Modular Multitask Reinforcement Learning via Policy Sketches

Matthew Feng

September 27, 2018

1 Important Citations

In your opinion, what are the 1-2 citations that are most important to this paper? Why?

I think two citations that are particularly important to the paper are *Sutton, 1999* on the options framework and *Parr & Russell, 1998* on Hierarchical Abstract Machines. I have not yet been able to read the two papers in full, but because they are mentioned multiple times throughout the paper, and provide the foundations that policy sketches builds upon, I believe they are very important. In particular, the paper mentions that policy sketches are an "instantiation of the options framework", and so to fully appreciate policy sketches I must understand options.

2 Strengths and Improvements

What do you like about this paper? What do you think could be improved?

The paper explained the body of related work and the distinctions of the current work very well; I did not know much about modular and multi-task reinforcement learning before the paper, but the paper gave a very clear overview. Additionally, the paper explained the problem and model very clearly, as well as the motivations for developing the model the way they did.

One thing that could be improved about the paper is the introduction and discussion about the *adaptive learning*. In particular, I was not sure why the modular learning framework should make adaptation learning any more easier than other RL techniques; they do not seem too related to or correlated with each other (on the other hand, I can see how the zero-shot learning task is quite relevant).

Additionally, the training procedure, in particular the math behind the task-specific critic scores, was not too clear in the explanation and so I still have questions about how to implement that part of the training process myself.

3 Curriculum Learning

What is the role of curriculum learning?

Curriculum learning is a training scheme that solves two problems: (1) overly complex tasks will result in the agent never recieving any positive reward unless subpolicies have already been learned; and (2) training too long on simple tasks may cause overfitting and reduce applicability of the subpolicies to other more complex tasks τ. Additionally, spending too much time on easy tasks is wasteful.

4 Random Walks

Look at the code to find the size of the map in the crafting environment. Suppose you start in the middle of the grid and an object is in a corner; how many steps do you need to get there if you act optimally? Give a rough estimate of the expected number of steps you would need to get there by moving randomly.

The size of the grid is 10×10, so being in the center would be at position `[4, 4]` on the board. We can reach the goal at `[9, 9]` optimally in **10 steps**: go straight right, then straight up.

A rough estimate for the number of steps necessary to reach the goal with random steps is 400 steps, using the fact that a random walk reaches a distance of $\sqrt{n}$ after n steps. Since we need to travel a distance of 10, we would need 100 steps. However, there are four different corners we could end up in, so we would need about $4 \times 100 = 400$ steps the reach all the corners.

Running a quick simulation over 50000 trials yields an average of 499 steps.

5 Hypothesize

In the cliff environment, imagine you have mastered actions North, East, West but have never seen the 'South' primitive. Now you receive a policy sketch which also contains action 'South', which of the following is true and why?

(a) North-South-East-West is faster to learn
(b) South-North-West-East is faster to learn
(c) Both equally are equally easy to learn

I believe policy sketch (b) should be will learn faster. Because the unmastered skill appears first in the sequence, the agent will spend more time in subpolicy π_{South} because it has not yet mastered it, which means the estimates for the parameter updates will be more accurate. In other words, even though the *North* task is mastered, it is possibl that the ending of the *North* task leaves the agent in a poor position for the *South* task, which would couple the rewards for π_{South} with the *North* task, resulting in a longer training time to determine the true reward signals.