Inferring bounds on the performance of a control policy from a sample of one-step system transitions

Raphael Fonteneau, Susan Murphy, Louis Wehenkel and Damien Ernst
University of Liège - University of Michigan
{fonteneau, lwh, ernst }@montefiore.ulg.ac.be, samurphy@umich.edu

In financial, medical, and engineering sciences, as well as in artificial intelligence, variants (or generalizations) of the following discrete-time optimal control problem arise quite frequently: a system, characterized by its state-transition function $x_{t+1} = f(x_t, u_t),\ x_t \in X,\ u_t \in U,\ f : X \times U \rightarrow X$, should be controlled by using a policy $u_t = h(t, x_t),\ h : \{0, \ldots, T-1\} \times X \rightarrow U$ so as to maximize a cumulated reward $\sum_{t=0}^{T-1} \rho(x_t, u_t),\ \rho : X \times U \rightarrow \mathbb{R}$ over a finite horizon $T \in \mathbb{N}$.

Different approaches have been proposed for solving this class of problem, such as dynamic programming [1] and model predictive control [2], reinforcement learning approaches [3, 4, 5] or approximate dynamic programming approaches [6] . Whatever the approach used to derive a control policy for a given problem, one major question that remains open today is to ascertain the *actual* performance of the derived control policy [7] when applied to the *real* system behind the model or the dataset (or the finger). Indeed, for many applications, even if it is perhaps not paramount to have a policy h which is very close to the optimal one, it is however crucial to be able to guarantee that the considered policy h leads for some initial states x_0 to high-enough cumulated rewards on the real system that is considered.

Motivated by these considerations, we have focused on the evaluation of control policies on the sole basis of the actual behaviour of the concerned real system. This has lead us to develop an approach for computing a lower bound on the sum of rewards generated by a policy h based on the sole basis of a sample of one-step system transitions $\mathscr{F} = \{(x^l, u^l, r^l, y^l)\}_{l=1}^{|\mathscr{F}|}$. Each one-step system transition provides the knowledge of a sample of information (x, u, r, y), named four-tuple, where y is the state reached after taking action u in state x and r the instantaneous reward associated with the transition. The assumptions under which the approach works are similar to those made usually in the dynamic programming literature when studying problems with infinite state-action spaces: the state and action spaces X and U are normed and the functions f, ρ, and h are Lipschitz continuous.

The approach, which is fully detailed in [8], works by identifying in $\mathscr{F}$ a sequence of T four-tuples $[(x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0}), (x^{l_1}, u^{l_1}, r^{l_1}, y^{l_1}), \ldots, (x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}})]$ $(l_t \in \{1, \ldots, |\mathscr{F}|\})$, which maximizes a specific numerical criterion. This criterion is made of the sum of the T rewards corresponding to these four-tuples ($\sum_{t=0}^{T-1} r^{l_t}$) and T negative terms. The negative term corresponding to the four-tuple $(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})$ of the sequence represents an upper bound variation of the cumulated rewards over the remaining time steps that can occur by simulating the system from a state x^{l_t} rather than $y^{l_{t-1}}$ (with $y^{l-1} = x_0$) and by using at time t the action u^{l_t} rather than $h(t, y^{l_{t-1}})$. Once this best sequence of tuples has been identified - something that can be achieved by using an algorithm whose complexity is linear with respect to the optimization horizon T and quadratic with respect to the size $|\mathscr{F}|$ of the sample of four-tuples - a lower bound on the sum of rewards can be computed in a straightforward way. Furthermore, it can be shown that this lower bound converges at least linearly towards the true value of the return with the density of the sample (measured by the maximal distance of any state-action pair to this sample).

References

[1] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2005, vol. III.

[2] E. Camacho and C. Bordons, *Model Predictive Control*. Springer, 2004.

[3] R. Sutton and A. Barto, *Reinforcement Learning, an Introduction*. MIT Press, 1998.

[4] M. Lagoudakis and R. Parr, "Least-squares policy iteration," *Jounal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.

[5] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, pp. 503–556, 2005.

[6] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[7] R. Schapire, "On the worst-case analysis of temporal-difference learning algorithms," *Machine Learning*, vol. 22, no. 1/2/3, 1996.

[8] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst, "Inferring bounds on the performance of a control policy from a sample of trajectories," in *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (IEEE ADPRL 09)*, Nashville, TN, USA, 2009.