Using Agent-based Computational Economics to Model Behaviour under Bounded Rationality

Exam Number: B084774[1,*]

University of Edinburgh

Abstract

This paper reviews the potential advantages of the agent-based computational economics (ACE) paradigm when applied to problems where agents are forced to make decisions under limited rationality. The objective of the paper is two-fold. First, it hopes to theoretically explain why ACE is a good framework for testing behaviour under bounded rationality. It does this by reviewing the ACE methodology and how types of ACE models have successfully been used for this purpose in the past. Second, it formulates an ACE model to test Nash equilibrium play in an N-person Prisoner's Dilemma using adaptive agents. The main purpose of this is to directly apply the methodology in order to highlight the tractability of the ACE approach. It finds that agents do not learn to play the one-shot Nash equilibrium strategy. After many sampling iterations, roughly 70% of the actions they choose are cooperative.

*Word count 9302

[1]I hereby confirm that I wrote this dissertation independently, using only the listed references. I owe many thanks to Professor Ed Hopkins and Dr. Tatiana Kornienko for their valuable comments and suggestions.

1. Introduction

The purpose of this introduction is to frame the agent-based computational economics paradigm in the context of traditional agent-based economic modelling. This will hopefully motivate why the continued development of ACE is important to economic theory in general. It will also hopefully demonstrate how ACE makes the agent-based approach more powerful, specifically highlighting its ability to easily evaluate models involving agents who are boundedly rational. The reason for introducing ACE in this manner is that recent ACE literature tends to have a very interdisciplinary focus. The next section will more rigorously define ACE in the way that it is now typically introduced, which is as an application of complexity science to economics (Tesfatsion, 2001). The aim of this section is to trace the history of ACE, and make its deep-rooted economic motivations and applications clear.

We will begin with the Walrasian equilibrium model, which is generally regarded by economists as the beginning of agent-based economic modelling. Chen (2012) discusses the relationship between ACE and the Walrasian equilibrium model at length and terms it as the "markets origin" approach to understanding ACE. The key observation he makes is that ACE is primarily motivated by the decentralized and imperfect real processes which are abstracted away from Walrasian equilibrium (Chen, 2012). As Tesfatsion (2006) notes, Walrasian equilibrium is powerful because it demonstrates how individually rational market participants interacting through a decentralized pricing mechanism can generate efficient market outcomes. This is a fundamental result in economics and not one that ACE seeks to challenge. ACE models are instead usually concerned with, and excel at, predicting the behaviour of sub-rational agents. They do this by simulating local, direct interactions between agents whose beliefs can be heterogeneous and arbitrarily flawed. This makes the ACE paradigm extremely flexible and well-suited to modelling economic problems involving "asymmetric information, strategic interaction, and mutual learning" (Tesfatsion, 2006).

The next theoretical inspiration behind the development of ACE is Hayek (1945) and his seminal paper about the use of knowledge in society. In this paper, Hayek criticizes both the concept of central planning and the aggregate approach to modelling (Hayek, 1945). A critical observation he makes is that information in society "never exists in concentrated or integrated form, but solely as dispersed bits of incomplete and frequently contradictory

knowledge which separate individuals possess" (Hayek, 1945). It follows that models constructed from the top-down will never truly be able to recreate the underlying mechanics of the economic processes they hope to explain. This is simply due to the fact that aggregate models are functions of aggregated information. However, despite their actual infeasibility, top-down models have very practical qualities which is why they dominant economic theory. Specifically, they are able to make accurate and easily understandable predictions across a range of economic problems. Contrarily, agent-based models have historically been difficult to analyse due to their abstractness, and generally struggle to make concise predictions. ACE remedies these issues through computation; the "executable" nature of ACE models allows results to be verified exactly, step by step (Holland and Miller, 1991). The computational foundation championed by ACE adds a concreteness to the agent-based approach, increasing its feasibility as an effective modelling technique for a range of economic problems. Vriend (2000) makes the insightful observation that ACE is the tangible realisation of much of Hayek's more abstract methodology, especially that which focuses on the emergence of complex order.

