Review of Trust Region Policy Optimization

Guannan Hu

March 21, 2018

Most algorithms for policy optimization can be classified into three broad categories:

- policy iteration methods, which alternate between estimating the value function under the current policy and improving the policy;
- policy gradient methods, which use an estimator of the gradient of the expected return obtain from sample trajectories;
- derivative-free optimization methods, such as the cross-entropy method (CEM)and covariance matrix adaptation (CMA), which treat the return as a black box function to be optimized in terms of the policy parameters.

1 The Cross-Entropy Method

The Cross-entropy (CE) [1] method is a general algorithm for (approximately) solving global optimization tasks of the form

$$w^* = \arg\max_w S(w) \tag{1}$$

where S is a general real-value objective function, with an optimum value $\gamma = S(w^*)$. The main idea of CE is maintain a distribution of possible solutions and update this distribution at each step.

The CE method starts with a parametric family of probability distributions $\mathcal{F}$ and an initial distribution $f_0 \in \mathcal{F}$. Under this distribution, the probability of drawing a high-value sample (having value near γ^*) is presumably very low; therefore, finding such samples by naive sampling is intractable. For any $\gamma \in \mathcal{R}$, let $g_{\geq\gamma}$ be uniform distribution over the set $\{w : S(w) \geq \gamma\}$. If one finds the distribution $f_1 \in \mathcal{F}$ closest to $g_{\geq\gamma}$ with regard to the cross-entropy measure, the f_0 can be replaced by f_1 and γ-valued samples will have larger probabilities. For many distribution families, the parameters of f_1 can be estimated from samples of f_0. This estimation is tractable if the probability of the γ-level set is not very low with regard to f_0. Instead of the direct computation of the $\mathcal{F}$-distribution closest to $g_{\geq\gamma^*}$, we can proceed iteratively. We select a γ_0 appropriate for f_0, update the distribution parameters to obtain f_1, select γ_1, and so on, until we reach a sufficiently large r_k. Below we sketch the special case when w is sampled from a member of gaussian distribution family.

Let the distribution of the parameter vector at iteration t be $f_t \sim N(\mu_t, \delta_t^2)$. After drawing n sample vector $w_1, ..., w_n$ and obtaining their value $S(w_1), ..., S(w_n)$, we select the best $\lfloor \rho \cdot n \rfloor$ samples. where $0 < \rho < 1$ is the selection ratio. This is equivalent to setting $\gamma_t = S(w_{\lfloor \rho \cdot n}$). Denoting the set of indices of the selected samples by $I \in \{1, 2, ..., n\}$, the mean and the deviation of the distribution is updated using

$$\mu_{t+1} := \frac{\sum_{i \in I} w_i}{|I|} \tag{2}$$

and

$$\delta_{t+1}^2 := \frac{\sum_{i \in I} (w_i - \mu_{t+1})^T (w_i - \mu_{t+1})}{|I|} \tag{3}$$

To preventing early convergence, it adapt a trick frequently used in particle filtering: at each iteration, it adds some extra noise to the distribution: instead of equation 3, it used

$$\delta_{t+1}^2 := \frac{\sum_{i \in I} (w_i - \mu_{t+1})^T (w_i - \mu_{t+1})}{|I|} + Z_{t+1} \tag{4}$$

where Z_{t+1} is a constant vector depending only on t.

References

[1] I Szita and A Lrincz. Learning tetris using the noisy cross-entropy method. Neural Computation, 18(12):2936, 2006.