Robotic control through model-free reinforcement learning

Hofer Ludovic

1 Introduction

My PhD thesis aims at developping new reinforcement learning algorithms specifically designed to control model-free stochastic systems. This thesis is supervised by Hugo Gimbert and Olivier Ly from Bordeaux University, at the LaBRI. The main targeted application is learning of bipedal walking for low-cost humanoid robots, the experimental platform used is Sigmaban (Passault et al. 2015). Modeling such a task properly involves taking into account the backlash of the reduction gears, the bending of the parts as well as the contact with the ground. Therefore, model-based approaches are not suited to learn such a task. We chose to model this task as a Continuous State and Action Markov Decision Process (CSA-MDP). Since it is hard to predict the shape of the optimal policy, we use value iteration method based on the q-value.

It has already been exhibited that RL algorithms could bring improvements for dynamical tasks requiring a very high accuracy such as the ball-in-a-cup task (Kober and Peters 2009) or hitting a baseball (Peters and Schaal 2008). However most of the reinforcement learning involving robots are based on policy gradient method. While those methods are very effective, they present two major drawbacks: the motor primitive has to be defined by the user and they require the possibility of computing the gradient of the reward with respect to the parameters of the motor primitive. Therefore, applying those methods require a high repeatability of the system and they are limited to a family of solutions specified as a parameter of the algorithm.

Due to the lack of repeatability in low-cost robotics systems, it is quite common to represent them as CSA-MDP. This field has known major breakthrough recently, such as the possibility to find exact solutions when the model is known and has discrete noise, piecewise linear transitions and piecewise linear reward (Zamani, Sanner, and Fang 2012), based on the use of symbolic dynamic programming and extended algebraic decision diagrams (Sanner, Delgado, and de Barros 2012). Although these theorical results are outstanding, they cannot be used to control low-cost robots because of the requirements on the transition and reward functions.

Among the previous work on model-free solvers for MDP, we can note the Least-Square Policy Iteration (LSPI) algorithm (Lagoudakis 2003) which manage to learn hard tasks on problems with continuous state and discrete actions. Although this algorithm lead to satisfying results, it requires expert function approximators adapted to the problem. On the other hand, Fitted Q-Iteration (FQI) (Ernst, Geurts, and Wehenkel 2005) grows regression forests from gathered samples and achieve slightly lower performance than LSPI without requiring custom function approximators. While both methods were initially designed for discrete action choices, Binary Action Search (BAS) (Pazis and Lagoudakis 2009) allows to use them on CSA-MDP.

In order to apply algorithms such as LSPI or FQI to high-dimensional control problems, it is mandatory to use efficient exploration algorithm in order to reduce the number of samples required to learn a near-optimal policy. Optimistic algorithms such as Multi-Resolution Exploration (MRE) (Nouri and Littman 2009) allows to improve the process of collecting samples, while providing guarantees to converge to a nearly-optimal solution.

2 Tools and Methods

As mentioned previously, there is a gap between model-free CSA-MDP methods and robotic applications. Complex tasks such as bipedal walking involves high-dimensional spaces for state and actions. Therefore, it is hard to predict the time required to converge to a near-optimal strategy. Moreover, running experiments directly on robots require human supervision and the manufacturing process is costly and time consuming. In order to run realistic simulations and to make our source code more easy to use, we decided to use ROS[1] and Gazebo[2]. Once learning algorithms lead to satisfying results in simulation, it will be possible to test them directly on the robots.

[1] http://www.ros.org

[2] http://www.gazebosim.org

We base our learning of the q-value on the FQI algorithm (Ernst, Geurts, and Wehenkel 2005), which uses regression forests. While our current implementation of regression forest is based on Extra-Trees (Geurts, Ernst, and Wehenkel 2006), we also plan to test and develop other algorithms growing regression forests.

We compute an approximation of the greedy policy corresponding to the q-value calculated by FQI algorithm using regression forest. This process provides two advantages: firstly, it allows to retrieve actions at a very low computational cost, secondly, by smoothing the discretization noise on the q-value, it also improve the performance of the controler. Real-time constraint is particularly important to ensure that closed-loop control is available.

Currently, exploration is ensured by an algorithm based on MRE (Nouri and Littman 2009). This algorithm is based on the optimistic approach which considers the the couple state-actions which are unknown lead to a maximal reward. It allows to build a knowledge function based on kd-trees, this function provides a result in $[0, 1]$, which is mainly based on the ratio between the density inside the leaf and inside the whole tree. Using this information, collected samples are modified in order to increase the reward if they use an unknown transition or lead to an unknown state. In order to obtain a smoother value function, we use a forest of kd-trees.

While MRE update the policy at a fixed interval of step (chosen by the user), this method leads to an increasing time of update, even if the time required to compute the policy grows linearly with respect to the number of samples, the time required to collect n samples grow quadratically with respect to n. On the other hand, if we wait too many steps before updating the policy, there is a high risk of getting stucked in attracting trajectories. In this case, the collected samples will be redundant and will not improve quickly the knowledge of the MDP. Moreover, on real robots, all the updates to the policy have to be quick enough to ensure that the control frequency can be maintained. In other words, there are no ways to freeze the system in its state. In order to face this issue we plan to develop an algorithm allowing to insert dynamically new samples without needing to restart the learning process from scratch. Another option allowing to increase the space between two consecutive update of the policy would be to detect attracting trajectories.

Another issue relative to solving CSA-MDP is long-term reward. This problem is particularly strong for FQI, since each iteration on the value update involves an approximation. While some problems such as stabilizing an inverted pendulum are harder when the control frequency is lower, a very high frequency can make intractable problems such as inverted pendulum swing-up since it would require to compute the q-value at a very high horizon. We plan to test the effect of including the time during which an action should be applied as one of the dimension of the action.

References

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research* 6(1):503–556.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning* 63(1):3–42.

Kober, J., and Peters, J. 2009. Policy Search for Motor Primitives in Robotics. *Advances in Neural Information Processing Systems 21* 849–856.

Lagoudakis, M. 2003. Least-squares policy iteration. *The Journal of Machine Learning Research* 4:1107–1149.

Nouri, A., and Littman, M. L. 2009. Multi-resolution Exploration in Continuous Spaces. *Advances in Neural Information Processing Systems* 1209–1216.

Passault, G.; Rouxel, Q.; Hofer, L.; Guyen, S. N.; and Ly, O. 2015. Low-cost force sensors for small size humanoid robot. 33405.

Pazis, J., and Lagoudakis, M. G. 2009. Binary action search for learning continuous-action control policies. *Proceedings of the 26th International Conference on Machine Learning (ICML)* 793–800.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21(4):682–697.

Sanner, S.; Delgado, K. V.; and de Barros, L. N. 2012. Symbolic Dynamic Programming for Discrete and Continuous State MDPs. In *Proceedings of the 26th Conference on Artificial Intelligence*, volume 2.

Zamani, Z.; Sanner, S.; and Fang, C. 2012. Symbolic Dynamic Programming for Continuous State and Action MDPs.