Continuing chronologically, we will now address the early development of ACE itself. Schelling (1969) is one of the earliest papers which uses what would later be known as the ACE methodology. In this paper, Schelling presents models of segregation constructed using agents with simple beliefs (Schelling, 1969). His basic model is set up in the following way. A grid is initialized with some number of agents of type 1, some number of agents of type 2, and some empty spaces. All agents effectively have the same belief, which is that they will move location if they are locally neighboured by some proportion of agents of the other type (Schelling, 1969). In general, the proportion required to move can be different depending on agent type, and the number of agent types need not be limited to two. Schelling's model is powerful because it shows that agents repeatedly interacting with this belief produce type segregated clusters. Thus, this minimal assumption is, in theory, all that is required to cause segregation. ACE researchers generally point to Schelling's result as the beginning of ACE modelling, and use the intuitive nature of his model as inspiration.

We have now reached the beginning of the recognised ACE literature. Hopefully this introduction historically motivated the ACE paradigm, and made clear how ACE arises out

of established economic theory. The remainder of this paper will be structured as follows. Section 2 will more formally define the ACE methodology, as well as different types of ACE models. Section 3 will discuss criticism of the ACE methodology and challenges it faces. Section 4 presents an ACE model of an N-person Prisoner's Dilemma to test the robustness of Nash equilibrium (NE) play in the case where agents are provided with extremely limited information. Section 5 interprets the results, finding that agents do not tend to play the rational, one-shot strategy. Section 6 discusses potential limitations of the model presented in Section 4. Finally, Section 7 concludes by emphasizing the importance of ACE modelling, and the advantages it has in modelling behaviour under sub-rational assumptions.

2. Defining ACE and types of ACE models

2.1. Definition

ACE is typically defined using the complex adaptive systems (CAS) framework. Therefore, it is first necessary to describe the CAS framework. CAS are not universally well defined, so in this paper we will adopt Tesfatsion (2006) and draw heavily on her work for the following definitions. A system is "complex" if it is "composed of interacting units" and "exhibits emergent properties (i.e. properties arising from the interactions of the units which are not properties of the individual units themselves)" (Flake, 2000, Tesfatsion, 2006). A complex adaptive system is "a complex system that includes reactive units (i.e. units capable of exhibiting systematically different attributes in reaction to changed environmental conditions)" (Tesfatsion, 2006). The purpose of ACE modelling is to study economic processes computationally as complex adaptive systems.

ACE models are usually specified in the following manner. Firstly, agents are encapsulated with some set of beliefs and provided with an information set. In other words, each agent can be thought of as a complete function which takes some information as an argument (Tesfatsion, 2006). Agents then interact with each other, taking actions according to their beliefs and the information provided. Finally, the system formed through the repeated interactions of agents is studied and interpreted by the modeller (Tesfatsion, 2006). The abstractness of this specification supports a variety of ACE models.

Now it is important to discuss what can be learned from ACE models. These points will be elaborated on in following sections, but it feels necessary to introduce them here so as to motivate the previous definitions.

The main point to be made is that ACE models are very good at incorporating bounds on rationality. The sub-fields of economics where this quality of ACE is probably most useful are behavioural economics and game theory. Specifically, ACE models can be exploited to better understand how behaviour changes on the margins of individual rationality (Levy, 2012). It is straightforward to see how this possible. Since the set of beliefs agents are allowed to possess is arbitrarily large, the modeller simply needs to choose the margin by which agents are to deviate from individually rational behaviour. The deviation is programmed into the agents' beliefs, and the subsequent CAS solution is computed. The CAS solution can then be empirically evaluated against the individually rational solution. ACE is especially powerful because the extent of the deviation from individually rational behaviour can also be arbitrarily complex. This allows for the possibility of deviations which might be incredibly difficult to study using any other method.

