Reinforcement Learning using Kernel-Based Stochastic Factorization

Joelle Pineau, School of Computer Science, McGill University
In collaboration with Andre M. S. Barreto and Doina Precup

Background

Recent years have witnessed the emergence of several reinforcement-learning techniques that make it possible to learn a decision policy from a batch of sample transitions. Among them, *kernel-based reinforcement learning* (KBRL) [6] stands out for two reasons. First, unlike other approximation schemes, KBRL always converges to a unique solution. Second, KBRL is consistent in the statistical sense, meaning that adding more data improves the quality of the resulting policy and eventually leads to optimal performance. Despite its nice theoretical properties, KBRL has not been widely adopted by the reinforcement learning community. One possible explanation for this is that the size of the KBRL approximator grows with the number of sample transitions, which makes the approach impractical for large problems.

Batch Kernel-based Stochastic Factorization

In this work, we introduce a novel algorithm to improve the scalability of KBRL. We use a special decomposition of a transition matrix, called *stochastic factorization*, which allows us to fix the size of the approximator while at the same time incorporating all the information contained in the data. The stochastic-factorization trick is relatively simple. Let $P \in \Re^{n \times n}$ be a transition probability matrix and let $P = DK$ be a factorization such that $D \in \Re^{n \times m}$ and $K \in \Re^{m \times n}$ are stochastic matrices. Then *swapping* the factors D and K yields another transition matrix, $\bar{P} = KD$, with $\bar{P} \in \Re^{m \times m}$, which is potentially much smaller than the original P (i.e. $m << n$). We can show that $\bar{P}$ retains some fundamental characteristics of P [1], such that we can use $\bar{P}$ for many operations. Applying this to MDPs, we replace the original MDP $M = \{S, A, P^a, r^a, \gamma\}$ with a smaller MDP $\bar{M} = \{\bar{S}, A, \bar{P}^a, \bar{r}^a, \gamma\}$, where $\bar{S}$ is a set of representative states (e.g. sampled via trajectories, or from clustering), $\bar{P}^a = K^a D^a$, and $\bar{r}^a = K^a r^a$. So instead of solving an MDP with n states, one solves a model with m states only, and D^a and K^a are built using kernels on those representative states, similar to the original KBLR. We apply this technique to compress the size of KBRL-derived models to a fixed dimension. This approach is not only advantageous because of the model-size reduction; it also allows a better bias-variance trade-off, by incorporating more samples in the model estimate.

The resulting algorithm, kernel-based stochastic factorization (KBSF) [2], is much faster than KBRL, yet still converges to a unique solution. We derive a theoretical

bound on the distance between KBRL's solution and KBSF's solution. We also present experiments on a variety of reinforcement-learning domains, including the double pole-balancing task, a difficult control problem representative of a wide class of unstable dynamical systems, and a model of epileptic rat brains in which the goal is to learn a neurostimulation policy to suppress the occurrence of seizures. We empirically show that the proposed approach is able to compress the information contained in KBRL's model, outperforming both least-squares policy iteration (LSPI) [5] and fitted Q-iteration (FQI) [4] on the tasks studied.

Online Kernel-based Stochastic Factorization

Despite these promising results, we observe that KBSF's memory usage grows linearly with the number of transitions, precluding its application in scenarios where a very large amount of data must be processed. In follow-up work, we show that it is possible to construct the KBSF solution in a fully incremental way, thus freeing the space complexity of the approach from its dependence on the number of sample transitions [3]. The incremental version of KBSF (iKBSF) is able to process an arbitrary amount of data, which results in a model-based reinforcement learning algorithm that can be used to solve large continuous MDPs in on-line regimes.

We present theoretical results showing that iKBSF can approximate the value function that would be computed by conventional kernel-based learning with arbitrary precision by minimizing the distance between sampled states and the closest representative state. This offers useful guidance towards where and when to add new representative states in on-line learning. We empirically demonstrate the effectiveness of the proposed algorithm in the challenging three-pole balancing task, an extension of the well-known double pole-balancing domain. Here, iKBSF's ability to process a large number of transitions is crucial for achieving a high success rate, which cannot be easily replicated with batch methods. We also present unpublished results on additional tasks, including the challenging Helicopter domain.

References

[1] A.M.S. Barreto & M. D. Fragoso. Computing the stationary distribution of a finite Markov chain through stochastic factorization. SIAM J. on Matrix Analysis and Applications, 32, 2011.

[2] A.M.S. Barreto, D. Precup & J. Pineau. Reinforcement Learning using Kernel-Based Stochastic Factorization. *NIPS*, 2011.

[3] A.M.S. Barreto, D. Precup & J. Pineau. On-line reinforcement learning using incremental kernel-based stochastic factorization. *NIPS*, 2012.

[4] D. Ernst P. Geurts & L. Wehenkel. Tree-based batch mode reinforcement learning. *J. Machine Learning Res.*, 6, 2005.

[5] M. Lagoudakis & R. Parr. Least-squares policy iteration. *J. Machine Learning Res.*, 4, 2003.

[6] D. Ormoneit & S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2-3), 2002.