Thompson Sampling in Linear Bandits with exact posterior sampling

Abstract

In this paper, we explore several Thomson Sampling methods for Linear Bandits. In particular we study theory and practice behind Thompson Sampling in a model applied in Display Advertising. In such a case, a model of logistic regression is used. We treat the logistic regression in two main ways : an approximation on the posterior distribution model (Laplace approximation) which leads to a usual Bayesian logistic regression and a Hamiltonian method to draw samples from the true posterior distribution.

1 Introduction

One of the most common algorithm in order to deal with the issue of exploration vs exploitation in linear bandit is Thompson Sampling. This algorithm aims to discover the optimal strategy in order to minimize the regret. It follows a Bayesian approach as the idea behind it is to assume a simple prior over the distribution of the rewards of every arm. Then, at any time, an arm is selected according to its posterior probability to be the best arm. The knowledge of a new batch of data enables to update the posterior. Nevertheless, the posterior probability rarely has closed form and thus, it requires an approximation such as the Laplace approximation in order to be able to compute a posterior distribution.

Another approach in order to avoid the Laplace approximation is to use Markov Chain Monte Carlo. In particular we use the Hamiltonian MCMC. In HMC we use Hamiltonian dynamics as a proposal function for a Markov Chain in order to explore the target (canonical) density $p(\theta)$ defined . It differs from the Metropolis–Hastings algorithm by reducing the correlation between successive sampled states by using a Hamiltonian evolution between states and additionally by targeting states with a higher acceptance criteria than the observed probability distribution. It involves that it converges more quickly to the absolute probability distribution.

In this work, we present some empirical results about the problem of display advertising selection through simulated data. For this situation, we compare three methods, all based on Thompson Sampling. A regular version of this algorithm performs better than an optimistic version of it. Then, we compare those models using the Lagrange approximation to update their distribution to a Thompson Sampling where MCMC are used to sample data.

2 Thompson Sampling Algorithm

We present first the general form of Thompson Sampling and the contextual bandit settings. Let x be the context and A the set of actions which can be chosen. Let r be the reward observed after choosing an action $a \in A$.

Thompson Sampling can be seen with a Bayesian point of view.

The set of past observations D is the triplets (x_i, a_i, r_i) and we introduce a parameter θ such that

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

we can use a parametric likelihood function $P(r|a, x, \theta)$. With a prior distribution $P(\theta)$, the Bayes Formula gives us the posteriori distribution

$$P(\theta|D) \propto \prod P(r_i|a_i, x_i, \theta)P(\theta) \tag{1}$$

We consider that the reward is a stochastic function of the action, context and the unknown true parameter $\theta^\star$. If we knew this parameter, we would like to choose the action that maximize the expected reward $max_a \mathbb{E}(r|a, x, \theta^\star)$. In an exploitation/exploration approach we want to select randomly an action a according to its probability of being optimal. If we draw a random parameter θ at each round, a can be chosen easily (see algorithm 2).

Algorithm 1 Thompson Sampling

```
$D = \emptyset$
for $t = 1, .., T$ do
    Receive context $x_t$
    Draw $\theta^t$ according to $P(\theta|D)$
    Select $a_t = argmax_a \mathbb{E}_r(r|x_t, a, \theta_t)$
    Observe reward $r_t$
    $D = D \cup (x_t, a_t, r_t)$
end for
```

We get an algorithm whose implementation is easy and efficient.

3 Display advertising model

Thompson Sampling can be applied to display advertising problem. The problem is to select the best advertisement for a user so that it clicks on it, according to the context of the user.
More precisely, we want to maximize the CTR (click-through-rate) estimation because in a CPC (cost per click) campaign the advertiser pays only if the user clicks on the ad.
The problem of trade-off exploration/exploitation is present in our example because in order to learn the CTR of an advertisement we must display it, and that can lead to a short term loss.
In a classical dataset about advertising the features contain a lot of information, for example : user,ad,publisher,visited page. All of these features are hashed and being represented by a big sparse vector. In our project we apply the model to synthetic data.
Mathematically, one of the models used in the environment of display advertising is the logistic regression :

$$\mathbb{P}(r_t = 1|x_t, \theta) = \frac{1}{1 + exp(-\theta^T x_t)}$$

$r_t \in 0, 1$ the reward is binary and represent the click or not on the advertisement.
The prior distribution on the weights is a Gaussian, $\theta \sim \mathcal{N}(0, \frac{1}{\lambda} I_d)$.
To use Thompson Sampling, we need the posterior distribution of this model. $\mathbb{P}(\theta|x_1, r_1..x_t, r_t)$
To overcome this problem we explore two main ways.
In a first hand, we use the Laplace approximation to compute the posterior distributions.
In the other hand we use Hamiltonian MCMC methods to draw directly the θ from the true posterior distribution.