A related, follow-up point is that ACE can also be a convenient way to select between equilibria under bounded rationality (Arifovic, 1994). Using the ACE methodology, models can be constructed using agents subject to different rationality constraints. Particular equilibria could then be computed under different bounds on rationality. This would be useful in situations where constraints may be difficult to describe analytically, or equilibrium may be difficult to solve for analytically (Arifovic, 1994).

A final point is that ACE is a methodology that allows economists to perform experiments more similar to that of the natural sciences (Tesfatsion, 2006). This is evident from the isolated, and targeted nature of ACE models. What the modeller hopes to learn greatly depends on the way the experiment is constructed, and the problem that is being studied. For example, in the case of the first point made, the modeller might hope to learn how much utility is lost from a particular deviation from individually rational behaviour. ACE results can potentially be very powerful, but only if the design of the computational experiment is well motivated and results are properly benchmarked.

2.2. Simple rule-based automata models

The first ACE models we will examine in this paper are simple rule-based automata models. These models are popularly implemented as cellular automata, so we will begin by describing this sub-class of models and then generalise. Cellular automata models are usually specified in the following manner. Firstly, agents are created and provided with some set of explicit, static rules which govern their actions in all possible scenarios. A more game theoretic way to express this would be to say that each agent has a complete strategy. Secondly, agents are positioned on a 1D or 2D grid. Finally, agents interact with each other according to their rules (or strategy) and form a CAS (Chen, 2012). These models are referred to as "cellular" because agents are positioned on a grid and occupy cells. More generally, there is no need to position agents and they need only function according to their rules.

Cellular automata models have a long, interdisciplinary history. Schelling's segregation model serves as a good example of a cellular automata model in economics. More broadly, these models have fascinated a number of famous academics, such as von Neumann. The fascination stems from the observation that agents interacting using easy to understand rules can sometimes create aggregate systems that appear extremely complex to humans.

Simple rule-based automata models may be the most intuitive ACE model for economists in general. The reasoning behind this is that the assumptions on agents' behaviour are clear and understandable. Furthermore, the rules dictating agent behaviour can be minimal without necessarily trivializing results. This makes it possible to test the marginal effect of imposing a rule on agents' behaviour, as the impact of the rule can effectively be tested under controlled conditions. Gode and Sunder (2002) realised the potential of this set-up, and popularised the concept of "zero-intelligence" traders. In their model, agents randomly choose between all individually rational options. Agents participate in a double auction and are allowed to submit any bid or ask as long as it does not result in them making a loss. They find that imposing this condition on traders is powerful enough to "raise the allocative efficiency of the auction to close to 100 percent" Gode and Sunder (2002).

While simple rule-based automata models may be insightful in particular cases, they lack the general robustness needed to effective across most sub-fields of economics. For example,

it seems unlikely that a simple automata model would be able to provide any insight into macroeconomic processes. Consider that a simple automaton acting according to static rules is fundamentally unable to respond to changes in policy or generalise behaviour in response to new information. It follows that the model essentially falls under the "Lucas critique", as its predictions are not sensitive to exogeneous, fundamental changes (Lucas, 1976). To some extent this limitation might be overcome by allowing agents to update their beliefs using some learning rule and hence dynamically incorporate new information (Lucas, 1986). This point helps to motivate the next sub-sections on genetic algorithms, reinforcement learning, and neural networks.

