

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

---

# Symbolic Reinforcement Learning using Inductive Logic Programming

---

*Author:*  
Kiyohito Kunii

*Supervisor:*  
Professor Alessandra Russo

Submitted in partial fulfillment of the requirements for the MSc degree in MSc in  
Computing Science of Imperial College London

June 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Inductive Logic Programming . . . . .	4
2.1.1	Stable Model Semantics . . . . .	5
2.1.2	Answer Set Programming (ASP) . . . . .	6
2.1.3	Learning from Answer Sets (LAS) . . . . .	7
2.1.4	Inductive Learning of Answer Set Programs (ILASP) . . . . .	7
2.2	Reinforcement Learning . . . . .	8
2.2.1	Markov Decision Process(MDP) . . . . .	8
2.2.2	Temporal-Difference (TD) Learning . . . . .	9
2.2.3	Q-Learning . . . . .	9
2.3	GVGAI Framework . . . . .	10
<b>3</b>	<b>Related Work</b>	<b>11</b>
3.1	Symbolic Reinforcement Learning . . . . .	11
<b>4</b>	<b>Project Overview</b>	<b>13</b>
4.1	Proposed Architecture . . . . .	13
4.2	Project outline . . . . .	13
<b>5</b>	<b>Ethics Checklist</b>	<b>14</b>

# Chapter 1

## Introduction

There has been successful applications of deep reinforcement learning (DRL) a number of domains, such as games and robotics (TODO INSERT REFERENCES). DRL is considered to be a step towards artificial general intelligence. As pointed by XXX, however, there are still a number of issues to overcome in this method. First, it requires a large amount of data for training the model, which requires a long time of learning process. Second, it is considered to be a black-box, meaning the decision making process is unknown to human user and therefore lacks explanation ability. Third there is no thoughts process to the decision making, which, as XX points out, is a fundamental to the artificial general intelligence. To tackle these problems, there are 3 main streams of research on this field. First main research is focused on applying Bayesian statistics XXX, the second main research is XXX. Recently, the XXX attempted to incorporate symbolic representations into the system to achieve more data-efficient learning, which shows a promising results of this approach.

### MOTIVATION

Why symbolic reinforcement learning is good attempt

Reason 1: Comprehensive by humans → Explainable rather than black-box

Reason 2: Similar to the human, it uses reasoning

Use of previous experience (background knowledge) As discussed in XX, there are many research that attempted to incorporate symbolic reasoning into DRL, but there is not much research on incorporating symbolic learning with reinforcement learning. However there is a room for exploration on this field.

Reason 3: Recent advance of ILASP is promising

Because of the recent advancement of logic-based learning and deep reinforcement learning, combination of both approach would be a next exploration toward artificial general intelligence.

**OBJECTIVES** combining the two novel approaches to overcome the problems of the QND

In this paper, I further explore incorporation of symbolic machine learning into reinforcement learning to achieve data-efficient learning using Inductive Learning of Answer Set Programs (ILASP), which is the state-of-art symbolic learning method that can be applied to incomplete and more complex environment. This research is inspired by [1], but in this paper I explore symbolic representations process and include more learning aspect.

This background report is organised as follows: Chapter 2 introduces

# Chapter 2

## Background

This section introduces introduces Inductive Logic Programming (Section 2.1), Reinforcement Learning (Section 2.2) and GVGAI Framework, an open-source game platform for AI methods (Section 2.3), which provide the foundations of this work.

### 2.1 Inductive Logic Programming

Inductive Logic Programming (ILP) is a subfield of research area aimed at the intersection between machine learning and logic programming [3]. The purpose of ILP is to inductively derive a hypothesis  $H$  that is a solution of a learning task, which covers all positive examples and any of negative examples.

ENTAILMENT

Satisfiable

$$B \cap H \models E \quad (2.1)$$

where  $E$  consists of positive examples ( $E^+$ ) and negative examples ( $E^-$ )

To compute this inference task, the syntax of the predicate logic program needs to be converted into Conjunctive Normal Form (CNF), or clausal theory, which consists of a conjunction of clauses, where a clause is disjunction of literals, and a literal can include positive literals and negative literals.

