Project Proposal for DS8010
Empirical Analysis of Built-In POMDP.jl Algorithms: A Comparative Study

Alice Yang (501149113)

Summary of the Proposal

Partially Observable Markov Decision Processes (POMDPs) is a mathematical framework for modelling an agent decision process under uncertainty where the underlying states are unobserved. Several algorithms have been developed for solving POMDPs, each with its own strengths and weaknesses. In this study, we aim to perform an empirical analysis of multiple algorithms using the package POMDPs.jl [3] and examine the performance of each algorithm.

Background

POMDPs is a popular tool for modelling decision-making problems in various fields such as aircraft [4], finance [1], and healthcare. However, solving POMDPs problems can be computationally challenging due to the number of state and action spaces involved in the problem set. POMDPs.jl is written in Julia programming language [2] and it provides functionality for defining models, simulating problems, and solving them with built-in algorithms.

Goal and Objectives

The objectives of this study are to implement several algorithms using the POMDPs.jl package, compare the performance of these algorithms on several POMDPs problems, and generate a performance table comparing the algorithms on different metrics, such as computation time and collected rewards. The results will be analyzed to identify each algorithm's strengths and weaknesses and determine which algorithm performs best on which type of problem. We also expect to identify areas for future research and improvement.

Methods

To achieve our objectives, we will first implement the selected algorithms. Some of the most popular solvers in POMDPs.jl uses algorithms including Incremental Pruning (IP), QMDP, Successive Approximations of the Reachable Space under Optimal Policies(SARSOP), Partially Observable Monte Carlo Planning (POMCP), POMCP with observation widening (POMCPOW), Fast Informed Bound (FIB), and Point-Based Value Iteration (PBVI).

We will then evaluate the performance of each algorithm on several models with varying levels of complexity, for instance, tiger problem, crying baby, paint problem, query problem, mini hallway problem, rock problem, simple grid world problem, and maze problem. The performance of each algorithm will be evaluated based on several metrics such as computation time and collected rewards. We will also compare the performance of each algorithm on each model.

Timeline

- Week 1-2 (March 13-26):
  - Review relevant literature on POMDPs and their algorithms
  - Prepare the POMDPs problems (crying baby, paint problem, query problem)for evaluation and evaluate the performance of each algorithm on the selected POMDPs problems
- Week 3-4 (March 27-April 9):
  - Review relevant literature on POMDPs and their algorithms, and complete intermediate report (introduction, literature review, methods section) which is **due April 10**
  - Prepare the POMDPs problems (simple grid world problem/ maze problem) and evaluate the performance of each algorithm on the selected POMDPs problems
- Week 5-6 (April 10-23):
  - Prepare the remaining POMDPs problems (simple grid world problem/ maze problem) and evaluate the performance of each algorithm on the selected POMDPs problems
  - Draft the research report (experimental setup, results, conclusions)
- Week 7 (April 24-26):
  - Review and finalize the research report, submit the research report by the **due date of April 26**

References

[1] Xi-Ren Cao De-Xin Wang. Event-Based Optimization for POMDPs and Its Application in Portfolio Management. *ScienceDirect*, 44:3228–3233, January 2011.

[2] Viral B. Shah Jeff Bezanson, Stefan Karpinski and Alan Edelman. Julia: A Fast Dynamic Language for Technical Computing. April 2012.

[3] Edward Balaban Tim A. Wheeler Jayesh K. Gupta Maxim Egorov, Zachary N. Sunberg and Mykel J. Kochenderfer. POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty. *Journal of Machine Learning Research*, 18, April 2017.

[4] Travis B. Wolf and Mykel J. Kochenderfer. Aircraft Collision Avoidance Using Monte Carlo Real-Time Belief Space Search. *Jornal of Intelligent Robotic Systems*, January 2011.