2.3. Genetic algorithms

Now we will discuss ACE models constructed using agents evolved through genetic algorithms. Before reviewing the economic applications of these models, it is first important to explain how they work. Genetic algorithms were first conceived by Holland (1970) as a formal framework for "problems of adaptation". They are a set of heuristic optimization techniques which have generally been used in interdisciplinary research across computer science, biology, and economics. Inspired by the theory of natural selection, genetic algorithms use operators such as mutation, crossover and selection to optimize the genetically coded parameters of an agent in order to maximize a fitness function (Mitchell, 1998). Informally, genetic algorithms usually have the following structure. First, a population of agents is initialised, with each agent possessing random genetic parameters. Agents in the population then interact and are scored using a specified fitness function. The best performing agents (top % greater than some threshold) are selected while the rest of the agents are discarded. The surviving agents then undergo crossover and mutation operations to form new agents. The crossover operation concatenates parameters from different surviving agents, and the mutation operator randomly changes parameters of surviving agents. The surviving, crossed over, and mutated agents form the next generation of agents, and the process is repeated indefinitely (Mitchell, 1998).

Genetic algorithms have most commonly been used in economics for two purposes. The first purpose is to test the robustness of models formulated using rational expectations. The

second purpose is to model experimentally observed irrationality in human behaviour. In the first case, researchers are interested in whether the predictions of models formulated using adaptive agents converge to those of models formulated using rational agents. If they do, the experiment strengthens the rational expectations model as a solution to the given problem. In the second case, genetic algorithms are used intuitively to better understand how irrational behaviour might be naturally selected (Crawford, 1991).

Arifovic (1994) demonstrates how adaptive agents refined using genetic algorithms can be used to support rational expectations theory. In her study, she formulates various cobweb models based on genetic algorithms, and through simulations demonstrates that in some cases predictions converge to the rational equilibrium outcome (Arifovic, 1994). The first cobweb model was formulated by Ezekiel (1938); it investigates how firms make production decisions for a single good when prices are taken as exogeneous and not observed until after the production decision is made. Since then the problem has been extensively modelled using the rational expectations approach. The rational expectations approach assumes agents possess the rational belief that the market price in the next period will be the same as its expected value from the previous period (Muth, 1961). It follows that agents implicitly incorporate all available information from the previous period into their production decision, and hence choose production optimally given their information set. The major contribution that Arifovic (1994) makes is that she shows the equilibrium formed under rational expectations can be achieved without the assumption that agents possess the rational expectations belief. Adaptive agents evolved using genetic algorithms can produce the rational expectations equilibrium solely using feedback from the environment. The result demonstrates that agents can learn optimal production quantities without knowing how to explicitly maximise their objective function (Arifovic, 1994). Since people do not usually explicitly maximise when making decisions, the genetic algorithm framework seems more intuitive to how people actually make decisions.

Andreoni and Miller (1990) use agents evolved through genetic algorithms to study "bidder errors" commonly observed in real auction experiments. One of the errors that they attempt to model is the "winner's curse" in common value auctions. The "winner's curse" refers to the experimental result that the winner of a common value auction usually overbids,

failing to account for the fact that there is information in being the highest bidder (Andreoni and Miller, 1990). Andreoni and Miller (1990) set up their model as follows. First, the common value x_0 is drawn random uniformly from the interval $[1000, 2000]$. Bidders are then each given a signal x_i about the common value, x_i is drawn random uniformly from the distribution $[x_0 - \epsilon, x_0 + \epsilon]$, where ϵ is drawn random uniformly from $(0, 500]$ (Andreoni and Miller, 1990). Kagel and Levin (1986) show that the Nash Equilibrium (NE) bid function for this model is given by:

$$b(x_i) = x_i - \epsilon * (\frac{n-2}{n}) \tag{2.1}$$

Andreoni and Miller (1990) test how closely the bids of agents selected using genetic algorithms approximate the NE bid function. They evolve different sets of agents; some learn the 4-bidder case, while others learn the 8-bidder case. In both cases they find that agents overbid, but that agents play closer to the NE in the 8-bidder case. On average in the 4-bidder case agents bid above the NE by 11.65, while in the 8-bidder case they bid above the NE by 3.15 (Andreoni and Miller, 1990). This result does not support experimental studies which usually find that the inclusion of more participants results in bids further from the NE. However, their model is interesting because it endogenously selects agents who overbid. It is a good example of how an adaptive, agent-based framework can be used to model irrational behaviour.

