Exploration complexity

Question 1. How does efficient exploration depend on the learning mechanism used and the topology of the environment?

Random search takes $O(n^k)$ steps to achieve ϵ uniform coverage. How to define exploration complexity? Regret versus random? Number of steps to achieve approximately max entropy $\epsilon > H(\pi) - log(n)$.

We are going to explore intrinsic motivation as a solution to efficient exploration. Specifically, using negative density estimates as the reward, the surprise.

$$r_t = p(s_t) \tag{1}$$
$$r_t = p(s_t \mid s_{t-1}) \tag{2}$$
$$r_t = p(s_t, a_t) \tag{3}$$
$$r_t = p(s_t, a_t \mid s_{t-1}, a_{t-1}) \tag{4}$$
$$r_t = 1[p(s_t \mid s_{t-1}) > 0.001] \quad \text{(in memory novelty)} \tag{5}$$

(also there are many ways we could decompose these probabilities?? $p(s_t, a_t) = p(s_t)p(a_t)$. Sparse vs dense rewards, …?)

- What are the limits of intrinsic motivation (and various densities / rewards)? What problems can it solve? How much resources (data) does it need?
- What properties does the density function need to have, it we want to efficiently explore environments with certain exploitable structure?

To prove

- Prove that our algorithm is unstable. It does not get stuck.
- The exploration complexity is $O(n)$ steps to achieve ϵ uniform coverage!?

Metrics

Ok, firstly. What does it mean to explore efficiently? Coverage, speed of diffusion, generalisation, …?

A natural measure of coverage is entropy (?).

$$L_{entropy} = \mathbb{E}[-log(p(s,a))] \tag{6}$$

Without a reward fn, all states are equally valuable. (but if we were given a reward fn, how could we tune this?)

Ok, with enough iterations. Random search will achieve uniform coverage! We care about achieving this faster!

Want to measure the probablity that a state (-action) will be reached after n steps.

$$p(s_j \mid n, s_i) = ??? \tag{7}$$
$$L = \sum p(s_j \mid n, s_i) \tag{8}$$

Density fn

Two cases, tabular, fn approximation.

In the fn approximation case, we also care about generalisation. But how do we measure this? Also, want a motivating example.

Steps

1. What does a random policy look like? And what is its complexity?
2. Given arbitrary π, G. Visualise how π diffuses over G.
3. Adapt π via a learning algol, A and intrinsic motivation.
4. ???

Learning dynamics

(closely related to solving diffusion systems. should look into?)

$$\frac{ds}{dt} = ??? \tag{9}$$
$$\frac{d\xi}{dt} = \text{(normalised count)}$$
$$R(t) = (I - \gamma P_{\pi(t)})^{-1} \cdot \xi(t) \tag{10}$$
(11)

Learning algols

$$\begin{aligned} \frac{d\pi}{dt} &= E[\log \pi \cdot R(t)] && \text{(policy grads)} \\ &= softmax(R) && \text{(Q-learning)} \end{aligned} \tag{12}$$

Thoughts

- The most valuable node should be the most central node? (well depends on the policy / learning algol$)

Background: Random walks on graphs

Given an adjacency matrix, A (where $A_{ij} = 1$ if nodes i, j are connected, else 0). Then let P be the transition matrix, where $P = \frac{A}{\sum_i A_i}$ (the rows have been normalised).

Then the distribution following a random walk starting from e_i for k steps is $x(e_i, k, P) = P^k e_i$. We want to know, $x(e_i, \infty, P)$ and how different that distribution is from uniform, $\epsilon = 1/n - x(e_i, \infty, P)$. And also, how quickly it converges with the number of steps taken. Let $c(k) = x(e_i, \infty, P) - x(e_i, k, P)$.

Eigenvectors etc.

Resources

- Provably Efficient Maximum Entropy Exploration