3.1 Bayesian Logistic Regression with Laplace Approximation

3.1.1 Laplace Approximation

To simplify, let consider that $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$ The posterior distribution on θ is given by :

$$\mathbb{P}(\theta|r, x) \propto \mathbb{P}(\theta)\mathbb{P}(r|x, \theta) \propto \sigma(\theta^T X)\mathbb{P}(\theta)$$

where $r = (r_1...r_n)$.
Then for new data x_{new}, we can derive the prediction distribution :

$$\mathbb{P}(r_{new}|x, y, x_{new}) = \int \mathbb{P}(r_{new}|r, x_{new})\mathbb{P}(\theta|r, x)d\theta$$

We want to approximate $\mathbb{P}(r_{new}|r, x_{new})$.
The Laplace approximation, that aims to find a Gaussian approximation to a probability density, can be applied. Let recall the concept of Laplace approximation.
We assume that there is a a $f(x)$ where $\int exp(Nf(x))$ has no closed form solution, the Laplace method can find a Gaussian approximation $g(z)$ which is centred on a mode of the distribution $f(x)$, i.e a point x_0 such that $f'(x_0) = 0$.
The Taylor expansion of $f(x)$ on x_0 is : $f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$. As we have $f'(x_0) = 0$, then:

$$f(x) \approx f(x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$$

. Thus:

$$\int exp(-Nf(x))dx = \int exp(-N(f(x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2))dx$$

$$= exp(-Nf(x_0)) \int exp(-N(\frac{1}{2}f''(x_0)(x - x_0)^2))dx$$

$$= exp(-Nf(x_0))\sqrt{\frac{2\pi}{N\left|f''(x_0)\right|}}$$

Then, we can use the Laplace approximation to approximate the posterior $P(\theta \mid r, X)$ as a Gaussian. Therefore, we finally get the following approximation : $P(\theta \mid x_1, r_1, .., x_t, r_t) \sim \mathcal{N}(m, diag(q_i^{-1}))$ where :

$$\theta = argmin_{\theta \in \mathbb{R}^d} \frac{\lambda}{2} \left\|\theta^2\right\| + \sum_{i=1}^{n} log(1 + exp(-r_i\theta^T x_i))$$

and

$$q_i = \sum_{i=1}^{n} x_{j,i}^2 p_j(1 - p_j)$$

where

$$p_j = \frac{1}{1 + exp(-\theta^T x_j)}$$

3.1.2 Regularized logistic regression

In order to select which ads to display, a performing solution is to use the Thompson Sampling algorithm where the prior is defined at each iteration (when we receive a new batch) using the Laplace approximation defined in the previous section.

Algorithm 2 Regularized logistic regression

```
$\lambda$ is defined.                                              ▷ regularization parameter
$m_i = 0$ and $q_i = \lambda$ ( each $w_i$ has a prior $\mathcal{N}(m, diag(q_i^{-1}))$
for $t = 1, .., T$ do
    Get a new batch of training data $(x_1, r_1, .., x_n, r_n)$
    Find w and q as defined in the Laplace approximation section and m = w
end for
```

3.2 Sampling from exact distribution via MCMC

An other approach in order to avoid the Laplace approximation is to use Markov Chain Monte Carlo. In particular we use the Hamiltonian MCMC which draws inspiration from Hamiltonian Dynamics in Physics. We briefly describe here HM-MCMC (for more information refer to [4]) Let define the energy function $\mathbb{E}(\theta)$ such that $p(\theta) = \frac{1}{C}exp(-\mathbb{E}(\theta))$ with C a normalization constant.
The energy function for Hamiltonian Dynamics is a combination of potential and kinetic energies :

$$\mathbb{E}(\theta) = H(x, p) = U(x) + K(p)$$

Starting at an initial state (x_0, p_0), we simulate Hamiltonian dynamics for a short time using the Leap Frog method (a method for numerically integrating differential equations of this form). We then use the state of the position and momentum variables at the end of the simulation as our proposed states variables x^* and p^* The proposed state is accepted using an update rule analogous to the Metropolis acceptance criterion.

