POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE

École Polytechnique Montréal
Department of Computer Engineering and Software Engineering

INF8953DE - Reinforcement Learning

Project Proposal

Adam Prévost 1947205
Alexandre Morinvil 1897222
Maude Nguyen-The 1843896

October 27th, 2021

1. Project title and track.

Our project is based on the paper "How to Combine Tree-Search Methods in Reinforcement Learning" (https://arxiv.org/pdf/1809.01843.pdf) and we will be doing a reproducibility study. The paper is mainly theoretical in its approach and we will try to reproduce the results empirically.

2. Motivation and Problem Definition.

Lookahead policies are widely used in RL and are often implemented with specific planning methods such as Monte Carlo Tree Search. When referring to this problem as a tree search, the convergence of the procedure is not guaranteed, generally because it is non-contractive. The paper proposes an enhancement that leads to a contracting procedure. The authors introduce a new notion called *multiple-step greedy consistency* which can guarantee convergence.

3. Summary of the paper: Model/Approach details.

The base framework is the Markov Decision Process (MDP) viewed in class, using an *h*-greedy policy, which is just a normal greedy policy looking *h* steps (or actions) ahead to select the best action, which can be obtained by using a policy iteration procedure called *h*-PI.

The authors introduce a key notion: *h-greedy consistency*. Any pair (v: value function, π: policy) is *h-greedy consistent* if the policy improves, component-wise, the value of $T^{h-1}v$ where T is the optimal Bellman operator. They also prove that ensuring that the first pair (v_0, π_0) follows this property is sufficient to ensure that the pairs in the next iterations also do.

They then prove the convergence of two modified versions of the algorithm using *h-greedy consistency*: *hm*-PI & *hλ*-PI (which are based on two improved versions of *h*-PI: NC-*hm*-PI & NC-*hλ*-PI). These algorithms allow to find a h-greedy policy without having to do an exact value estimation, which was needed to prove the convergence of *h*-PI.

The "Experiments" section studies and compares NC-*hm*-PI and *hm*-PI. In particular, the number of calls to the simulator is plotted and used to quantify the running time before convergence of the algorithm is attained. The environment is a simple NxN gridworld environnement of uniformly distributed rewards, with & without noise.

4. List Research/Analysis Questions that you will pursue.

Reproducibility of experiments:

- Can we reproduce the empirical results of NC-*hm*-PI & *hm*-PI demonstrated in their experiments, using the same hyperparameters?
  - Regions of the hyperparameters space having corresponding convergence time.
  - Having the same "total queries" curves for each tested hyperparameter.

    - Obtain comparable performances for several hyper-parameters (measured through the error with the value function)

Additional:

- Do the same results hold true for NC-$h\lambda$-PI & $h\lambda$-PI? (compare results obtained with the same experiments for $h\lambda$-PI)
- Which algorithm converges faster?
- Do empirical results validate the expected convergence rate of γ^h *hm*-PI & $h\lambda$-PI? (Theorem 4)
- Are there special cases in regards to the previous questions?

5. Experiment setup, Dataset/Environment details.

Our environnements will be identical to the paper. We will make a simple NxN gridworld with uniformly distributed random rewards with (±0.3) & without (±0.1) noise, except one random state which has a reward of 1. Also the entries of the initial value function are drawn from $N(0,1)$ and there are no terminal states. We will use emdp, as used in assignment 1 to simulate the gridworld. We will monitor each call to the environment and use it to evaluate the algorithms based on section 4's questions. We will use plain policy iteration to get the optimal policy and compare our results.

6. Plan for contributions by each team member.

Planned Tasks:

- Set up the environment: **Adam**
- Implement classic value/policy iteration: **Maude**
- Implement h-PI: **Adam**
- Implement hm-PI and $h\lambda$-PI: **Maude**
- Implement NC-hm-PI and NC-$h\lambda$-PI: **Alexandre**
- Perform experiments: **Alexandre**
    - Convergence time heatmap
    - Total queries curves
    - Distance from optimum heatmap
- Redact final report: **Everyone**
- Prepare oral presentation: **Everyone**

Reference

Y. Efroni, G. Dalal, B. Scherrer, en S. Mannor, "How to Combine Tree-Search Methods in Reinforcement Learning", *CoRR*, vol abs/1809.01843, 2018.