A literal is either an atom  $p$  or negation by failure, which is used to derive not  $p$ .

**Example 2.1.1** (*Conjunctive Normal Form*)

The *Herbrand domain* (a.k.a *Herbrand universe*) of clause sets  $Th$  is the set of all ground terms that are constants and function symbols appeared in  $Th$ .

**Example 2.1.2** (*Herbrand Domain*)

The *Herbrand base* of  $Th$  is the set of all ground predicates that are formed by predicate symbols in  $Th$  and terms in the *Herbrand domain*.

**Example 2.1.3** (*Herbrand Base*)

The *grounding* of  $Th$  is the set of all clauses that are  $c \in Th$  and variables are replaced by terms in the *Herbrand Domain*.

#### Example 2.1.4 (Grounding)

TODO Explain grounding in ASP context.

A *Horn clause* is a subset of CNF, which is a clause with at most one positive literal, where a *definite clause* (also called a *rule* or a *fact*) contains exactly one positive literal, and a *denial* (or a *constraint*) is a clause with no positive literals.

#### Example 2.1.5 (Horn clause)

Limitations of Inductive Logic Programming:

### 2.1.1 Stable Model Semantics

TODO Relationship from the previous Section

Definite Logic Program is a set of definite rule where:

a definite rule is of the form  $h \leftarrow a_1, \dots, a_n$ , where  $h, a_1, \dots, a_n$  are all atoms.

whereas

Normal Logic Program is a set of normal rule where

a normal rule is of the form  $h \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_n$  where  $h$  is the head of the rule, and  $a_1, \dots, a_n, b_1, \dots, b_n$  are the body of the rule (both the head and body are all atoms).

The grounding of a normal logic program  $P$  can be obtained by replacing each rule in  $P$  with a ground instance of the rule, such that for each atom  $A$  in  $\text{body}^+(R)$  (TODO EXPLAIN WHAT THIS IS), already occurs in the head of another ground rule.

#### RELATIONSHIP WITH PROLOG

The entire program needs to be grounded in order for ASP solver to work, and, unlike Prolog, each rule must be “safe”. A rule  $R$  is safe if every variable that occurs in the head of the rule occurs at least once in  $\text{body}^+(R)$  (TODO check if this is correct).

An *Herbrand interpretation* of a set of definite clauses  $Th$  is a subset of the Herbrand base of  $Th$ , which is a set of ground atoms that are true in terms of interpretation.

Interpretation evaluate it to true Interpretation evaluate it to false

A *Herbrand Model* is a Herbrand interpretation if and only if a set  $Th$  of clauses is satisfiable.

Algorithm for computing Herbrand model is shown in

XXXX

A set of clauses  $Th$  is unsatisfiable if the algorithm terminates and does not return Herbrand model.

The *Least Herbrand Model* is an unique minimum Herbrand model if and only if none of its subsets is an Herbrand model (TODO MATHEMATICALLY SAY THIS)

For a normal program, there is usually no least Herbrand model. TODO Explain MORE

Stable Model of a normal logic program is XX [2]

Any stable model is a minimal Herbrand model, and stable sets is stable models. The stable models can be found by constructing the result of the program with respect to sets of atoms  $X$  ( $P^x$  in the following 2 steps  
does not contain negation and a unique minimal Herbrand model.

**Example 2.1.6** (*Horn clause*)

### 2.1.2 Answer Set Programming (ASP)

Answer set of normal logic program  $P$  is a stable model, and Answer Set Programming (ASP) is a normal logic with extensions: constrains, choice rules and aggregates. ASP program consists of a set of rules, where each rule consists of an atom and literals. A literal is either an atom  $p$  or its *default negation*  $\text{not } p$  (Negation as a failure).

Head and body

A Constraint is of the form

**Theorem 1**  $\leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_n$

which is to filtering any irrelevant answer sets. There are two types of constraints: soft and hard constraints. Soft constraint is XXX

Hard constraint is XXX

Choice rule can express possible outcomes given an action choice, which is of the form

**Theorem 2**  $lh_1, \dots, h_mu \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_n$

An answer set of ASP program is interpretations that make all the rules true.

