta_{k_t}^T x_t)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_t)}\right)^{1-y_t}$, where $k_t \in \{0, ..., K-1\}$ is our action in period t. In this occasion,
$p_t(\theta) \propto p_0(\theta)\Pi_{\tau=1}^{t}\left(\frac{\exp(\theta_{k_\tau}^T x_\tau)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_\tau)}\right)^{y_\tau}\left(1 - \frac{\exp(\theta_{k_\tau}^T x_\tau)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x_\tau)}\right)^{1-y_\tau}$

Laplace Approximation [**chapelle2011empirical**]

- For each period, to obtain a Laplace Approximation, we originally seek 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 series, we obtain $\delta_t \approx -\left(\nabla^2 f_t(\theta_{t-1})\right)^{-1} \nabla \ln p(y_t | \theta_{t-1}, x_t, k_t)$.

Laplace Approximation

Algorithm 2: 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**
    $H_{k,t} \leftarrow H_{k,t-1} - \nabla_\theta^2 \ln p(y_t|\theta, x_t, k_t)\big|_{\theta=\mu_{k,t-1}}$;
    $\mu_{k,t} \leftarrow \mu_{k,t-1} + H_{k,t}^{-1} \nabla_\theta \ln p(y_t|\theta, x_t, k_t)\big|_{\theta=\mu_{k,t-1}}$;
  end
  $p_t(\theta) = \Pi_{k=1}^{K-1} N(\mu_{k,t}, H_{k,t}^{-1})$;
end

Ensemble Sampling [lu2017ensemble, russo2018tutorial]

- 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

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

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

Ensemble Sampling

Algorithm 3: Incremental Update: Ensemble Sampling

```
Create N models with parameters {θ^n_{k,0}, H^n_{k,0} : n = 1, ..., N; k = 1, ..., K − 1};
for n = 1, ..., N do
    for k = 1, ..., K − 1 do
        θ^n_{k,0} ~ p_0(θ_{k,0});
        H^n_{k,0} = ∇²_θ ln p_0(θ)|_{θ=θ^n_{k,0}};
    end
end
for t = 1, ..., T do
    Receive context x_t;
    Sample uniformly in {1, ..., N} and obtain corresponding θ̂_1, θ̂_2, ..., θ̂_{K−1};
    Take action k_t = arg max_{k∈{0,...,K−1}} {θ̂_k^T x_t} (θ̂_0 = 0);
    Receive r_t and y_t;
    for n = 1, ..., N do
        z^n_t ~ Poisson(1);
        for k = 1, ..., K − 1 do
            H^n_{k,t} ← H^n_{k,t−1} − z^n_t ∇²_θ ln p(y_t|θ, x_t, k_t)|_{θ=θ^n_{k,t−1}};
            θ^n_{k,t} ← θ^n_{k,t−1} + z^n_t (H^n_{k,t})^{−1} ∇_θ ln p(y_t|θ, x_t, k_t)|_{θ=θ^n_{k,t−1}};
        end
    end
end
```

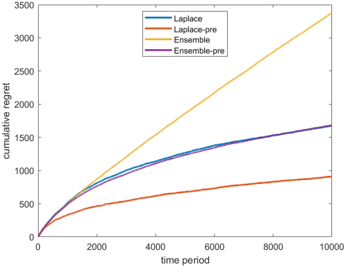
Challenges and Solutions

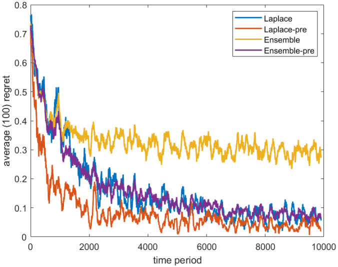
- In semi feedback setting, the likelihood is *log-concave* to θ_{k_t} but not necessarily to $k \neq k_t$ when $y_t = 0$. Update of Precision Matrices may violate the positive definiteness. We propose the following strategy in semi feedback setting:
  - When $y_t = 1$, the log-likelihood is log-concave, and we update as usual.
  - When $y_t = 0$, we only update parameters relative to θ_{k_t}.
- ([**riquelme2018deep**]) Another technique for improving the performance is to *diagonalizing* the *Precision Matrix* (*H* in previous slides) after each time we update it.

