Reinforcement Learning, a derivation

October 17, 2019

Suppose you are trying to learn good parameters θ for a distribution $p_\theta(x)$ such as to maximize a reward function $f(x)$. As an example, x can be an action taken by a robot and $f(x)$ is a reward for actions.

In other words, we'd like to maximize

$$\mathbb{E}_{x\sim p_\theta(x)}[f(x)] \tag{1}$$

In other words, we would like to maximize the *expectation* of the reward over our choices of actions. We could use various algorithms to compute $p_\theta(x)$, such as a deep net (then θ is the set of weights of the neural net). To optimize the objective (1) with gradient descent, we would need to compute

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)}[f(x)] \tag{2}$$

At first this seems untractable; how do we compute the derivative of an expectation with respect to the parameters of the expectation's distribution?

0.1 Exercise 1

Prove the following theorem

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)}[f(x)] = \mathbb{E}_{x\sim p_\theta(x)}[f(x)\nabla_\theta \log p_\theta(x)] \tag{3}$$

First, use the Leibniz integral rule (note the case where the bounds of integration are constants, such as $-\infty, \infty$). Assume p, f are continuous as are all of their derivatives. Next, consider the properties of the derivative of logarithms.

We are aware that this is a well known equation and you may have seen the derivation before: for this exercise, please carefully justify each step and explain precisely why it is true.

0.2 Exercise 2

Suppose that at a particular time in our training process we have some parameters θ. How can we approximate

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)}[f(x)] \tag{4}$$

(i.e. show a consistent estimator; use the theorem!)

0.3 Exercise 3

Based on your estimator, explain in words the "intuition" for why it works for finding good parameters θ (i.e. give an explanation for why it "works" that may convince someone without the mathematical derivation).

0.4 Exercise 4

At first glance, our derivation isn't obviously useful. Normally we aren't interested in taking actions x with no context; rather, we want to take a good action x given a scenario y. Our reward function can be written $f(x, y)$ and we want to learn a distribution $p_\theta(x|y)$.

Is our derivation still useful? If not- what breaks? If so- how does the estimator change?