Least-squares policy iteration for reinforcement learning

Lucian Buşoniu

Consider the least-squares policy iteration (LSPI) algorithm for reinforcement learning, described by Lagoudakis and Parr (2003).[1] Write a brief paper about this topic, integrating answers to the following questions in the logical flow of your paper. Do not restrict your reading to this single article (Lagoudakis and Parr, 2003); e.g., some of the questions below may require further reading.

- Dynamic programming is the class of *model-based* techniques to solve Markov decision processes. It can optimally control general, nonlinear and stochastic processes. It poses, however, significant challenges; the most important is the computational complexity of a numerical solution to the problem. What are the additional advantages offered by the *model-free*, *reinforcement learning* class of techniques; and the additional challenges they pose? Which of the two categories does LSPI fit in?
- What is the relation between policy evaluation and policy iteration? Mention other algorithms to compute a solution of a Markov decision process.
- Why is approximation necessary in RL, and in particular, in policy iteration algorithms? Which elements of a classical RL solution are approximated in LSPI, and which elements are *not* approximated?
- The performance of LSPI crucially depends on the predefined basis functions. In view of this, would you agree with the statement that "LSPI has no parameters to tune", made by the authors on page 1129 of the journal publication? Motivate your answer.
- How does it help to consider only discrete actions in the implementation of LSPI? What are the limitations of this choice? Give an example of a control problem where continuous actions are necessary. How would you propose to consider continuous actions within LSPI?

References

Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149.

[1] While reading the article (Lagoudakis and Parr, 2003), you can safely skip some parts (although it is of course not required to skip them). Namely, you can skip the Bellman residual minimization (Sections 5.1 and 5.3), the optimized and model-based policy evaluation (the algorithms of Figures 6 and 7), and Sections 9-10.