Models constructed using genetic algorithms are popularly used in the ACE literature. Their popularity probably stems from the intuitiveness of the theory of natural selection. In the next two sub-sections we will examine more heuristic ACE models, specifically those based on reinforcement learning and neural networks.

2.4. Reinforcement learning

This sub-section will concentrate on ACE models which use agents trained through reinforcement learning algorithms. The modern field of reinforcement learning can be traced to the 1980s with the fusion of optimal control theory, trial and error learning from psychology, and temporal difference methods. In the context of ACE, reinforcement learning is a promising approach for incorporating adaptive behaviour into computational agent models. Intuitively, reinforcement learning is focused on goal-directed learning stemming from

repeated interactions with the environment. Agents' beliefs are now normally represented using Bellman equations; these are updated iteratively in response to new information (Sutton and Barto, 1998).

Arthur (1993) is a good example of reinforcement learning in the ACE literature. His model blends artificial and experimental results in an attempt to create human "calibrated" artificial agents (Arthur, 1993). He uses the N-armed bandit problem as a toy problem to test his methodology. This is a central problem in reinforcement learning and intuitively demonstrates how reinforcement learning agents form beliefs, so before explaining his approach we will briefly divert to explain its importance. The N-armed bandit problem is posed as follows. A participant is given the choice between N options. Each option provides feedback to the participant using rewards drawn from a unique, hidden probability distribution. The participant's only goal is to maximise the reward earned, and he updates his preference over the choices using information gathered from previous observations. Studies of this problem usually centre their analysis on how choice governing policies evolve over time, hoping to learn something about how participants make decisions (Arthur, 1993).

The problem is more well-studied in psychology and computer science, but variants have been shown to possess economic applications. For example, Rothschild (1974) demonstrated the feasibility of the two-armed bandit problem as a pricing model for firms which attract steady, unrepeated business (-e.g. retail firms in airports, high-end appliance firms). His model assumes that these firms inherently start with poor knowledge of the demand functions of their customers. In each period, firms choose whether to set prices high or low. They are able to observe the profit earned after making their decision, and use that to adjust their pricing preference for future periods. Over time firms learn good pricing strategies, implicitly learning about demand for their good (Rothschild, 1974).

Now we will return to Arthur (1993). The first stage of his model trains an artificial reinforcement learning agent to play the N-armed bandit game and records the results. The agent is defined and updates his beliefs as follows. First, the agent is initialised as a vector of strengths. This vector represents values the agent associates with choosing each of the N possible actions. Each period, the probability of choosing a specific action is computed by dividing the value associated with that action by sum of all values in the strengths vector.

The agent randomly chooses an action according to these probabilities. He then observes the reward and adds the observed value to the strengths vector for the chosen action. The strengths vector is then renormalized to sum to a pre-determined constant (Arthur, 1993).

The next stage of his model focuses on calibrating the behaviour of the artificial agent. The goal of his study is to create artificial agents that exhibit behaviour which is statistically similar to the behaviour of humans. The way that he does this is as follows. He first compares the results generated by the artificial agents with those generated by students in an experimental study. He then fits a least squares model to minimise the difference between the choices of the artificial agents, and those observed in the student trials. The expectation is that the least squares model will capture the error between the artificial and experimental trials. The reinforcement learning and least squares models can then be combined to create agents whose behaviour exhibits features observed in experimental data (Arthur, 1993).

His results show that these agents are able to produce some of the stylized trends observed in the experimental data. For example, he finds that agents tend to "meliorate" in the same way that humans do (Arthur, 1993). They overwhelmingly choose the action that has given them the highest average payoff over some number of previously observed trials. This is often a sub-optimal strategy because it prevents them from discovering that other options might draw rewards from better distributions.

Reinforcement learning agents are interesting because they learn from rewards attained through direct interaction. There is no assumption that agents will learn to behave rationally; the behaviour that they learn can be arbitrarily irrational. This makes them useful for modelling boundedly rational behaviour. Agents can learn internal parameters which make them limitedly rational.