Algorithm 3 Hamiltonian MCMC sampling

```
set $t = 0$
generate an initial position state $x^{(0)} \sim \pi^{(0)}$
while $t < M$ do                                        ▷ Drawing $M$ samples
    $t = t + 1$
    sample a new initial momentum variable from the momentum canonical distribution $p_0$
    $x_0 = x^{t-1}$
    run Leap Frog algorithm starting at $[x_0, p_0]$ for L steps to obtain proposed states $x^*$ and $p^*$
    calculate the Metropolis acceptance probability :
    $\alpha = \min(1, \exp(-U(x^*) + U(x_0) - K(p^*) + K(p_0)))$
    draw a random number $u \sim Unif(0, 1)$
    if $u \leq \alpha$ accept the proposed state position $x^*$
    and set the next state in the Markov chain $x^{(t)} = x^*$
    else set $x^{(t)} = x^{(t-1)}$
end while
```

Now that we explain how Hamiltonian MCMC works, we can apply it to our problem to sample the parameter θ.
The steps are described below :

Algorithm 4 Sampling from the posterior distribution with HM-MCMC

```
for $t = 1, .., T$ do
    Get a new batch of training data $(x_1, r_1, .., x_n, r_n)$
    Sampling with HM-MCMC starting from the previous $\theta$.
    Get a set $(\theta_1, .., \theta n)$
    $\theta = \frac{1}{n} \sum \theta_i$
end for
```

4 Implementation and results

We used Python and we have relied on the package PYMC3 and we were inspired by this implementation of the bayesian logistic regression of this github [5].
In a first representation, the features representing the contexts were of dimension 31. We observe 1000 periods. At each period t, the vector of weights θ is updated with the Bayesian Logistic Regression explained previously. Then, we compare the probability for each new context to receive a click : $P = \frac{1}{1+exp(-\theta^T x)}$. Finally, we only selected the 15 contest with the highest probability. After that, we can observe the reward of each of those arms , which provides a new batch of data for the next period.
For this model, we compared the Optimistic TS to the regular TS and quite surprisingly , the regular

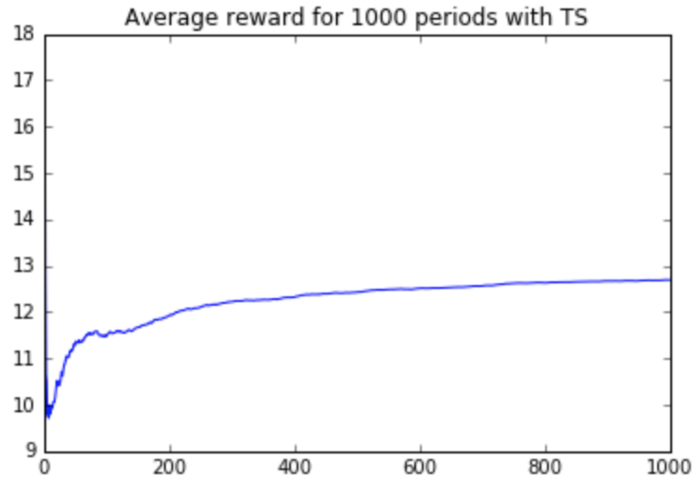

Figure 1: Regular Thompson Sampling

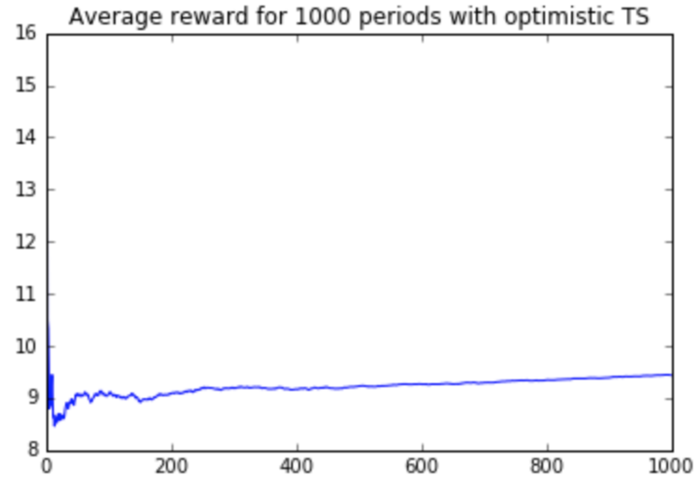

Figure 2: Optimistic Thompson Sampling

one performs far better than the optimistic one. The average rewards are presented on the Figure 1 and 2.

Nevertheless, the computation with the previous setting was too costly in order to to test the MCMC sampling with it (for a reason of time).

For this reason, we decided to decrease the number of features of the context to 7 and the number of periods to 50. We only pick 2 elements per period. In this case, the optimistic an the regular TS are almost equivalent even if the optimistic one performs slightly better.

