

3rd Year Project Report

Catherine Vlasov

May 2018



Abstract

This is my abstract

Contents

1	Introduction	2
2	Background	3
2.1	Reinforcement Learning	3
2.2	Learning Algorithms	3
2.2.1	Monte Carlo Learning	3
3	Design & Implementation	4
3.1	Overview	4
3.2	Agents	4
3.3	Tic-Tac-Toe	4
3.3.1	Rules	4
3.3.2	Implementation Details	4
3.3.3	Breaking Symmetries	4
3.4	Chung Toi	4
3.4.1	Rules	4
3.4.2	Implementation Details	4
3.4.3	Breaking Symmetries	4
4	Results	5
4.1	Overview	5
4.2	Monte Carlo Learning Parameter	5
4.3	Policies	5
4.4	Time	5
4.5	Win/Learning Rate	5
5	Analysis	6
6	Conclusion	7
6.1	Future Work	7
6.2	Personal Reflections	7
7	Acknowledgements	9

Chapter 1

Introduction

Chapter 2

Background

2.1 Reinforcement Learning

2.2 Learning Algorithms

2.2.1 Monte Carlo Learning

Chapter 3

Design & Implementation

3.1 Overview

3.2 Agents

3.3 Tic-Tac-Toe

3.3.1 Rules

3.3.2 Implementation Details

3.3.3 Breaking Symmetries

3.4 Chung Toi

3.4.1 Rules

3.4.2 Implementation Details

3.4.3 Breaking Symmetries

Chapter 4

Results

4.1 Overview

4.2 Monte Carlo Learning Parameter

4.3 Policies

4.4 Time

4.5 Win/Learning Rate

Chapter 5

Analysis

Chapter 6

Conclusion

6.1 Future Work

6.2 Personal Reflections

This has been a really great learning experience, not only in terms of discovering the field of reinforcement learning and its applications, but also in terms of gaining experience with building up a large, complex programming project from scratch using a variety of technical tools.

When I was in high school, I designed and implemented the classic board game Nine Mens Morris where a user plays against an algorithm I wrote. My plan for the project was to ask my friends and family to play my game and to store the moves made in each game so that I could build up a database of sequences of moves and outcomes of the game. I hoped to somehow compute which moves were the best and get my algorithm to learn which moves made winning the most likely. Little did I know, I wanted to invent RL from scratch unaware of its existence or of the amount of research in the field. Several years later, I am delighted to have had the chance to work on a similar project, but this time on a deeper level using well-known computational algorithms.

When I started working on this project, I decided to treat it as an opportunity to develop many skills that are critical for a career in software engineering:

- Writing clean, maintainable, well-documented code
- Designing and implementing tests for my code
- Using a build tool to manage dependencies between packages in my project as well as with external libraries
- Using version control effectively

I tested the methods in my API using a unit-testing framework for Java called JUnit. I also used a testing framework for Java called Mockito this allowed me to verify the behaviour of objects with external dependencies by creating mock objects for these dependencies, which mimic real objects but do so in a particular way that I can specify.

In order to save my experiment results in a format that would facilitate the creation of graphs, I used opencsv, a CSV parser library for Java.

To manage my projects dependencies, I used a build tool developed by Google called Bazel and I used Git for version control.

Overall, I really enjoyed learning about RL algorithms and exploring their applications and success rates, all while developing strong programming skills that will help me throughout my career.

Chapter 7

Acknowledgements

Thanks for reading.