Thompson Sampling in Adversarial Environments

Travis Dunlop

July 1, 2018

Advisors: Gergely Neu, Mihalis Markakis, Gabor Lugosi

Abstract

Thompson Sampling is an increasingly popular algorithm for decision making in online optimization.

1 INTRODUCTION

William R. Thompson first proposed Thompson Sampling in 1933 as a strategy to estimate treatment effects while minimizing negative outcomes. It has since become a popular algorithm to balance the tradeoff between *exploration* and *exploitation* in repeated games. And, for good reason - Thompson sampling performs empirically quite well. In particular, when the losses are *stochastic* (they come from a fixed distribution over time), Thompson Sampling has guarantees on it's worst case performance. However, what is not well understood, is the performance in *adversarial* environments. It is this question that we try to illuminate.

2 PROBLEM SETUP

We consider a version of the 'prediction with expert advice' framework from the textbook *Prediction, Learning, and Games* [1]. In this setup, there is a forecaster that plays a repeated game against their adversary - the environment.

First, the environment choses a loss for each action and time step $\ell_{i,t}$. Where $i \in \{1, 2, ...N\}$ corresponds with the possible actions of the forecaster and $t \in \{1, 2, ..., T\}$ is the timestep. Then, the forecaster plays the game. For each timestep t: the forecaster chooses an action $a_t \in \{1, ..., N\}$, and incurrs loss $\ell_{a_t,t}$. The losses for all actions are then revealed to the forecaster.

The forecaster tries to learn from the losses it's seen to choose good actions and the environment tries to trick the forecaster into incurring high loss. This setup can be thought of as a mulit-armed bandit problem with full information.

Problem Framework

Parameters: Number of actions: N, Number of timesteps: T
Environment chooses losses $\ell_{i,t} \in [0, 1]$ for $i \in \{1, 2, .., N\}$ & $t \in \{1, 2, ..., T\}$
For each timestep $t = 1, 2, ..., T$

1. Forecaster chooses action $a_t \in \{1, 2, ..., N\}$
2. Environment reveals losses $\ell_{i,t}$ for $i \in \{1, 2, ..., N\}$
3. Forecaster suffers loss $\ell_{a_t,t}$

Now, we need some way of scoring the game between the forecaster and environment. At first glance, a natural choice is the cummulative loss of the forecaster: $\widehat{L} = \sum_{t=1}^{T} \ell_{a_t,t}$. However, this gives too much power to the environment. They could simply maximize loss by choosing $\ell_{i,t} = 1$ for all actions and time steps.

A better choice of metric is cummulative regret. Regret is the difference between the forecasters loss and that of the best fixed action:

$$R_T = \widehat{L}_T - L_T^*$$

Where $L^* = \min_j \sum_{t=1}^{T} \ell_{j,t}$

3 THOMPSON SAMPLING

Thompson Sampling: Beta-Bernoulli

```
Set parameters $\alpha_i = 1$ and $\beta_i = 1$ for all $i \in \{1, ..., N\}$
For each timestep $t = 1, 2, ..., T$
    1. Sample $\theta_{i,t} \sim \text{Beta}(\alpha_i, \beta_i)$ and choose action $a_t = \text{argmin}_i \theta_{i,t}$
    2. Observe losses $\ell_{i,t}$ for $i \in \{1, 2, ..., N\}$
    3. Perform Bernoulli trial for each action: $\widetilde{\ell}_{i,t} \sim Bernoulli(\ell_{i,t})$
    4. Update parameters: $\alpha_i = \alpha_i + \widetilde{\ell}_{i,t}$
                          $\beta_i = \beta_i + 1 - \widetilde{\ell}_{i,t}$
```

4 OTHER ALGORITHMS

4.1 *Follow the Perturbed Leader*

4.2 *Exponential Weighted Forecaster*

4.3 *Follow the Regularized Leader*

5 COMPARISON OF ALGORITHMS

6 EVOLUTIONARY STRATEGIES

REFERENCES

[1] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.