

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Data Efficient Deep 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

1	Introduction	1
2	Background	2
2.1	Data Efficient Learning	2
2.2	ASP in Reinforcement Learning	2
3	Background	3
3.1	Inductive Logic Programming	3
3.1.1	Stable Model Semantics	3
3.1.2	Answer Set Programming (ASP)	3
3.1.3	Learning from Answer Sets (LAS)	5
3.1.4	Inductive Learning of Answer Set Programs (ILASP)	5
3.2	Reinforcement Learning	5
3.2.1	Markov Decision Process(MDP)	5
3.2.2	Temporal-Difference (TD) Learning	6
3.2.3	Q-Learning	6
3.3	Deep Reinforcement Learning	7
3.4	Symbolic Reinforcement Learning	7
3.5	GVGAI Framework	7
4	Project Overview	8
4.1	Motivation	8
4.2	Objectives	8
4.3	Project outline	8
4.4	Contribution	9
4.5	Methods	9
4.6	Legal and Ethical Issues	9
4.7	References	9

Chapter 1

Introduction

There has been successful applications of deep reinforcement learning (DRL) a number of domains, such as games and robotics. DRL is considered to be a step towards artificial general intelligence. However there are still a number of issues to overcome in this method. As pointed by XXX, there are 3 major problems. 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 and therefore not explainable. Third there is no thoughts process to the decision making, which, as XX points out, is a fundamental to the artificial general intelligence. To overcome 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. The paper was the application of symbolic representations into a very simple game to demonstrate this proof of concept would actually work.

By contrast, there is active research 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.

In this paper, I explore incorporation of symbolic machine learning into XXX 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 based on XX, but in this paper I explore symbolic representations process and include more learning aspect.

Problem setting Garnelo et al. (2016)

Chapter 2

Background

2.1 Data Efficient Learning

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

Also business success.

Study of symbolic machine learning roots from

Baysian Optimisation

RNN approach

Symbolic Deep reinforcement learning

Some implementation: German paper

2.2 ASP in Reinforcement Learning

Chapter 3

Background

3.1 Inductive Logic Programming

Inductive Logic Programming (ILP) is a subfield of research area aimed at the intersection of machine learning and logic programming (MUGGLETON 1991). The purpose of ILP is to derive a hypothesis H that is a solution of a learning task, which covers all positive examples and any of negative examples.

$$B \cap H \models E \quad (3.1)$$

TODO Herbrand Model

TODO Least Herbrand Model

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).

3.1.1 Stable Model Semantics

Stable Model of a normal logic program

3.1.2 Answer Set Programming (ASP)

Answer Set Programming (ASP) is a declarative programming

Literals

Negation as a failure

constraints is of the form $\leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_n$

Constraints are to filtering any irrelevant answer sets.

Syntax Examples

There are two types of constraints: soft and hard constraints. Soft constraint is XXX

Hard constraint is XXX

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

$H \in ILP_{\text{cautious}} \langle B, E^+, E^- \rangle$ 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

$H \in ILP_{\text{brave}} \langle B, E^+, E^- \rangle$ 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

3.1.3 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 $\langle E^{inc}, E^{exc} \rangle$

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

3.1.4 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. $\forall e^+ \in E^+ \exists A \in AS(B \cup H) \text{ subject to } A \text{ extend } e^+$

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

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

3. $\exists e^- \in E^- \exists A \in AS(B \cup H) \text{ subject to } A \text{ extend } e^-$

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

ILASP task containing a contex-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.

3.2 Reinforcement Learning

An RL agent may include either Policy, Value function or Model,

where policy

On-Policy Off-policy

Bellman Equation

3.2.1 Markov Decision Process(MDP)

MDPs formally represent a fully observable environment of an agent for reinforcement learning.

A state S_t is Markov if and only if

$P[S_{t+1} | S_t] = P[S_{t+1} | S_1, \dots, S_t]$ therefore the probability of reaching S_{t+1} depends solely on S_t , which captures all the relevant information from earlier history.

A MDP is of the form $\langle S, A, T, R, \gamma \rangle$ where :

- S is the set of finite states that is observable in the environment
- A is the set of finite actions executable by the agent

- T is a state transition in the form of probability matrix XXXXX
- R is a reward function
- γ is a discount factor $\gamma [0, 1]$

3.2.2 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

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) \quad (3.2)$$

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

3.2.3 Q-Learning

Q-learning is off-policy TD learning defined in [Watkin 1989], which is of the form:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

(3.3)

where α is the learning rate, γ is a discount rate between 0 and 1.

this equation is used to update action-value function

Model free learning: directly derive an optimal policy by interacting with the environment without the model

Model free can be done using Monte Carlo Policy evaluation

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

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

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 repeatedly updates $Q(s, a)$ using the Bellman Equation.

TODO INSERT Q-learning ALGORITHM HERE

Epsilon greedy

3.3 Deep Reinforcement Learning

3.4 Symbolic Reinforcement Learning

Explain the paper

3.5 GVGA Framework

Chapter 4

Project Overview

4.1 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)

Not much explored yet. 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.

4.2 Objectives

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

4.3 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,

would not be realistic in this case.

Pipeline Implement the CNN side that is able to extract features of the game and convert into ASP syntax.

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

Implement based on Towards Deep Symbolic Reinforcement Learning

Transfer learning

4.4 Contribution

To my knowledge, this is the first time that both symbolic learning method is incorporated into a reinforcement learning to facilitate learning process

4.5 Methods

4.6 Legal and Ethical Issues

???

4.7 References

Bibliography

Garnelo, M., Arulkumaran, K., and Shanahan, M. (2016). Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*. pages 1