Parameterizing exploration

Author 1
Institution 1

Author 2
Institution 2

Author 3
Institution 3

Abstract

Abstract

1 INTRODUCTION

2 SEQUENTIAL DECISION PROBLEMS

In a sequential decision problem, an agent observes a sequence of world-states; makes a decision in that state; observes a reward that depends on the current state and the action they took; and transitions to the next state. In the most general settings commonly studied in the RL literature, partially observable Markov decision processes (POMDPs), the states transition according to underlying Markov dynamics, and the agent sees only observations that are stochastically related to the underlying state. On the other hand, in the multi-armed bandit (MAB) setting, the state is constant and therefore the rewards depend only on the actions taken at each step — each action corresponds to an "arm" which draws a reward from an underlying distribution when pulled. For simplicity we will consider only MABs, but the extension of our methods to more general settings is conceptually straightforward.

Let $T < \infty$ be the time horizon, k be the number of arms, and $D_1, \ldots, D_k$ be the unknown reward distributions at each arm with means $\mu_1, \ldots, \mu_k$ and maximum mean $\mu^* = \max_i \mu_i$. At each time $t = 1, \ldots, T$, we take an action $A^t \in \{1, \ldots, k\}$ and observe reward $U^t \sim D_{A^t}$. Write the observed sequences of rewards and actions up to time t as $\overline{A}^t = (A^1, \ldots, A^t)$ and $\overline{U}^t = (U^1, \ldots, U^t)$, and define the history until time t to be $\mathbf{H}^t = (\overline{A}^t, \overline{U}^t)$. The goal is to take a sequence of actions such that the expected cumulative regret $\mathbb{E}\left[\sum_{t=1}^{T}\left(\mu^* - U^t\right)\right]$ is minimized.

To achieve this, we would like to identify a learning algorithm which optimally balances exploiting the information in $\mathbf{H}^t$, and gaining information (i.e., in the MAB setting, obtaining more precise estimates of each μ_i). Formally, define a learning algorithm as a sequence $\mathbf{\Gamma} = (\Gamma^1, \ldots, \Gamma^T)$, where for each t, $\Gamma^t : \operatorname{dom} \mathbf{H}^t \longrightarrow \mathcal{S}_{\mathcal{A}}$, where $\mathcal{S}_{\mathcal{A}}$ is the set of probability distributions over the set $\mathcal{A}$ of actions.

We consider variants of three of the most popular learning algorithms. The first is ϵ-greedy: for a decision problem with k actions available at time t, this algorithm takes the greedy (estimated-optimal) action with probability $1 - \frac{\epsilon^t}{k}$, and acts uniformly at random otherwise. The second is upper confidence bound (UCB) exploration, which takes the action which has the greatest upper $100 \times (1 - \alpha^t)$ confidence bound on its true value. The third is Thompson sampling. In classical Thompson sampling for multi-armed bandits, a sample is taken from the posterior distribution over reward means at each arm, and the action which maximizes these posterior draws is taken. However, we can consider a more general class of Thompson sampling algorithms: If C_i is a confidence distribution for the value of action i that belongs to a scale family, we can sample from distributions $\lambda^t C_i$, where $\lambda^t \geq 0$ is a tuning parameter controlling the variance of the distribution from which reward means are sampled and therefore the rate of exploration (as ϵ^t and α^t do in ϵ-greedy and UCB, respectively).

We can write each of these learning algorithms in terms of their respective exploration parameters. For instance, the ϵ-greedy algorithm can be written as

$$\Gamma_{\epsilon^t}(\mathbf{H}^t) = \begin{cases} \arg\max\limits_{i=1,\ldots,k} \bar{U}_i^t, \text{with probability } 1 - \frac{\epsilon^t}{k}, \\ j, \text{with probability } \frac{\epsilon^t}{k} \text{ for } j = 1, \ldots, k, \end{cases}$$

where $\bar{U}_i^t$ is the sample mean of rewards observed from arm i until time t. We write the subscript ϵ^t to emphasize the dependence on the exploration parameter ϵ^t. More generally, we can write a decision rule with

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

a generic exploration parameter η^t as Γ_{η^t}. In the case of ϵ-greedy, $\eta^t = \epsilon^t$; for UCB, η^t represents the confidence level, $\eta^t = 1 - \alpha^t$; and for (tuned) Thompson sampling, η^t is the multiplier of the posterior variance, such that (in the MAB setting) the means at the i^{th} arm are sampled from distributions with standard deviation $\eta^t \sigma_i^t$, where σ_i^t is the posterior standard deviation of arm i at time t.

