COMP8620: MC-AIXI-CTW
Group 3

Jarryd Martin, John Aslanides, Yadunandan Sannappa,
Nrupendra Rao, Cheng Yu, Ryk Budzynski

October 2015

We outline an implementation of Veness et al.'s Monte Carlo AIXI approximation[?], and report our simulation results on a number of toy domains.

1 Introduction

Recall that the AIXI agent is defined by its actions, which for each cycle k are given by

$$a_k^{\text{AIXI}} = \arg\max_{a_k} \sum_{o_k r_k} \cdots \max_{a_m} \sum_{o_m r_m} [r_k + \cdots + r_m]\, \xi\left(o_1 r_1 \ldots o_m r_m | a_1 \ldots a_m\right),$$

where the o_n and r_n are the observation and reward provided by the environment at cycle n, and ξ is a Bayesian mixture model for the environment.

Following Veness et al., we approximate a_k^{AIXI} using Monte Carlo tree search (upper confidence bound) to approximate the expectimax, we compute a mixture over variable-order Markov models using the context-tree weighting algorithm. This approximation is denoted MC-AIXI-CTW.

We present a lightweight C++ implementation of MC-AIXI-CTW, along with implementations of a number of simple games: Pacman, Tic-Tac-Toe, Biased Rock-Paper-Scissor, Extended-Tiger, and Cheesemaze.

1.1 Files

The report archive should contain the following:

```
MC-AIXI-CTW-Grp3.zip
    \report
        report.pdf // this report
        report.tex
        cheesemaze_01.png // results plots
        extended_tiger_01.png
        biased_rock_paper_scissor_01.png
        tic_tac_toe_01.png
        pacman_01.png
    \src
        main.hpp
        main.cpp
        environment.hpp
        environment.cpp
        agent.hpp
        agent.cpp
        search.hpp
        search.cpp
```

```
predict.hpp
predict.cpp
util.hpp
util.cpp
README.md
cheesemaze.conf // environment configuration files
rockpaper.conf
tictactoe.conf
coinflip.conf
tiger.conf
```

1.2 User Manual

2 MC-AIXI-CTW Implementation

2.1 Monte Carlo Tree Search (MCTS) Algorithm

1. Source code files
2. Class structure (SearchNode, DecisionNode, ChanceNode, ...)
3. Description of the algorithm (Veness...)

2.2 Context Tree Weighting (CTW)

2.3 Environments

2.3.1 Cheesemaze

2.3.2 Extended Tiger

2.3.3 Biased Rock-Paper-Scissor

2.3.4 Tic-Tac-Toe

2.3.5 Pacman

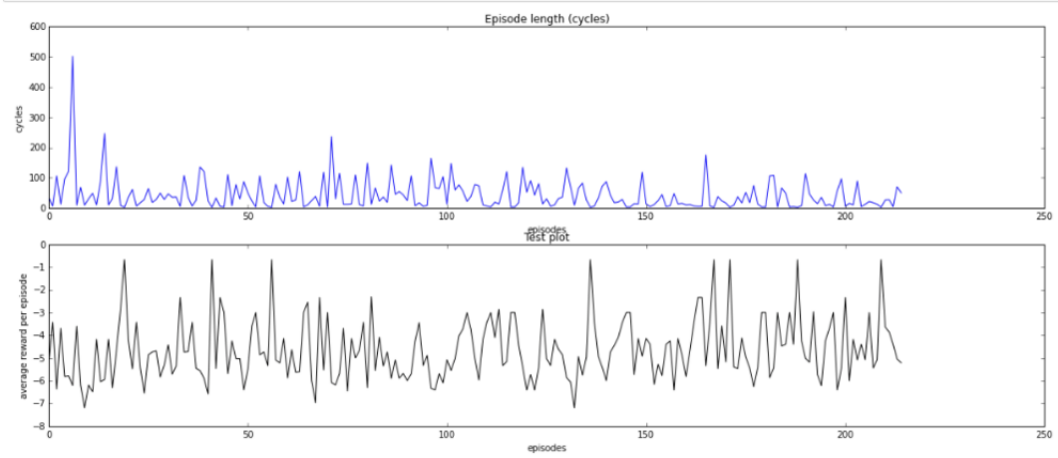
3 Simulation Results

3.1 Cheesemaze

- Experimental setup ...
  Any simulation provided should include detailed description of experimental setup; selected parameters of algorithms and examples; and concise interpretations of obtained simulation results.

Environment	MCTS	CTW	
1	$m = 100$	ct-depth $= 96$	
	$C = \sqrt{2}$		

- Plots ...



◇ Interpretation of Results

3.2 Extended Tiger

Experimental setup ...
Plots ...

3.3 Biased Rock-Paper-Scissor

Experimental setup ...
Plots ...

3.4 Tic-Tac-Toe

Experimental setup ...
Plots ...

3.5 Pacman

Experimental setup ...
Plots ...

4 Cross Domain Simulation Results

Cheesemaze and Extended Tiger

5 Possible Other things

Cross domain simulation on more difficult environments...
Separate CTW for Obs and Rews...