2.5. Neural networks

This sub-section will discuss agents represented by neural networks. Artificial neural networks (ANN) are compositional models which can be trained to approximate high dimensional nonlinear functions given example data. Early research in neural networks was focused on small models with limited practical applications. Since 2012, however, deep learning (DL) has been used to achieve state-of-the-art results in various fields such as image

and speech recognition, natural language processing, and robotic control. The availability of large amounts of digital data and computational power is generally considered to be the enabling technology behind recent success in DL (Schmidhuber, 2015). ANN models are trained using backpropagation and stochastic gradient descent (SGD) algorithms in order to learn a set of weights which minimize a specified loss function.

In this section, we will discuss how ANN are used in the ACE literature to evaluate boundedly rational behaviour. In the context of ACE, ANN almost exclusively concentrate on predicting the behaviour of agents (e.g. other ACE models, humans). They do not attempt to explain the behaviour; their purpose is to learn a general function that well approximates it.

This leads to the main point, which is that ANN can be used to emulate the behaviour of economic agents interacting in some environment. Van der Hoog (2016) refers to ANN which are used this way as "Doppelgangers". The idea that ANN might be useful in emulating individual behaviour stems from the universal approximation theorem. Informally, this theorem essentially states that under some structural assumptions, ANN are capable of approximating any real and continuous function. A good use of ANN is as a tool to evaluate other ACE models. This evaluation is done as follows. First, formulate some other ACE model and collect simulated data from agents' behaviour. Next, fit an ANN model to make predictions based on the simulated data. Then validate the predictions of the ANN model by forcing it to make predictions using a real dataset. The extent to which the ANN can make useful predictions on the real dataset is a measure of how much real predictive power the simulated dataset has (Van der Hoog, 2016). In other words, it measures how well the system formed by the ACE model predicts the real world system which is being studied. Since agents in the ACE model can possess sub-rational beliefs, this emulation method is potentially useful for evaluating how well systems formed by boundedly rational agents predict those observed in the real world.

Another point is that ANN agents can also naturally be fit using reinforcement learning techniques, or on human generated data. In the context of ACE, the purpose of ANN agents which are fit this way is usually to generate good policy. Van der Hoog (2016) gives an example where the ANN agent is asked to set policy for a central bank. In his example,

the ANN agent observes historical macroeconomic data and is tasked with certain goals. For example, these goals could be to maintain a low unemployment rate and a stable rate of inflation. His argument is that an ANN agent might be able to implicitly learn more nuanced relationships between macroeconomic variables than are described by "hand-crafted" rules (-e.g. Taylor rule) (Van der Hoog, 2016). In this example, the ANN agent is bounded in that he can only form beliefs based on past data. This is a severe constraint, as the agent's prediction will quickly become very bad if the data is drawn from a non-stationary distribution. One way this might be overcome is through online learning, where present data is continuously used to update the fit of the model. In the present this is not very practical as ANN are very computationally intensive to fit. As computation continues to become powerful, this approach might become more feasible.

Out of all of the ACE models described in this section, ANN are probably the least commonly used in economics. In the future this might change as their prevalence continues to increase in other fields. In the next sub-section, we will focus on agents constructed from econometric foundations.

2.6. Econometrics-based agents

One major objective in the ACE literature is to specify agents using econometric methods in order to increase their understandability. This sub-section will discuss two ways that this is done. The first way is to specify agents such that their internal parameters can be meaningfully interpreted. This makes it easier to understand which assumptions are driving their behaviour since there is some estimate associated with each assumption. The second way is to augment agents' behaviour after initially observing it. This is done when the modeller is interested in using artificial agents to produce features of real data.

