STA571: Advanced machine learning

Optimizing Trust Region Policy Optimization (March 17, 2017)

Authors: Yixin Lin

1 Motivation

Trust region policy optimization (or TRPO) is a reinforcement learning algorithm which is a policy gradient approach that uses natural gradients. Though it has proven both successful on a wide variety of tasks and scalable to large amounts of policy parameters, one issue that it (and policy gradient approaches in general) is that it is highly sample inefficient. This means that though there are theoretical convergence guarantees, in practice it may take a long time to reach convergence and require many simulations.

2 Problem definition

Trust region policy optimization is a reinforcement learning algorithm which attempts to optimize a policy $\pi(a|s)$ in order to maximize the standard Q-function $Q^\pi(s,a) = r + \gamma \max Q^\pi(s',a')$ and it does this by assuming a parameterized policy π_θ, iteratively improving the policy by optimizing the parameters subject to a maximum KL divergence which is controlled by a hyperparameter δ. This is then approximated by linearization and a Monte Carlo sampling approach.

Our goal is to optimize this using method by using ideas from Bayesian optimization of hyperparameters, e.g. in the Snoek paper we read in class, to optimize the hyperparameters used in each iteration. Specifically, we have the hyperparameters δ as well as the number of timesteps of data per iteration which can both be optimized every iteration in the way described in the paper.

3 Models and methods

We will estimate the optimal policy through the TRPO algorithm, but apply the Gaussian process Bayesian hyperparameter optimization which ideally will result in higher convergence rates. The main issue in TRPO is simply the time and iteration speed, and so a proper Bayesian treatment of hyperparameter optimization should produce better results. The parameters which we estimate will define the best policy we can get on the problem.

4 Results and validation

We will show the learning curves of the algorithm over repeated iterations by comparing to the original TRPO algorithm benchmarks on various reinforcement learning environments. Because of the high variance of the potential results due to the reinforcement learning setup, we will train and validate repeatedly in order to reduce the variance in our results.