Project Update for Replication Review for Coordinated Exploration in Concurrent Reinforcement Learning

Ed Fancher [1]

Abstract

This is a replication attempt for a a paper on seed sampling. Seed sampling is a way to provide a stable and efficient mechanism for exploring the state action space for multi-agent reinforcement learning algorithms. In the replicated paper, 3 exploration methods are compared, using a common multiple agent learning algorithm : Upper Control Bounds, Thompson Sampling, and Seed Sampling.

1. Introduction

There are multiple approaches to exploration in single agent settings. These approaches may not always extend well to multiple agent settings. There are 2 single agent methods which have had some success: Upper Control Bound (Auer et al., 2009) and PSRL (Bayesian) methods (Strens, 2000) In this project, I attempt to replicate the findings in (Dimakopoulou & Roy, 2018). In this paper 3 approaches are compared: UCB based multiple agent approaches, Thompson Sampling and Seed Sampling. Seed Sampling is based on the single agent PSRL approach. In the UCB approach, as presented in the paper, the agents do not coordinate directly on exploration. Instead, they maintain a shared model and at each time step will calculate the best approach based on that model using an upper control bound.

In the Thompson Sampling approach, at each time step, a potential MDP is constructed on the transitions and/or Rewards, based on transitions/rewards seen up to this point. Each agent then samples independently from the MDP to take the next step. In seed sampling, an MDP is constructed at the beginning of each episode, similarly to Thompson Sampling, and this is used to generate a policy for the episode. So the main difference between Thompson Sampling and Seed Sampling is that seed sampling samples per episode, and Thompson Sampling samples per time step.

An important consideration is that, in terms of the paper's construction of the problem, an MDP (could be an approximation) must be solved for all three methods. For UCB and Seed Sampling, it's done once per episode, for Thompson Sampling it's every time step.

Each set of three approaches is tested on three simple problems.

1) A bipolar chain, constructed as $-10 \longleftrightarrow -1 \longleftrightarrow$ ... $\longleftrightarrow -1 \longleftrightarrow 10$. The chain can be reversed. So there are 3 rewards: -10, 10, -1 and 2 actions: left, right. All transactions are deterministic. The -10 and 10 states are also stopping states.

Here is an image from the paper showing a bipolar chain:

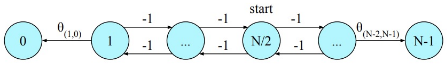


Figure 1: Graph of "Bipolar Chain" example

2) Parallel chains, constructed as a a set of linked lists with a common node at one end. All nodes have 0's except the the end nodes. Those have arbitrary unique rewards.

Here is an image from the paper showing a set of parallel chains:

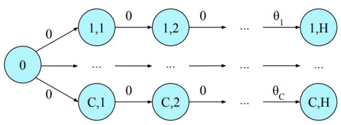


Figure 2: Graph of "Parallel Chains" example

3) Maximum reward path. This is constructed as a graph with nodes connected with probability p and rewards that are normally distributed.

[1] Department of Computer Science Stanford University, Stanford, California. Correspondence to: Ed Fancher <efancher@stanford.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

2. Approach

Although this is a multiple agent problem, the Seed Sampling paper treated actions as occurring within discrete time steps, so it wasn't necessary to use a distributed (including multi-threaded) approach. So, I decided to take advantage of the fairly nice POMDP.jl framework in Julia.

Steps:

- Solve the true MDP using some method. I've been using Q-learning with e-greedy so far.

- Use the POMDP.jl methods to run a simulation on the problem, providing a function policy that matches the exploration method. For the chain, this stops as soon as it hits a goal state. For the other two, likely a complete exploration will be necessary.

  Each successive time step will see a new agent started, so if there are k agents, there would be one agent started at each time step t_1, t_2, ..., t_k.

- Repeat for multiple runs, stepping through different #'s of agents. Calculate the average regret.

  Currently I have all three exploration methods working for the chain MDP.

Next steps:

- Do a full run for bipolar chain with 100 states and a horizon of 150, per the Seed Sampling paper.

  I expect to start this right after assignment 3 is due.

- Repeat for parallel chains

  This should be a relatively straight forward update from the bipolar chain.

  I expect to start this within a day of the assignment 3 deadline.

- Repeat for Maximum reward path (likely the most difficult)

  This one will likely be more complicated since there are a much larger possible number of actions. I expect to start this by March 6th

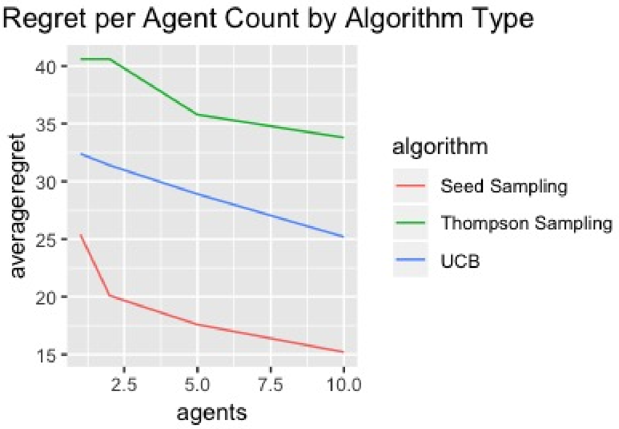
- Various items to handle reporting and graphing (probably in R). Hopefully a few days before the 20th. :)

- Final report.

3. Initial Results

I have done runs with a short 20 node bipolar chain for all three exploration types for 1, 2, 5, and 10 agents.

algorithm	agents	average regret
Seed Sampling	1	25.4
Seed Sampling	2	20.1
Seed Sampling	5	17.6
Seed Sampling	10	15.2
Thompson Sampling	1	40.6
Thompson Sampling	2	40.6
Thompson Sampling	5	35.78
Thompson Sampling	10	33.8
UCB	1	32.4
UCB	2	31.4
UCB	5	28.9
UCB	10	25.2



There are some small discrepancies. Thompson sampling shows a drop for 5 and 10 agents, but I think this is likely due to the small chain size, which allows an agent to accidentally hit the end goal, with some higher than expected frequency. In addition, in the paper, UCB was a little closer to the performance of Seed Sampling. This could be that the paper was a little vague about the parallel UCB algorithm used. I implemented a version of UCB1 that used Q and N tables that were common across the agents. Despite this, the improvement when using Seed Sampling is still quite remarkable, and generally consistent with the improvement shown in the paper (if not better, really.)

3.1. Citations and References

References

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21*, pp. 89–96. Curran Associates, Inc.,

2009. URL http://papers.nips.cc/paper/3401-near-optimal-regret-bounds-for-reinforcement-learning.pdf.

Dimakopoulou, M. and Roy, B. V. Coordinated exploration in concurrent reinforcement learning. *CoRR*, abs/1802.01282, 2018. URL http://arxiv.org/abs/1802.01282.

Strens, M. J. A. A bayesian framework for reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 943–950, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2. URL http://dl.acm.org.stanford.idm.oclc.org/citation.cfm?id=645529.658114.