In each case, it is clear that the optimal sequence $\{\eta^t\}_{t=1}^T$ is nonincreasing in t, as the value of exploring goes to 0 as we approach the time horizon. In the next section, we present a simple strategy for tuning the sequence $\{\eta^t\}_{t=1}^T$ for a given class of learning algorithms.

3 PARAMETERIZED EXPLORATION

In order to adaptively tune the rate of exploration, we propose to parameterize the sequence $\{\eta^t\}_{t=1}^T$ using a family of nonincreasing functions, such that for each t, $\eta^t = \eta(T,t,\theta)$ for some $\theta \in \Theta$. Then, if Γ_η is a decision rule with exploration parameter η, each value of θ leads to a learning algorithm $\mathbf{\Gamma}_{(\eta,T,\theta)} = (\Gamma_{\eta(T,1,\theta)}, \ldots, \Gamma_{\eta(T,T,\theta)})$. We will refer to this algorithm as $\mathbf{\Gamma}_\theta$, suppressing the dependence on the time horizon T and the class of functions η.

Now, let $\widehat{\mathcal{M}}^t$ be an estimate of the generative model $\mathcal{M}$ underlying the sequential decision problem at time t — in the MAB setting, the generative model consists of the reward distributions at each arm, i.e. $\mathcal{M} = (D_1, \ldots, D_k)$. Define $V^T(\theta, M) = \mathbb{E}_{M,\theta}\Big[\sum_{t=1}^T U^t\Big]$ to be the expected cumulative reward until the horizon T if actions are chosen according to $\mathbf{\Gamma}_\theta$ and the true generative model is $\mathcal{M}$. Then, at each time step t we can solve

$$\widehat{\theta}^t = \arg\max_{\theta\in\Theta} V^T(\theta, \widehat{\mathcal{M}}^t), \tag{1}$$

and take the action $A^t = \Gamma_{\eta(T,t,\widehat{\theta}^t)}(\mathbf{H}^t)$.

However, in early episodes, point estimates $\widehat{\mathcal{M}}^t$ of $\mathcal{M}$ may be poor; in particular, if $\widehat{\mathcal{M}}$ simultaneously underestimates the variances of each reward distribution and mis-estimates the ordering of reward means, this method may lead to under-exploration and therefore long sequences of suboptimal actions. In order to account for uncertainty over $\mathcal{M}$, we can instead maximize the expected value of the above objective quantity against a confidence distribution C^t for $\mathcal{M}$ at time t. That is, we can solve

$$\widehat{\theta}^t = \arg\max_{\theta\in\Theta} \mathbb{E}_{\widetilde{\mathcal{M}}\sim C^t} V^T(\theta, \widetilde{\mathcal{M}}) \tag{2}$$

to get $\widehat{\theta}^t$ and associated decision rule at each time.

Algorithm 1 Parameterized exploration for MABs

```
Input Function class $\{\eta(T,\cdot,\theta) : \theta \in \Theta\}$; decision rule class $\{\Gamma_\eta : \eta \geq 0\}$; reward distributions $\{D_i\}_{i=1}^k$; time horizon $T$
$\overline{A}^1 = (1, \ldots, k)$
$\overline{U}^1 = (U_1^1 \sim D_1, \ldots, U_k^1 \sim D_k)$
$\mathbf{H}^1 = \{\overline{A}^1, \overline{U}^1\}$
for t = 1, ..., $T$ do
    Obtain $C^t$ from $\mathbf{H}^t$
    $\widehat{\theta}^t \leftarrow \arg\max_{\theta\in\Theta} \mathbb{E}_{\widetilde{\mathcal{M}}\sim C^t} V^T(\theta, \widetilde{\mathcal{M}})$
    $A^t \sim \Gamma_{\eta(T,t,\widehat{\theta}^t)}(\mathbf{H}^t)$
    $U^t \sim D_{A^t}$
    $\mathbf{H}^{t+1} \leftarrow \mathbf{H}^t \cup \{A^t, U^t\}$
```

4 EXPERIMENTS

To examine the performance of our algorithm, we tested PE versions of ϵ-greedy, UCB, and (a frequentist variant of) Thompson sampling against un-tuned counterparts. Following (Kuleshov and Precup, 2014), we test our algorithms on normal MABs with reward means in $[0,1]$ and different choices of variance. We chose a time horizon of $T = 50$, given the expense of the optimization routine and the fact that significant differences in performance were evident on this scale.

