An algorithm for the associative reinforcement learning problem

Gábor Bartók, Csaba Szepesvári {BARTOK,SZEPESVA}@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada

In our work we present the algorithm OnLinRel for a variant of the multi-armed bandit problem [2], called *Associative reinforcement learning with linear value functions* [1]. The algorithm achieves a high-probability regret bound of $\tilde{O}(d\sqrt{T})$.

1. The problem

The problem can be described as follows: consider the sequential decision making problem where, at each time step t, an agent has to choose between K alternatives (arms). In addition, a *feature vector* $z_{t,k} \in \mathbb{R}^d, 1 \le k \le K$ is revealed for every arm. After pulling the chosen arm $i(t)$, the agent receives a *reward* according to the expression $r(z_{t,i(t)}) = r_t = \theta^\top z_{t,i(t)} + \sigma_{t,i(t)}$, where $\theta \in \mathbb{R}^d$ is the parameter of the system and $\sigma_{t,i(t)}$ is a zero-mean noise. The agent's goal is to maximize the *cumulative reward* $\sum_{t=1}^{T} r_t$, where T is the time horizon.

Equivalently to the above, one can aim to minimize the *cumulative regret* $R_T = \sum_{t=1}^{T} (\arg\max_k \mathbb{E}[r(z_{t,k})] - r_t)$. For the sake of simplicity, we will use the notation $z_t = z_{t,i(t)}$.

2. The OnLinRel algorithm

The algorithm *Online Upper Confidence Bound for Linear Associative Reinforcement Learning* (OnLinRel), described in Figure 1, computes an upper confidence bound for each arm and pulls the arm with the highest value. For calculating these *ucb*-values, OnLinRel uses a similar idea as that of Auer's LinRel algorithm [1]: instead of estimating the parameter θ, one can directly estimate the expected reward corresponding to a feature vector using the knowledge of previous feature vectors and rewards. The intuitive explanation of the algorithm is the following:

Algorithm OnLinRel
Parameters: T, δ

For each time step $t \le T$ do the following:

1. Receive feature vectors $z_{t,k} \in \mathbb{R}^d \quad (k = 1, \dots, K)$
2. **For each** $1 \le k \le K$
   (i) Calculate vectors defined by the recursion
   $b_{t,1,k} = z_{t,k}$
   $b_{t,s,k} = \frac{(s-1)\|z_{s-1}\|^2 P_{s-1} b_{t,s-1,k} + z_{t,k}}{s\|z_s\|} \quad (2 \le s < t)$
   where $P_i = I_d - \frac{z_i z_i^\top}{\|z_i\|^2}, \quad i = 1, \dots, t-1.$
   (ii) Calculate coefficients
   $a_{t,s,k} = \frac{s}{t-1} z_s^\top b_{t,s,k}, \quad 1 \le s \le t-1$
   (iii) Calculate upper confidence bounds $ucb_{t,k} = \sum_{s=1}^{t-1} a_{t,s,k} r_s + \sqrt{\sum_{s=1}^{t-1} a_{t,s,k}^2} \sqrt{\ln(2TK/\delta)}$
3. Choose the arm with the highest upper confidence bound
4. Receive reward r_t

Figure 1. The algorithm OnLinRel

Suppose that we want to estimate the expected reward corresponding to a feature vector z. If we have a linear combination $z = \sum_{s=1}^{t-1} a_s z_s$ then, with the same coefficients, we can estimate the reward corresponding to z with the linear combination of previous rewards: $r(z) = \sum_{s=1}^{t-1} a_s r_s$.

Furthermore, we want to minimize the 2-norm of the coefficient vector $a = (a_1, a_2, \dots, a_{t-1})^\top$, because our regret bound will depend on this value.