Chen et al. (2012) demonstrates how agents' beliefs can be specified and interpreted using a binary logit model. The example they give is inspired by computational finance. In each period, agents choose whether they want to be a fundamentalist or a chartist. Either choice will provide some utility to the agent. The utility gained is random, so the agent should choose randomly between the two options. As Chen et al. (2012) notes, if agents' beliefs are represented using a logit model, then the probability of an agent choosing to be

a fundamentalist is equal to the probability that the utility earned from choosing to be a fundamentalist is greater than that of choosing to be a chartist. They derive the probability of the agent choosing to be a fundamentalist as:

$$Prob(X = f, t) = \frac{exp^{\lambda V_{f,t-1}}}{exp^{\lambda V_{f,t-1}} + exp^{\lambda V_{c,t-1}}} \tag{2.2}$$

In (2.2), the deterministic and random components of the agent's choice are isolated. $V_{f,t-1}$ is the deterministic utility gained from being a fundamentalist, $V_{c,t-1}$ is the deterministic utility gained from being a chartist, and lambda is a random component. They further show that (2.2) can be rearranged such that:

$$Prob(X = f, t) = \frac{1}{1 + exp^{-\lambda(V_{f,t-1} - V_{c,t-1})}} \tag{2.3}$$

The interpretation of (2.3) is that the agent's choice between whether to be a fundamentalist or a chartist should depend on the difference between the deterministic components. The reformulation makes this concept more visible. Now, returning to the original purpose of detailing this model, it will be explained as to how beliefs can be interpreted. The parameter to be interpreted in this model is λ. Chen et al. (2012) refers to λ as the "intensity of choice". It measures the sensitivity of the agent's choice between fundamentalist and chartist to changes in the deterministic components of utility gained from being either a fundamentalist or a chartist (Chen et al., 2012).

Now we will examine ways in which econometric methods can be used to make agent behaviour more realistic. These methods typically compare the dataset created by agent behaviour with a real dataset. The aim of these methods is to augment the results of the simulated data so that some of its features are indistinguishable from real data. The techniques used to do this are known as simulation-based econometric methods. Chen et al. (2012) lists some examples of these methods, such as the method of simulated moments and the method of simulated scores. We will focus on explaining the method of simulated moments as it seems more intuitive. First proposed by McFadden (1989), the general idea is as follows. First specify the agent-based model, execute it, and collect the results. Compute the moments of the simulated results, and compare them with those of the real dataset which is being modelled. Now the goal is to fit a new model which minimises the difference

between the real and simulated moments. The data generated by this model can then be added to the original simulated data. The hope is that this process effectively corrects agent behaviour ex post so that it better captures features observed in real data.

Econometrics based agents are promising because they are interpretable and easy to understand. These are two points which are normally used to criticise simulation-based analysis in economics. This criticism will be discussed in the next section.

3. Known Limitations

Thus far this paper has focused on the potential advantages of using the ACE methodology. It has argued that the ACE approach is compelling for multiple reasons. Firstly, it is a consistent method to easily evaluate models where agents possess relatively descriptive beliefs that would be difficult to capture analytically. Secondly, ACE could be a "truer" way to model decentralised processes due to its use of individual, stylised interaction. Finally, following from the first point, ACE models are good at predicting the behaviour of agents possessing potentially complex, sub-rational beliefs. This property of ACE models makes them useful as a way to study irrationality in human decision making.

Now the focus of the paper will shift to address weaknesses in the ACE methodology. ACE has many shortcomings, and is criticised for a variety of reasons which will be discussed in this section. Before discussing this criticism, it is first important to quickly note the role of ACE within economics as a whole. As has been shown by the studies examined in Section 2, ACE models can be used to make robust predictions about economic behaviour in certain cases. However, in a larger sense, ACE models serve as powerful supplements to analytical microeconomic models. They are often easily visualised and may be a convenient way for economists to gather intuition about the effect of a particular behavioural assumption without extensive derivation. The rest of this section will discuss the following two main criticisms of the ACE methodology. These are that ACE models lack explanatory power and transparency (GräBner, 2016, Richiardi, 2003).