Non-monotonicity.

Clingo

ASP has true, false and unknown

for non-deterministic to describe transition possibilities.

Syntax Examples

optimisation statement is of the form.

Which is useful to order the answer sets in terms of preference.

Syntax examples.

Aggregate

choice rules

An ASP program  $P$  is normal logic program with addition of choice rules, constraints and optimisation statement.

Answer set of  $P$  is

**Cautious Induction**

Sakama 2008

no concept of negative examples in this paper.

Cautious Induction task is of the form  $\langle B, E^+, E^- \rangle$  where:

$B$  is the background knowledge

$E^+$  is a set of positive examples

$E^-$  is a set of negative examples

$H \in \text{ILP}_{\text{cautious}} \langle B, E^+, E^- \rangle$  if and only if

there is at least one answer set  $A$  of  $B \cup H$  such that:

for every answer set  $A$  of  $B \cup H$ :  $\forall e \in E^+ : e \in A$

$\forall e \in E^- : e \notin A$

One of the limitation of cautious induction is that XX

Sakama 2009

**Brave Induction**

Similarly, Brave Induction task is of the form  $\langle B, E^+, E^- \rangle$  where:

$B$  is the background knowledge

$E^+$  is a set of positive examples

$E^-$  is a set of negative examples

$H \in \text{ILP}_{\text{brave}} \langle B, E^+, E^- \rangle$  if and only if there is at least one answer set  $A$  of  $B \cup H$  such that:

$\forall e \in E^+ : e \in A$

$\forall e \in E^- : e \notin A$

One of the limitation of brave induction is that it cannot learn constraints as shown in the example

Sakama 2009

**2.1.3 Learning from Answer Sets (LAS)**

Learning from Answer Sets was developed in [Law et al 2014] to overcome limitations of cautious induction and brave induction.

Examples used in LAS is converted from  $\langle E^+, E^- \rangle$  into

Partial Interpretations are of the form  $\langle E^{\text{inc}}, E^{\text{exc}} \rangle$

A Herbrand Interpretations extends a partial interpretation if it include all of the inclusions and none of the exclusions.

**2.1.4 Inductive Learning of Answer Set Programs (ILASP)**

ILASP is an algorithm that is capable of solving LAS tasks



It is based on two fundamental concepts: positive solutions and violating solutions.

A hypothesis  $H$  is a positive solution if and only if 1.  $H \subseteq S_M$

2.  $\forall e^+ \in E^+ \exists A \in AS(B \cup H)$  subject to  $A$  extends  $e^+$

A hypothesis  $H$  is a violating solution if and only if 1.  $H \subseteq S_M$

2.  $\forall e^+ \in E^+ \exists A \in AS(B \cup H)$  subject to  $A$  extends  $e^+$

3.  $\exists e^- \in E^- \exists A \in AS(B \cup H)$  subject to  $A$  extends  $e^-$

$ILP_{LAS}$  is positive solutions that are not violating solutions.

ILASP task containing a context-dependent example

TODO Explain how symbolic learning works

TODO What would you learn in my context? Relationship of the objects? Objects, types, locations and interactions.

## 2.2 Reinforcement Learning

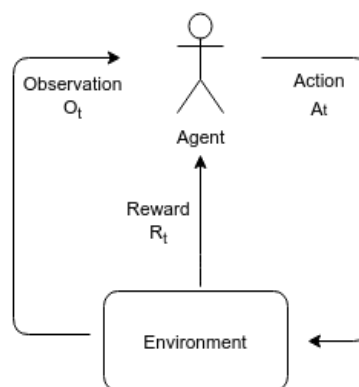


Figure 2.1: Agent and Environment

On-Policy learn policy  $\pi$  from its past experience Off-policy

Model-based vs model-free learning

### 2.2.1 Markov Decision Process(MDP)

(Puterman, 1994) There is an agent who interacts with an environment (Figure 2.1). At each time step, an action taken by the agent affect the environment state and the reward (or penalty) it receives from the action outcome. (TODO Observations) When an agent must make a sequence of decision, the sequential decision problem can be formalised using Markov decision process (MDP). MDPs formally represent a fully observable environment of an agent for reinforcement learning.