Preliminary results suggested that the objective function in Equation 2 outperformed that in Equation 1, so for a confidence distribution we used the posterior distribution on the reward distributions of each arm. For simplicity we used conjugate priors for the normal mean and variance, with a prior precision of $\tau = \frac{1}{10}$ and gamma hyperparameters $a = b = 0.001$.

For a class of tuning functions, we chose the class of logistic functions $\{\eta(T,t,\theta) = \theta_0\big(1 - \frac{1}{1+\exp[-\theta_2(T-t-\theta_1)]}\big) : (\theta_0,\theta_1,\theta_2) \in \Theta\}$, where the parameter space Θ depended on the class of decision rule. We chose this function class for its balance of being able to flexibly represent exploration decay schedules and the simplicity of the parameter space, which allows the identification of good parameter settings much faster than more complex parameter spaces when using Bayesian optimization.

For each candidate θ, we approximated the expectation in Equation 2 by drawing a generative model $\widetilde{\mathcal{M}} = (\widetilde{\mu}_1, \widetilde{\sigma}_1, \widetilde{\mu}_2, \widetilde{\sigma}_2)$ from the posterior distribution, and simulating one trajectory under $\mathbf{\Gamma}_\theta$ and the generative model $\widetilde{\mathcal{M}}$; the estimated expected value of the policy was computed as the sample mean of the observed cumulative rewards of these trajectories over

Methods	$\sigma = 1$	$\sigma = 0.1$
Tuned ϵ-greedy	**2.35**(0.32)	**0.09**(0.02)
ϵ-greedy ($\epsilon = 0.05$)	4.51(0.59)	0.40(0.07)
ϵ-greedy ($\epsilon = 0.1$)	5.13(0.56)	0.74(0.06)
Tuned TS	**5.17**(0.65)	**0.08**(0.02)
TS	5.20(0.60)	0.27(0.02)
Tuned UCB	**4.62**(0.67)	
UCB ($\alpha = 0.05$)	5.63(0.69)	

Table 1: Mean Cumulative regrets and standard errors (in braces) of different methods for different σ generative models over 96 replicates.

1000 Monte Carlo replicates.

To solve the optimization problem in Equation 2, we used Bayesian optimization as implemented in the Python package `BayesianOptimization`. Bayesian optimization is well-suited to this type of problem, which is characterized by an objective of unknown form and noisy and expensive function evaluations (i.e. the Monte Carlo estimate of the expectation in Equation 2). In order to improve the quality of the optimization routine, we passed initial values of θ which approximately corresponded to the baseline methods to which we were comparing. For instance, in the case of Thompson sampling, the method without tuning corresponds to $\eta^t = 1.0$ (i.e. no shrinkage of the confidence distribution variance) for each t; thus we passed the initial exploration point $\theta = (1.0, 50, 5.0)$, which gives a sigmoid function that is roughly constant at 1.0 on the interval $[0, 50]$.

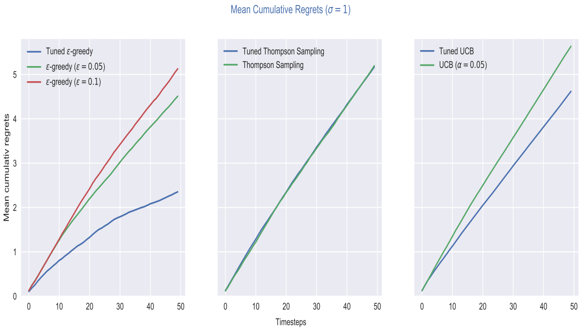
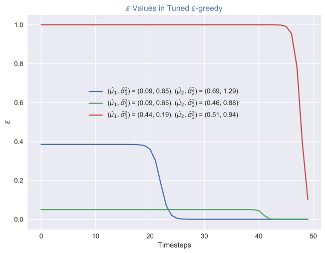
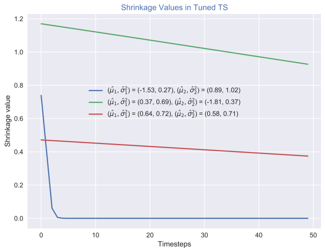


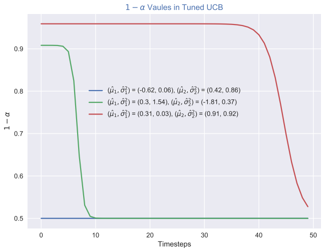
Figure 1: Mean cumulative regrets for PE-tuned learning algorithms and un-tuned variants (96 replicates).

Acknowledgements

Acknowledge someone.







References

References

[1] Kuleshov and Precup, 2014