To calculate the coefficients, LinRel uses the Least Squares solution for the equation system $r_s = \theta^\top z_s \quad s = 1, \dots, t-1$. The problem with

this approach is rather technical: for calculating the coefficient for r_s, it uses all the feature vectors $z_1, \ldots, z_{t-1}$. This prevents us from proving the regret bound we aim to achieve. Instead, we can calculate coefficients in an online manner. In the following recursion, the coefficient for r_s is expressed in terms of $z_1, \ldots, z_s$ and z:

$$\begin{aligned} a_{s,s}(z) &= \arg\min_a \| \sum_{i=1}^{s-1} a_{s,i} z_i + a z_s - z \| \\ a_{s,l}(z) &= \left(1 - \frac{1}{s}\right) a_{s-1,l}(z) \quad 1 \le l < s, \end{aligned}$$

and the new feature vector z is approximated by $z \approx \sum_{l=1}^{t-1} a_{t-1,l} z_l$. The above trick of discounting the coefficients of "old" feature vectors keeps the norm of the coefficient vector small while maintaining a sufficient approximation for z. The way the coefficients are computed in Figure 1 can be derived from the above recursion method using basic algebraic manipulations.

3. Our main result

Before presenting the main theorem of this paper we have to introduce a condition on the feature vectors.

Condition 1 Let $\{z_{t,k}\}$ be a stochastic process and $\forall t : k_t \in \sigma(z_{1,k_1}, \ldots, z_{t-1,k_{t-1}})$, the σ-algebra generated by the previously chosen arms. Then, $\exists c < 1$ such that $\sup_{j \in \mathbb{N}} \mathbb{E}\left[\left\| \prod_{i=j}^{j+d-1} P_i \right\| \middle| z_{1,k_1}, \ldots, z_{t-1,k_{t-1}} \right] \le c$ almost surely where $P_i = I_d - \frac{z_{i,k_i} z_{i,k_i}^\top}{\|z_{i,k_i}\|^2}$

Intuitively, Condition 1 means that the expected value of the term above is isolated from 1, independently from the algorithm. Note that the condition trivially holds if the feature vectors are drawn in an i.i.d. manner from a distribution whose support spans $\mathbb{R}^d$. Now we are ready to state our main theorem.

Theorem 1 *Assuming Condition 1 holds, for any $0 < \delta < 1$ the algorithm* OnLinRel *with parameters T and δ achieves the following regret bound with probability at least $1 - \delta$:*

$$R(T) \le \frac{2d\sqrt{2T \ln(2TK/\delta)}}{\min_{1 \le t \le T} \|z_t\|^2 (1 - c)} + \frac{2d}{1 - c}.$$

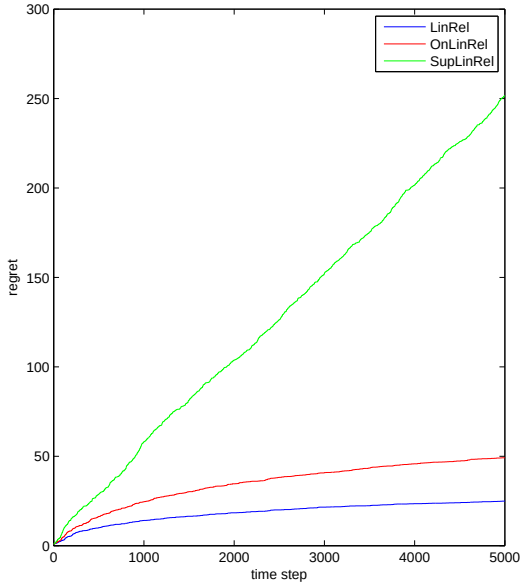

Figure 2. Cumulative regret for three algorithms

This regret bound is slightly worse than that of Auer's SupLinRel algorithm [1], which achieves a regret of $\tilde{O}(\sqrt{dT})$. Nevertheless, as experimental results show, OnLinRel performs significantly better than SupLinRel in the non-adversarial case. Furthermore, one should note that although LinRel achieves smaller regret than OnLinRel, there is no proven regret bound for LinRel to the best of our knowledge. Figure 2 shows the cumulative regret for the three algorithms with parameters $d = 5, T = 5000, \delta = 0.1$ and i.i.d. noise with variance $\sigma = 0.1$.

References

[1] P. Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[2] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin American Mathematical Society*, 55:527–535, 1952.