A MDP is of the form  $\langle S, A, T_a, R_a, \gamma \rangle$  where:

- $S$  is the set of finite states that is observable in the environment

- $A$  is the set of finite actions taken by the agent
- $T_a(s, s')$  is a state transition in the form of probability matrix  $\Pr(S_{t+1} = s' \mid s_t = s, a_t = a)$ , which is the probability that action  $a$  in state  $s$  at time  $t$  will result in state  $s'$  at time  $t+1$ .
- $R$  is a reward function  $R_a(s, s') = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$ , the expected immediate reward that action  $a$  in state  $s$  at time  $t$  will return
- $\gamma$  is a discount factor  $\gamma \in [0,1]$ , which represents the preference of the agent for present rewards over future rewards

In MDPs, there is an element of delayed reward tradeoff between immediate rewards and its delayed reward

A state  $S_t$  is Markov if and only if  $P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \dots, S_t]$  therefore the probability of reaching  $S_{t+1}$  depends solely on  $S_t$ , which captures all the relevant information from earlier history.

A solution to the sequential decision problem is called a policy  $\pi$ , a sequence of actions that leads to a solution

The transition and reward functions are not necessary to be known to compute  $\pi$ .

An optimal policy  $\pi^*$  is the one that maximise the total rewards in the environment.

The total reward with a discount factor is

The existence of the discount factor can be justified as follows:

TODO  $\mathbb{E}[R_{t+1} \mid S_t = s]$

Reinforcement learning is a method to get approximated optimal solution.

### 2.2.2 Temporal-Difference (TD) Learning

To solve MDP, one of the approaches is called Temporal-Difference (TD) Learning.

TD learns directly from episodes of experiences, which can be incomplete. TD does not require knowledge of MDP transitions and rewards (model-free) Sutton 1988

Update value

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) \quad (2.2)$$

where  $R_{t+1} + \gamma V(S_{t+1})$  is the estimated return (a.k.a TD target)

$R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$  is TD error.

TD updates the estimate by using the estimates of XXX (bootstrap).

The advantages of TD methods - does not require any model of an environment. - online learning

### 2.2.3 Q-Learning

model-free learning, where the agent will not rely on the models of the environment.

Q-learning approximate the  $Q(a,s)$  from the samples of

the agent only knows about the possible states and actions but the transition state and reward probability functions are unknown.

Q function is the state-action pair

the optimal Q-function  $Q^*(s,a)$  represents XXX for the agent to selection action  $a$  given that it is in state  $s$ .

Q-learning is off-policy TD learning defined in [Watkin 1989], which is of the form: where  $\alpha$  is the learning rate,  $\gamma$  is a discount rate between 0 and 1.

this equation is used to upate action-value function

Model free learning: derectly derive an optimal policy by interacting with the envi-  
ronment without the model

Model free can be done using Monte Carlo Policy evaluation

One way to solve the Bellman Optimality equation is Q-leraning

$U(s) = \max_a Q(s,a)$

a function  $Q(S,A)$  which predicts the best action  $A$  in state  $S$  to maximise the total cumulative rewards.

The function is estimated by Q-learning, which repeately updates  $Q(s,a)$  using the Bellman Equation.

TODO INSERT Q-learning ALGORITHM HERE

Epsilon greedy

## 2.3 GVGAI Framework

The Video Game Definition Language (VGDL)

SpriteSet defines all the sprites available in the game. LevelMapping defines relationships among characters, and the sprites available. InteractionSet defines what events occur when two sprites collide TerminationSet defines the end conditions of the game, and decides whether the player wins or not.

# Chapter 3

## Related Work

Transparency and interpretable capability of the model is another important aspect for machine learning applications.

XXX[Programmati...] developed a programmatically interpretable reinforcement learning which finds a policy that can be represented in a human-readable programming language.

Two most studied approach for using previous learning exprience is meta-learning and transfer learning

Artificial General Intelligence

Definision of AGI

The history of data-efficient learning What other people have done in this.

