System design report: Multi-Agent System

James King
Supervisor: Kostas Stathis

October 2018

Abstract

Prolog services, Flask web application
https://users.ece.cmu.edu/ koopman/essays/abstract.html

I. Introduction

One of the aims of my project is to produce a multi-agent system to play games of indirect reciprocity, the model for which I have defined in my second report: "Report on indirect reciprocity, strategies for agents and the development of a concrete model to implement". For this I have designed a system that includes two main components: The environment and a web service to host agent's decision making components.
In my project I will be hosting the environment in my Flask web application, but, the aim is that the environment can be used as if it is a library as long as it is connected to an application/service that runs agent's decision making component. As such I will not be addressing how the web application hosting the environment, but the environment itself and the agents service.

II. Content and Knowledge

Talk about overall system design
Talk about choice between kostas' idea and mine.
Talk about proof of concept applications and lessons learnt from them.
Talk about learning from Annie Ogborne.
Talk about security: Prolog injection attacks, shell injection attacks, sql injection, remove library(http/http_errors.pl) in production to make it harder

i. Agents

There are many different definitions of what properties a system needs in order to be considered an agent. One widely accepted definition containing 4 properties was given by Wooldridge and Jennings [3]: autonomy, reactivity, proactivity and social ability. This 'weak' notion of agency is widely accepted, but Wooldrige and Jennings note that some argue for a stronger notion that include human-like concepts. One such notion could be goals, humans tend to work towards goals so this could fit the agents paradigm.
The goals of agents in game-theoretic models vary, two examples of goals are: maximising social welfare for all and maximising your own fitness. The first example is attempted by the strategy always cooperate as always cooperating produces the maximum social welfare according to the payoff matrix in table 1. Due to the nature of the payoff matrix and the model, the last goal is not a simple goal to reach and strategies attempt to reach this (with varying degrees of success) by encoding a theory of what is best to do when. An example of this is the image scoring discriminator strategy which encodes a theory that when an image score is greater than or equal to k (a variable set at the

initialization of the agent) it is best to cooperate, else the agent will defect. But, some agents don't even have a goal and just seek to provide a theory to act on.

Donor Action	Payoffs	
	Donor	Recipient
Cooperation	-1	2
Defection	0	0

Table 1: *The payoff used in my indirect reciprocity model*

These indirect reciprocity strategies have remarkable resemblance to how theories are encoded in deductive reasoning agents. Deductive reasoning agents use theories to encode how it is best to act under any given situation [2]. Agents who follow the image score discriminator strategy could have a theory encoded in them such as:

$$
\begin{aligned}
&interaction(me, Recipient, time) \quad \wedge \\
&image\_score(me, Recipient, Score, time) \quad \wedge \\
&Score \geq k \rightarrow do(cooperate(Recipient))
\end{aligned}
$$

$$
\begin{aligned}
&interaction(me, Recipient, time) \quad \wedge \\
&image\_score(me, Recipient, Score, time) \quad \wedge \\
&Score < k \rightarrow do(defect(Recipient))
\end{aligned}
$$

$$
\neg interaction(me, \_, time) \rightarrow do(idle)
$$

Part of the deductive reasoning agents method of implementing agents is the logical database that includes information on the current state of the world. In the example above this would possibly include logical data such as:

$$
\begin{aligned}
&image\_score(agent1, agent2, 2, 9). \\
&interaction(agent1, agent2, 6).
\end{aligned}
$$

This logical database is similar to the human-like concept of belief that is included in the language Agent0 presented by Yoav Shoham [1] as one of his two mental categories: belief and commitment. Beliefs are a fact that is thought to be true by an agent at a specific time about a specific time (an agent is constrained to not believe contradictory facts). Commitments are a commitment to act (restricted by the agent's capabilities) not a commitment to pursue a goal. Agent capabilities to Shoham are relations between the agent's mental state and their environment. An agent is only capable of committing to an action iff they believe themself to be. In my example above this is shown by the agent only being capable of defecting or cooperating if they are a donor in an interaction.

In my system I wish to incorporate the idea of beliefs, commitments, capabilities and strategies (the theories encoded for an agent to decide how to act). An agent can also be thought of as a system perceives its environment and acts within it [**?**]. Russell and Norvig go on to explain that an agent maps percept sequences to actions. In the system I am producing this mapping will be provided by forming beliefs based on the percepts an agent receives and then deciding on an action to take based on these beliefs by way of a deductive reasoning theory.

This presents the question how do agents form beliefs based on the percepts they receive. The actual beliefs they will form based on percepts are strategy dependent, but the overriding concept is to (like in deductive reasoning) encode a theory for each strategy as to how agents form their beliefs from percepts.

Percepts cause an agent's beliefs about it's environment and other agents to change at specific timepoints. This is remarkably similar to an approach to reasoning about events (similar to percepts) and time (timepoints) and how events change 'fluents' (similar to beliefs) known as the event calculus, which was presented by Kowalski and Sergot [**?**]. There is an efficient implementation of the event calculus known as the multi-valued

fluent cached event calculus that I plan to use in order to encode theories for strategies on how agents should revise their beliefs based on the percepts they receive.

Beliefs, commitments and capabilities are 3 concepts which I will be incorporating into my system. Commitments to actions based on beliefs satisfy the autonomous property of agents, reactivity to environmental changes are satisfied by constraining capability based on environmental changes, agents can be proactive be committing to actions when possible that bring them closer to their goal and the social ability property can be satisifed by the agents taking social actions. Futhermore these 3 concepts are simple and intuitive to work with.

In my system agents will use their beliefs on other agents' reputations and to commit to actions that they believe will bring their situation closer to their goal. To formulate an idea about other agents reputations agents will receive percepts. Percepts are generated from actions. Actions can be any of the following 5: idle (produces no percepts), gossip positive or negative information to another agent, cooperate or defect. Agents will be constrained capability wise, so that when they are a donor they can only cooperate or defect, but when they are not they cannot cooperate or defect. At any other point when they are not a donor they will be able to gossip or be idle.
The capability of an agent will be constrained at a given timepoint by way of an agent perceiving that they are the donor of a donor-recipient pair at that timepoint. The decision on which agents are donors and recipients in pairs falls to the environment.
Agents must thus compose of 3 general steps. The first of which is perceiving, an agent receives a percept sent from the environment the second step revision of beliefs is triggered by perceiving. The last step is committing to an action based on their beliefs, capabilities and strategy to reach their goal.
Mention why mvfcec is good for beliefs + revision.
End with how my agents satisfy the 4 properties.

ii. Environment

iii. Communication and API Design

III. Discussion and Conclusion

Secur

References

[1] Yoav Shoham. Agent0: A simple agent language and its interpreter. In *AAAI*, volume 91, page 704, 1991.

[2] Professor Kostas Stathis. Lecture notes in intelligent agents and multi-agent systems, November 2018.

[3] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.