MB-PSRL: a Scalable Learning Algorithm for Markovian Bandits

Nicolas Gast[1], Bruno Gaujal[1], Kimang Khun[1]

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{nicolas.gast,bruno.gaujal,kimang.khun}@inria.fr

Keyword : *reinforcement learning, Markovian bandits, Gittins index.*

Markov Decision Processes (MDPs) are a powerful model to solve stochastic optimization problems. They suffer, however, from what is called the *curse of dimensionality*, which basically says that the size of the Markov process is exponential in the number of components of the system so that the complexity of computing a solution given the model parameters is exponential. As for existing general reinforcement learning algorithms which try to find a solution when the model parameters are unknown, they all have a regret and a runtime exponential in the number of components, so they also suffer from the same curse.

Very few MDPs are known to escape from this curse of dimensionality. The most famous example is certainly the Markovian bandits problem for which a solution can be computed in $O(n)$, where n is the number of bandits: the problem can be solved by using the Gittins indices which are computed locally (see for example [3]). Hence, we investigate whether reinforcement learning algorithms can also escape from the curse of dimensionality in this problem.

1 Markovian bandits problem and its solution

We have n bandits, each modeled by a Markov reward process with state space $\mathcal{S}_i$ of size S, a reward vector $\mathbf{r}_i \in [0,1]^S$ and a transition matrix Q_i for $i \in \{1,\dots,n\} := [n]$. At time step $t \geq 0$, observing the states of all bandits $\mathbf{x} = (x_i)_{i\in[n]}$, we activate one bandit $a \in [n]$ and receive a discounted random reward $\beta^t R_t$ where R_t is randomly drawn from some distribution on $[0,1]$ with mean $\mathbf{r}_a(x_a)$ and β is the discount factor of the problem. Bandit a transitions to new state y_a with probability $Q_a(x_a, y_a)$ while the unchosen bandits stay in their current state. The objective of the problem is to find a policy $\pi : \mathcal{S}_1 \times \cdots \times \mathcal{S}_n \mapsto [n]$ that maximizes the expected *return* $\mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t R_t\right]$.

When $(\mathbf{r}_i, Q_i)_{i\in[n]}$ are known, Gittins [3] defines the index, later called *Gittins index*, of state $x_i \in \mathcal{S}_i$ for bandit i by $\gamma(x_i) = \sup_{\tau>0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t \mathbf{r}_i(Z_t) | Z_0 = x_i\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t | Z_0 = x_i\right]}$ where Z is a Markov chain whose transitions are given by Q_i and τ can be any stopping time adapted to the natural filtration of $(Z_t)_{t\geq 0}$. It is shown in [3] that always activating the bandit having the largest current index is an optimal policy – *i.e.,* the policy maximizes $\mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t R_t\right]$. Such a policy can be computed very efficiently: The computation of the indices of bandit i can be done in $O(S^3)$ arithmetic operations, which means that the computation of the Gittins policy is linear in the number of bandits as it takes $O(nS^3)$ arithmetic operations.

2 Learning algorithms and regret

When $(\mathbf{r}_i, Q_i)_{i\in[n]}$ are unknown, the reinforcement learning algorithms try to find an optimal policy by interacting with the Markovian bandits, *i.e.,* the algorithms initially output a random policy and collect observations over time to improve their policy and ultimately deduce an optimal policy. The performance of an algorithm is measured by its regret which is the difference between the cumulative reward of the optimal policy and the one of the algorithm's policy. So, the regret quantifies how fast an algorithm can find an optimal policy.

To get enough information for deriving an optimal policy, the algorithms need to *explore* the dynamic of the MDP as much as possible. However, too much exploration hurts the algorithm's performance. So, a good algorithm should also *exploit* the gathered information. Unfortunately, untimely exploitation leads to suboptimal policies, thus unsatisfied performance. This is the famous "exploration vs exploitation dilemma" in the learning problems.

There are two classes of reinforcement learning algorithms. The first class is composed of algorithms that use the celebrated UCB approach known as the *optimism in face of uncertainty* (OFU) principle. OFU methods build a confidence set for the unknown MDP and execute an optimal policy of the "best" MDP in the confidence set, for example UCRL2 [1]. The second class are algorithms that use Bayesian approach, the Thompson sampling method introduced by [5] like PSRL [4]. Such algorithms keep a posterior distribution over possible MDPs and execute an optimal policy of a sampled MDP.

Unfortunately, UCRL2 and PSRL incur $\tilde{O}(S^n\sqrt{T})$ of regret and $O(S^n)$ of computational complexity in our problem where S^n is the state size of the MDP and T is total time. So, both algorithms suffer the curse of dimensionality. Having Gittins index in hand, an interesting question for us is to define an algorithm that escapes from this curse.

3 Our results

In this work, we propose a learning algorithm (called Markovian Bandit Posterior Sampling, or MB-PSRL for short) whose regret is sublinear in time, $\tilde{O}(S\sqrt{nT})$, and whose runtime is linear in the number of bandits, so that it escapes the curse of dimensionality. This algorithm is an adaptation of PSRL [4] for Markovian bandits. We also design an OFU algorithm (MB-UCRL2) adapted from UCRL2 [1]). The upper bound for its regret is similar to the bound for MB-PSRL. The runtime of this optimistic algorithm is however exponential in the number of bandits. We argue that it is likely that no OFU algorithms simultaneously have sub-linear regret in time and linear computation complexity in the number of bandits.

In this paper [2], we report a series of numerical experiments to analyse the performance of MB-PSRL and they confirm the good behavior of MB-PSRL, both in terms of regret and computation complexity.

References

[1] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal Regret Bounds for Reinforcement Learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 89–96. Curran Associates, Inc., 2009.

[2] Nicolas Gast, Bruno Gaujal, and Kimang Khun. MB-PSRL: a scalable learning algorithm for markovian bandits. unpublished.

[3] J. C. Gittins. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, January 1979.

[4] Ian Osband, Daniel Russo, and Benjamin Van Roy. (More) Efficient Reinforcement Learning via Posterior Sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3003–3011. Curran Associates, Inc., 2013.

[5] William R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294, 1933.