Interleaved Quadrature and Optimization for Planning in Continuous State-Action Space

Nishad Gothoskar

LIS

July 27 2018

Value Function: Discrete vs. Continuous

Discrete

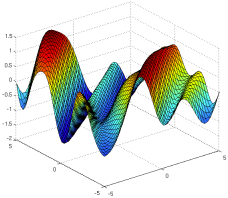
$$V^{\pi}(s) = \max_{a \in A} \Big[\sum_{s' \in S} p_{s'|s,a}(s'|s,a) \big(R(s'|s,a) + \gamma V^{\pi}(s') \big) \Big]$$

Continuous

$$V^{\pi}(s) = \max_{a \in A} \int_{s' \in S} p_{s'|s,a}(s'|s,a) \big(R(s'|s,a) + \gamma^{\Delta t} V^{\pi}(s') \big) \, \mathrm{d}s'$$

In the continuous setting, a value update consists of a maximation over an integral. More specifically, this is a maximization in a continuous action space and an integral of the product of a continuous state space value function and transition model.

Gaussian Process in RL



As opposed to discretizing the state space, we can use a Gaussian Process to model the value function across the entire domain. Previous work in [1][2][3] has presented approaches to using GPs as value functions and learning a policy for a simple reinforcement learning problem (mountain car/swinging pendulum).

However, these approaches do not scale well in higher dimensional or larger state spaces and are only designed to handle transition models with small Gaussian noise.

Background

Gaussian processes output a distribution of value at a state, which allows us to explicitly model our uncertainty about its value.

$$y^*|\mathbf{y} \sim \mathcal{N}(K_* K^{-1}\mathbf{y}, K_{**} - K_* K^{-1} K_*^T)$$

In addition, when using Gaussian kernels for the GP and assuming Gaussian or mixture of Gaussian transition models, the following integral will have a closed form solution (mean and variance). By taking this into account when designing an action selection policy, we can tune the "hyperparameters" (exploration, risk aversion) of our final policy.

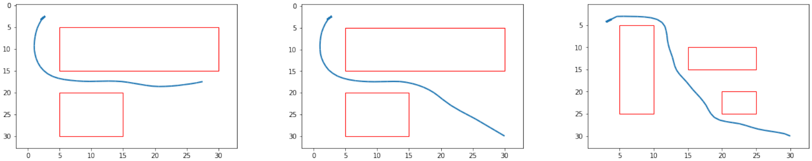
$$\int_{s' \in S} p_{s'|s,a}(s'|s,a)\big(R(s'|s,a) + \gamma^{\Delta t} V^{\pi}(s')\big)\ \mathrm{d}s'$$

Problem Statement

In this work, we present a framework for planning in stochastic continuous state-action spaces. We use a GP to model the value across the entirety of the state space. We use this model to actively select states to update in each iteration, therefore extending previous methods to be feasible in large high-dimensional state spaces. In addition, using the previously mentioned quadrature techniques paired with a learned transition model, we estimate the value and uncertainty of taking certain actions, aiding in the action selection procedure. We show this method of active state selection and interleaved quadrature and optimization allows us to learn robust policies efficiently. We evaluate these policies on a navigation/obstacle avoidance task in simulation as well as a block pushing scenario on the PR2.

Current Progress

In the current implementation we use a 30x30 grid as the support points for our GP. Our reward function has high value at the goal states and incurs penalty for leaving the grid or hitting the obstacles. Then we perform value iteration to converge to an optimal policy. Below are some paths in our navigation scenario, following a learned policy.



Next Steps

- Use Quadrature rules for GP and Gaussian/Mixture to estimate an action's value (mostly done)
- Implement active learning to select states to update in value iteration
- Continuous action optimization with BO (Zi's paper) but incorporating mean and variance of quadrature

Bibliography

[1] C. Rasmussen and M. Kuss. Gaussian Processes in Reinforcement Learning. In Proceedings of the International Conference on Neural Information Processing Systems, pages 751–759. MIT Press,2004
[2] M. P. Deisenroth, J. Peters, and C. E. Rasmussen, “Approximate dynamic programming with Gaussian processes,” in Proc. of the IEEE American Control Conference (ACC), 2008, pp. 4480–4485.
[3] Rasmussen, C.E., Ghahramani, Z.: Bayesian Monte Carlo. In Becker, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems. Volume 15. MIT Press, Cambridge, MA (2003)