In this case we can see that the MCMC method is less regular than the two others, so this method performs poorly compared to the two other ones with regards to the performance and the computational time.

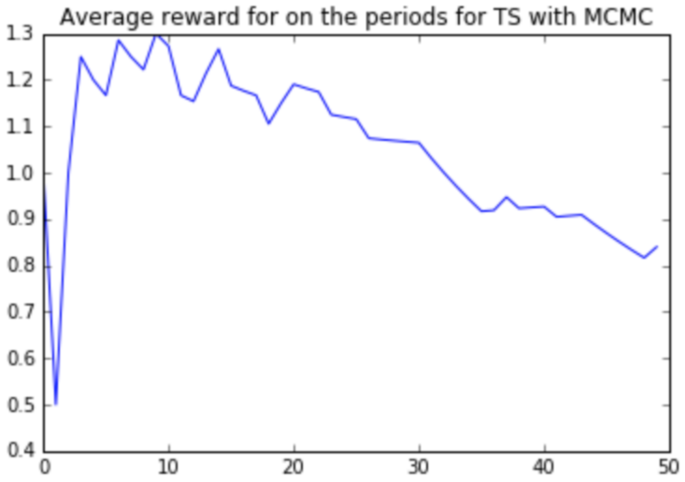

Figure 3: Thompson Sampling with MCMC

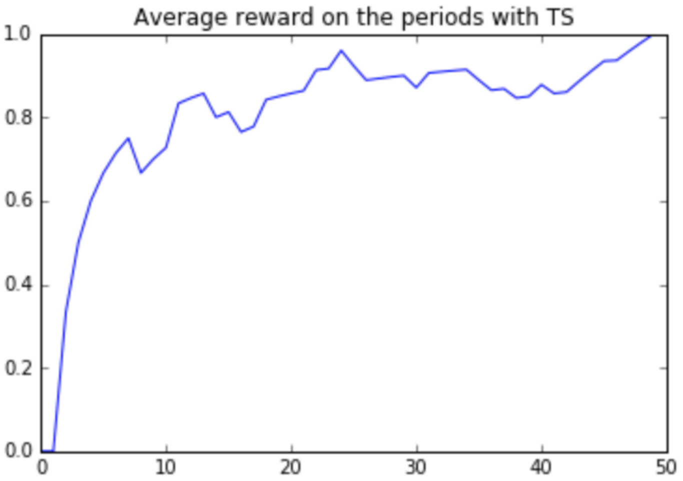

Figure 4: Regular Thompson Sampling

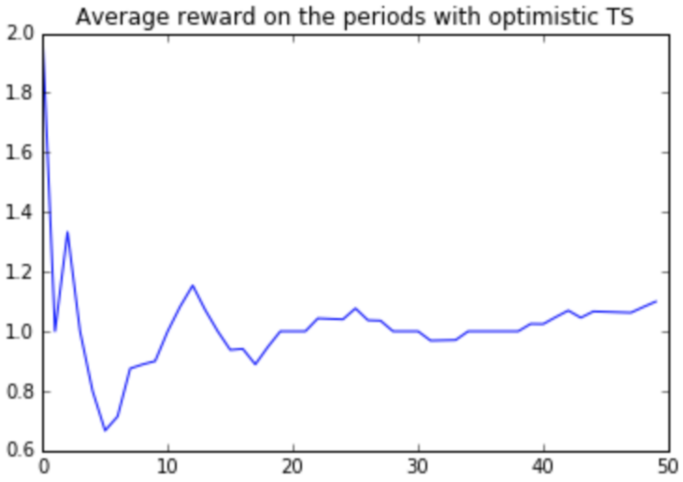

Figure 5: Optimistic Thompson Sampling

5 Conclusion

In this project, we focused on two very interesting mathematical theories but our empirical results have shown that the MCMC methods are harder to implement and might perform badly on real-world problems.

References

[1] B Kulis , CSE 788.04 : Bayesian Modeling and Inference Lectures Notes.

[2] E Kauffman, Thèse : Analyse de stratégies bayésiennes et fréquentistes pour l'allocation séquentielle de ressources

[3] O Chapelle & L Li , An Empirical Evaluation of Thompson Sampling

[4] An explication of the Hamiltonian MCMC
`https://theclevermachine.wordpress.com/2012/11/18/mcmc-hamiltonian-monte-carlo-a-k-a-hybrid-monte-carlo/`

[5] Github on Bayesian Linear regression
`https://github.com/fullflu/bandit`