Imperial College
London

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

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 Baysian statistics XXX, the second main research is XXX. Recently, the XXX attempted to incorporate symbolic representations into the system to achive more data-efficient learning, which shows a promissing results of this approach.
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 imcomplete and more complex environment. This research is inspired by [**?**], but in this paper I explore symbolic represetations 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), Deep Reinforcement Learning (Section 2.3) and GVGAI Framework, an open-source game platform for AI methods (Section 2.4), 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 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 coveres all positive examples and any of negative examples.
Limitations of Inductive Logic Programming:

$$B \cap H \models E \tag{2.1}$$

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, ..., a_n, where h , a_1, ..., 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, ..., a_n, *not* b_1, ..., *not* b_n where h is the head of the rule, and a_1, ..., a_n, b_1, ..., b_n are the body of the rule (both the head and body are all atoms).

2.1.1 Stable Model Semantics

Stable Model of a normal logic program

2.1.2 Anwer Set Programming (ASP)

Answer Set Programming (ASP) is a declarative programming

Literals
Negation as a failure
constrains is of the form
Clingo
$\leftarrow a_1, ..., a_\text{n}, notb_1, ..., notb_\text{n}$
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 anser 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 indction 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 indction is that it cannot learn constrains as shown in the example
Sakama 2009

2.1.3 Learning from Anwer 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.

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.$\forall e^{+} \in \exists A \in AS(B \cup H) subjecttoAextendse^{+}$

A hypothesis H is a violating solution if and only if 1. H $\subseteq S_{M}$
2.$\forall e^{+} \in E^{+} \exists A \in AS(B \cup H) subjecttoAextendse^{+}$
3.$\exists e^{-} \in E^{-} \exists A \in AS(B \cup H) subjecttoAextendse^{-}$

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? Relashinship of the objects? Objects, types, locations and interactions.

2.2 Reinforcement Learning

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

2.2.1 Markov Decision Process(MDP)

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

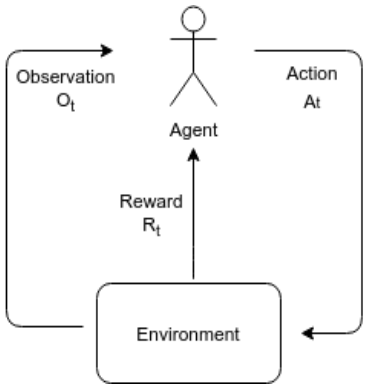

Figure 2.1: Agent and Environment

A MDP is of the form $\langle$ S, A, T_a, R_a, γ $\rangle$ 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 probablity 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
- γ 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 $P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, ..., S_t]$ therefore the probability of reaching S_{t+1} depends solely on S_t, which captures all the relevant information from earilier history.
A solution to the sequential decision problem is called a policy π, which is a distributin over actions given states.
An optimal policy π^* 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

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

Update value

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

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 leaning, 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:

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

(2.3)

where α is the learning rate, γ 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 environment 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 Deep Reinforcement Learning

2.3.1 Convolutional Neural Networks (CNN)

2.3.2 Deep Q-Networks (DQNs)

Explain the paper

2.4 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

3.1 Deep Reinforcement Learning and its issues

3.1.1 Data efficient learning

3.1.2 Transfer learning

3.1.3 Planning

3.1.4 Transparency

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.

3.2 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 Motivation

Why symbolic reinforcement learning is good attempt
Reason 1: Complehensive by humans → Explanable 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 symbolc learning with reinforcement learning However there is a room for exploration on this field.
Reason 3: Recent advance of ILASP is promissing
Because of the recent advancement of logic-based learning and deep reinforcement learning, combination of both approach would be a next explonation toward artificial general intelligence.

4.2 Objectives

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

4.3 Project outline

The probject outline
Implement baseline performance (DQN, Q-learning, ASPRL)
Due to the complex environment of the chosen game, I will not implemnet 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 word, something else? GVGAL games)
Summarise different types of knowledge representations (Objects ?? relationship?)
Common sense
Implement based on T̈owards Deep Symbolic Reinforcement Learning
Lastly, I will also test the capability of transfer learning for this new method.

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

???

Chapter 5

References

Bibliography