Actor-Critic Algorithms

The material I talked about in class is mostly from these papers (I have linked these papers on the class web page):

1. Policy Gradient Methods for Reinforcement Learning with Function Approximation. Rich Sutton, David McAllester, Satinder Singh and Yishay Mansour, NIPS 2000. - This paper give a easier to follow proof of the theorem I talked about in class. But the discussion lacks in motivation in general.

2. Actor-Critic Algorithms. Vijay Konda and John Tsitsiklis, NIPS 2000. - This is paper assumes the theorem from an earlier thesis, and develops the intuition for the form of the function approximator for the critic. It also presents a family of algorithms that use the results. What follows is from this paper.

Average Reward TD Actor-Critic Algorithm using Function Approximation

This writeup is not an exhaustive description of what was presented in class. I just describe one family of algorithms that uses the ideas presented. The title above says it all. We are looking at an Actor-Critic algorithm, that uses a policy gradient approach. We use the average reward criterion.

The policy is directly represented using a set of parameters. The parameters could be preferences, as we have seen earlier, or could be thresholds as discussed in class. A better choice might be to use one linear function approximator for each action that computes the preference for that action.

The critic, whose output is used in computing the gradient information for the actor, is represented by a separate linear function approximator.

Let $\Theta = \{\theta^1, \cdots, \theta^n\}$ be the parameters of the actor. Let $\Phi_\Theta = \{\phi^1_\Theta, \cdots, \phi^m_\Theta\}$ be the features of the critic. Note that the features are dependent on Θ. The dependence is explained below.

Consider the space spanned by the ϕs. This is the space comprising of

all the vectors that you can construct by linear combinations of the ϕs. Let

$$\psi_{\theta_i}(s,a) = \frac{\partial \pi(s,a;\Theta)}{\partial \theta_i} \frac{1}{\pi(s,a;\Theta)}.$$

Note that each ψ (there are n of them) can be represented as a vector of length $|S| \cdot |A|$, with a component for every pair (s,a). Similarly each ϕ_i can be represented as a vector of the same length. The conditions for convergence requires that the space spanned by the ϕs should contain the space spanned by the ψs. One straight forward way of achieving this is to set $m = n$ and choose $\phi^i_\Theta = \psi_{\theta_i}$. But this choice might be unstable in some boundary conditions and it is better to set $m > n$ and choose the features to achieve strict inclusion.

Let $\Omega = \{\omega_1, \cdots, \omega_m\}$ be the parameters of the critic. Now the action value function corresponding to parameters Θ is computed as follows:

$$Q^\Theta(s,a;\Omega) = \sum_{j=1}^{m} \omega_j \phi^j_\Theta(s,a) \tag{1}$$

Along with the parameter vector Ω, the critic has the following auxiliary parameters:

1. A scalar estimate ρ of the average reward.
2. A m-vector e which represents the eligibility trace.

At time t let the system be in state s_t and apply action a_t. Let s_{t+1} be the resulting state and r_{t+1} the resulting reward. Let a_{t+1} be the action the actor picks to apply in s_{t+1} according to its current policy. The following updates are carried out:
(i) **The Critic:**
First the parameters Ω are updated:

$$\Omega_{t+1} = \Omega_t + \alpha_k \left(r_{t+1} - \rho_t + Q^{\Theta_t}(s_{t+1}, a_{t+1}; \Omega_t) - Q^{\Theta_t}(s_t, a_t; \Omega_t) \right) e_t \tag{2}$$

Then the average reward estimate is updated:

$$\rho_{t+1} = \rho_t + \alpha_t \left(r_{t+1} - \rho_t \right) \tag{3}$$

where α_t is a positive stepsize parameter. There are many variants of the critic that differ in the way they update the eligibility trace. I present just

one here. Refer to the second paper above for other variants. This version is known as the $TD(1)$ critic since it tries to estimate the MC average return:[1]

$$e_{t+1} = \Phi_{\Theta_t}(s_{t+1}, a_{t+1}), \qquad \text{if } s_{k+1} = i^\star \tag{4}$$

$$e_{t+1} = e_t + \Phi_{\Theta_t}(s_{t+1}, a_{t+1}), \qquad \text{otherwise} \tag{5}$$

Here $i^\star$ is some designated recurrent state.
(ii) **The Actor:**
The actor updates it parameters as follows:

$$\Theta_{t+1} = \Theta_t + \beta_t \, \Gamma(\Omega_t) \, Q^{\Theta_t}(s_{t+1}, a_{t+1}; \Omega_t) \, \Psi_{\Theta_t}(s_{t+1}, a_{t+1}) \tag{6}$$

Here β_t is a positive step size parameters and $\Gamma(\Omega_t)$ is a normalization factor satisfying the following conditions: (i) $\Gamma(\cdot)$ is Lipschitz continuous and (ii) there exists $C > 0$ such that $\Gamma(\Omega) \leq \frac{C}{1+\|\Omega\|}$. These conditions can be satisfied by choosing Γ to be a small positive constant for all values of Ω.

There are other conditions on the various parameters to ensure theoretical convergence. The one of practical significance is that the critic should be updated at a much faster rate than the actor. In other words, α_t should be larger than β_t.

This writeup describes an update rule for the actor and the critic. There are a lot of parameters of the algorithm, such as the feature vectors and the stepsizes, that need to be chosen. I have mentioned some general guidelines, but suitable settings for these parameters depends on the problem domain. Unfortunately there are not many applications of this approach out there from which to draw examples.

[1] MC average return is the only one we have talked about in class. Something similar to $TD(\lambda)$ returns for average rewards also exists and the variants use them.