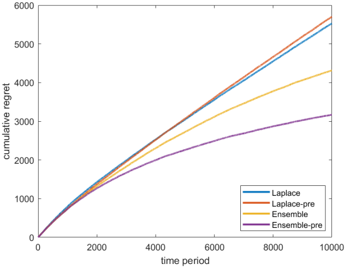
Numerical Experiments

- We conduct a small experiment, with 10 arms and 20 features. Each arm k corresponds to a vector $\theta_k \in \mathbb{R}^{20\times 1}$ sampled from $\mathcal{N}(0, 5I_{20})$.
- The true label of a context $x \in \mathbb{R}^{20\times 1}$ sampled from $\mathcal{N}(0, 10I_{20})$ is obtained by $\mathbb{P}(\text{True label} = k|x) = \frac{\exp(\theta_k^T x)}{\sum_{l=0}^{K-1}\exp(\theta_l^T x)}$.
- In each of our algorithm, we begin by the prior $\mathcal{N}(0, I_{20})$.
- 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 $(t-100, t](t \geqslant 100)$.
- The total period $T = 10000$. All results are averaged 50 times.

Numerical Experiments







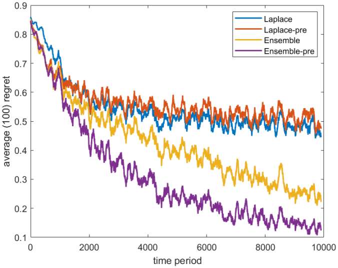


Table of Contents

Recall Our Problem

An easy case: full-feedback with two arms $K = 2$:

- T periods.
- Period t: receive a context x_t (and a hidden binary variable y_t), choose an action a_t from $\{0, 1\}$.
- If $a_t = y_t$: reward 1, $a_t \neq y_t$: reward 0.
- Goal: maximize reward, equivalent to guess y_t!

Logistic Model

- $y_t|x_t, \theta \sim Bernoulli(\frac{1}{1+e^{-\phi_t}})$
- $\phi_t = x_t^T \theta$
- Prior: $\theta \sim N(b, B)$
- Goal: calculate

$$\begin{aligned}
\hat{y}_{t+1} &= \underset{y_{t+1}}{argmax}\, p(y_{t+1}|x_{t+1}, D_t) \\
&= \underset{y}{argmax} \int p(y|x_{t+1}, \theta) p(\theta|D_t) d\theta \\
&\approx \underset{y}{argmax}\, p(y|x_{t+1}, \hat{\theta}_t)
\end{aligned}$$

D_t: all the information in the first t periods, $\hat{\theta}_t$: drawn from $p(\theta|D_t)$.

Data Augmentation

- Objective: Simulating from the posterior, say $p(\theta|D)$.
  - Difficult to sample directly.
- Strategy: Data-augmentation.
  - Auxiliary variable ω.
  - Gibbs sampler $p(\theta|\omega, D)$, $p(\omega|\theta, D)$

- For logistic models, choose ω as a Pólya-Gamma rv.[**Polson2013BI-PG**]
  - $PG(b, 0)$: $\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{Ga(b,0)}{(k-1/2)^2}$
  - $PG(b, c)$: $p(\omega|b, c) = \frac{\exp(-\frac{c^2}{2}\omega)p(\omega|b,0)}{E_\omega\{\exp(-\frac{c^2}{2}\omega)\}}$

Gibbs Sampler

Gibbs sampler:

- Set $\theta^0 = \theta_t$
- For $s = 1, 2, ..., S$

$$\omega_i | \theta^{s-1} \sim PG(1, x_i^T \theta^{s-1}) \tag{3}$$
$$\theta^s | \omega, D_t \sim N(m_\omega, V_\omega) \tag{4}$$

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

$$V_\omega = (X^T \Omega X + B^{-1})^{-1}$$
$$m_\omega = V_\omega (X^T H + B^{-1} b)$$

In the TS model, choose $S = 1$, i.e. sample **once** from the Gibbs sampler is enough!

$K > 2$?

What happens if there are more than 2 arms?

Good news is we can still use PG variables to do data-augmentation if it's full-feedback. The Gibbs sampler is

$$\omega_{ij}|\theta_j \propto PG(K, x_i^T\theta_j - c_{ij})$$
$$\theta_j|\Omega_j \propto N(m_j|V_j)$$

where

$$V_j = (X^T\Omega_j X + V_{0j})^{-1}$$
$$m_j = V_j(X^T(H_j - \Omega_j c_j) + V_{0j}^{-1}m_{0j})$$
$$c_{ij} = \log \sum_{k \neq j} \exp x_i^T\theta_k$$

Why do we use PG?

- Does not appeal directly to the random-utility interpretation of the logit model.
- Exact.
- Requires only a single layer of latent variables.
  - Combining data-augmentation and Gibbs sampler only require us to sample once from the conditional probabilities.[**Albert1993Bayes**]
  - 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.
- It is proved through experiments that PG is more efficient than all previously proposed data-augmentation schemes. [**Polson2013BI-PG**]

Future Work

- Experiments of Pólya-Gamma strategy in the full-feedback case.
- Experiments on practical data.
- Theory and method for semi-feedback problem.
  - Improved Laplace Approximation
  - Improved Pólya-Gamma Data Augmentation
  - ...
- A batch of observations instead of one single observation in each period.

Thank you!