The advance of statistical machine learning methods, especially deeep reinforcement learning

AlphaGo, and AlphaGo Zero

Study of symbolic machine learning roots from

Baysian Optimisation

RNN approach

Symbolic Deep reinforcement learning

Some implementation: German paper

The paper was the application of symbolic representatiojns into a very simple game to demonstrate this proof of concept would actually work. By contract, there is active reserach in symbolic machine learning, which focuses on logic=based learning rather than statistical machine learning. For example, XX shows the agent can learn XXX from a noisy examples, with only a very few training examples.

### 3.1 Symbolic Reinforcement Learning

Incorporation of logic into reinforcement learning dates back to the study of relational reinforcement learning,

More recently there has been a number of attempts to incorporate ASP into reinforcement learning.

There are a number of reseached conducted in applying DNN to symbolic reasoning. For example,

[From GamePlay to Symbolic Reasoning]

However, as XX points out, 1 pixel of the game could influence the decision making of the agent.

# Chapter 4

## Project Overview

### 4.1 Proposed Architecture

### 4.2 Project outline

The project outline

Implement baseline performance (DQN, Q-learning, ASPRL)

Due to the complex environment of the chosen game, I will not implement the original DSRL, since the feature extractions from the pixel would only work in a very simple pixel environment,

Pipeline

The basic architecture follows the similar method in XXX, but I also add symbolic learning to these extracted symbolic features using ILASP.

Apply ILASP to the ASP, which involves development of the pipeline of ILASP in Python

Finally use Q-learning that allow

which measurement would you use? (grid world, something else? GVGAL games)

Summarise different types of knowledge representations (Objects ?? relationship?)

Common sense

Lastly, I will also test the capability of transfer learning for this new method.

# Chapter 5

## Ethics Checklist

	Yes	No
<b>Section 1: HUMAN EMBRYOS/FOETUSES</b>		
Does your project involve Human Embryonic Stem Cells?		✓
Does your project involve the use of human embryos?		✓
Does your project involve the use of human foetal tissues / cells?		✓
<b>Section 2: HUMANS</b>		
Does your project involve human participants?		✓
<b>Section 3: HUMAN CELLS / TISSUES</b>		
Does your project involve human cells or tissues? (Other than from Human Embryos/Foetuses i.e. Section 1)?		✓
<b>Section 4: PROTECTION OF PERSONAL DATA</b>		
Does your project involve personal data collection and/or processing?		✓
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?		✓
Does it involve processing of genetic information?		✓
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.		✓
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?		✓
<b>Section 5: ANIMALS</b>		
Does your project involve animals?		✓
<b>Section 6: DEVELOPING COUNTRIES</b>		
Does your project involve developing countries?		✓

If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?		✓
Could the situation in the country put the individuals taking part in the project at risk?		✓
<b>Section 7: ENVIRONMENTAL PROTECTION AND SAFETY</b>		
Does your project involve the use of elements that may cause harm to the environment, animals or plants?		✓
Does your project deal with endangered fauna and/or flora /protected areas?		✓
Does your project involve the use of elements that may cause harm to humans, including project staff?		✓
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?		✓
<b>Section 8: DUAL USE</b>		
Does your project have the potential for military applications?		✓
Does your project have an exclusive civilian application focus?		✓
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?		✓
Does your project affect current standards in military ethics e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?		✓
<b>Section 9: MISUSE</b>		
Does your project have the potential for malevolent/criminal/terrorist abuse?		✓
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?		✓
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?	✓	
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?		✓
<b>Section 10: LEGAL ISSUES</b>		
Will your project use or produce software for which there are copyright licensing implications?	✓	



Will your project use or produce goods or information for which there are data protection, or other legal implications?		✓
<b>Section 11: OTHER ETHICS ISSUES</b>		
Are there any other ethics issues that should be taken into consideration?	✓	

# Bibliography

- [1] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards Deep Symbolic Reinforcement Learning. sep 2016. pages
- [2] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. *5th International Conf. of Symp. on Logic Programming*, (December 2014):1070–1080, 1988. pages
- [3] S Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991. pages