Notes on A tutorial on Thompson sampling

Bart Frenk

April 20, 2018

1 Tutorial on Thompson sampling

1.1 Introduction

1.2 Greedy decisions

1.3 Thompson sampling for the Bernoulli bandit

1.4 General Thompson sampling

1.5 Approximations

Waiting on background on Laplace approximations.

Approximations are not so useful, since computation time grows with the size of the size of the history.

In order to keep the computational burden manageable, it can be important to consider incremental variants of our approximation methods. We refer to an algorithm as **incremental** if it operates with **fixed** rather than growing per-period compute time. There are many ways to design incremental variants of approximate posterior sampling algorithms we have presented.

As concrete examples, we consider here particular incremental versions of Laplace approximation and bootstrap approaches. (p.19)

1.6 Practical modeling considerations

1.7 Further examples

1.8 Why it work, when it fails, and alternative approaches

2 An empirical evaluation of Thompson sampling

Article published in **december 2011**.

Has the example of optimizing the CTR of an ad display campaign. They use a Laplace approximation of the posterior, and a logistic regression model to relate context to CTRs. Very similar to what we might do. Has quite some references to articles dealing with display ad campaigns.

2.1 Notes

2.1.1 Introduction

> In this work, we present some empirical results, first on a simulated problem and then on two real-world ones: display advertisement selection and news article recommendation. In all cases, despite its simplicity, Thompson sampling achieves state-of-the-art results, and in some cases significantly outperforms other alternatives like UCB. The findings suggest the necessity to include Thompson sampling as part of the standard baselines to compare against, and to develop finite-time regret bound for this empirically successful algorithm. (p.1)

2.1.2 Algorithms

Upper confidence bound (UCB) Strong theoretical guarantees on the regret. There are various variants of the UCB algorithm, but they all have in common that the confidence parameter should increase over time. (p.2)

Bayes-optimal approach of Gittins Directly maximizes expected cumulative payoffs with respect to a given prior distribution.

Probability matching (Thompson sampling) Heuristic (**from 1933**)

Write

$$Q(a, x, r) = \mathbb{I}\Big(\mathbb{E}[r \mid a, x, \theta] = max_{a'} \mathbb{E}[r \mid a', x, \theta] \Big)$$

I find it easier to understand the expression

$$\int Q(a, x, r)\, P(\theta \mid D) d\theta$$

by considering the (hypothetical) case in which there is exactly one θ' such that $Q(a, x, r)$ is 1. The expression then reduces to $P(\theta'|D)$, which is exactly the probability that a is the action with the expected maximal reward.

2.1.3 Simulations

> We can thus conclude that in these simulations, Thompson sampling is asymptotically optimal and achieves a smaller regret than the popular UCB algorithm. It is important to note that for UCB, the confidence bound (1) is tight; we have tried some other confidence bounds, including the one originally proposed in [1], but they resulted in larger regrets. (p.4)

Optimistic Thompson selecting

[1] Weinberger, ea. Feature hashing for large scale multitask learning.

For an action a:

1. Draw θ' from the posterior.
2. Compute the expected reward for a conditional on θ'.
3. Compute the expected reward for a, unconditionally.
4. Take the maximum of step 2. and 3.

This results in a marginally better regret in the simulations. The difference is small.

Posterior reshaping

Widen the variance, by having a posterior with parameters a/α and b/α.

2.1.4 Display advertising

> In this paper, we consider standard regularized logistic regression for predicting CTR. There are several features representing the user, page, ad, as well as conjunctions of these features. Some of the features include identifiers of the ad, advertiser, publisher and visited page. These features are hashed and each training sample ends up being represented as sparse binary vector of dimension 2^{24}. (p.5)

They use the Laplace approximation of the posterior, in the following sense:

1. The Laplace approximation of the posterior is used as an approximation to the prior in the next Bayesian update step.
2. Instead of sampling from the posterior, they sample from the Laplace approximation to the posterior.

I think they use some type of per-period regret, since the regret is decreasing over time. Could not find how long the period was.

Note that clicks are simulated based on weight parameters estimated from real data.

They have 13000 contexts per hour; the number of eligible ads varies between 5,910 and 1, with a mean of 1,364 and a median of 514.

```
return 13000 / 3600
```

```
3.611111111111111
```

I think they use a single logistic regression model, with input features derived from a (context, ad) pair, i.e.,

$$\mathbb{E}[Y|X = x, A = a] = \sigma(h(x, a))$$

1. Look into feature hashing
   Start here: Why feature hashing? They link to an article [1].

3 Extra

3.1 Bounds

Mentioned in **An emperical evaluation of Thompson sampling**.

Chernoff bound
Markov inequality
Chebyshev bound