One of the biggest problems with ACE from an empirical point of view is that it lacks compatibility with standard econometric analysis. Although most ACE models use parameterized learning algorithms, these parameters usually have no real meaning. Most empirical

work in economics focuses on using econometric techniques to establish significant relationships between real variables. ACE models need to be empirically evaluated in order to be interpreted, but are not typically conducive to econometric analysis because the internal variables governing agent actions are normally arbitrarily assigned. There are few specialised methods for inferring causality in ACE models, so ACE analysis is limited in that it is mostly reduced to statistical tests relative to some benchmark.

This limit on ACE analysis signals bigger problems with the methodology. Broader criticism argues that ACE models are difficult to generalise, and unable to prove anything (Leombruni and Richiardi, 2005). Criticism about generality focuses on highly parameterised ACE models. The concern centres on the fact that the values these parameters converge to can be highly sensitive to their initial values. Therefore, different fits of the model could produce distinctly different results simply because of their initial conditions. This is a major reason that economists are skeptical of the ability of ACE simulations to really explain anything. As a result of this skepticism, a major research objective in the ACE literature has been to develop more precise models in an attempt to gain explanatory power. As has been discussed in Section 2.6, active research concentrates on building ACE models from econometric foundations (Chen et al., 2012). A goal of this research is to specify the internal parameters of agents in such a way such that the magnitude and significance of particular assumptions can be inferred.

Another problem with ACE models in economics is their lack of transparency. As can be noted from the models detailed in Section 2, many types of ACE models are built from machine learning foundations. The field of machine learning largely focuses on maximising the predictive power of models, often at the cost of understandability. One issue with using these types of ACE models in economics is that they are often arbitrarily specified (GräBner, 2016). In contrast to econometric models where model specification can be well justified, the number of structural parameters included in machine learning models usually cannot be. This tends not to be a significant issue in machine learning literature, as structural parameters can be tuned through trial and error in response to in response to predictive indicators (-e.g. logarithmic loss, area under ROC curve, etc.). However, in the context of the ACE literature this is a major problem. This is due to the fact that ACE models are

more concerned with understanding how behaviour adapts under particular assumptions, and not necessarily with optimising for a particular outcome. Therefore, it is often difficult for ACE modellers to justify the structure of their models. This decreases the transparency of their results because, in this context, the structure of the model can effectively be viewed as an additional, uninterpretable assumption.

This section summarised two common critiques of the ACE methodology. These reasons largely capture why the ACE approach has not been popularly adopted in economics. The purpose of including this section was to acknowledge valid criticism. ACE is not a standard approach and it is important to understand why so that the methodology can continue to develop constructively. In the future, some of these issues might be overcome through interdisciplinary research between machine learning and econometrics.

4. Building an ACE model of an N-person Prisoner's Dilemma

4.1. Characterising N-person Prisoner's Dilemmas

The purpose of this sub-section is to explain how N-person Prisoner's Dilemmas (NPPD) are formulated, and why they are important to economics. NPPD are generalisations of the classic two-person Prisoner's Dilemma. They are not well-defined and can be set up differently depending on the goals of the modeller. We will begin by describing a common formulation of the NPPD, and discuss its applications in economics. Then we will specialise further by detailing the spatial NPPD, and discussing why ACE is particularly well suited to modelling this problem.

NPPD are normally specified as follows. Let $N = \{1, 2, ..., n\}$ be the set of players. Let action $A = \{C, D\}$, where C=cooperate and D=defect. Each player $i \in N$ chooses an action $a \in A$. Payoffs are then computed as function of overall levels of cooperation; Okada (1991) uses the form $f(a, h)$, where h is the total number of co-operators. In order for this this game to form the Prisoner's Dilemma, $f(a, h)$ must satisfy the following conditions:

$$f(D, h) - f(C, h) > 0 \text{ and constant for } h \in [0, n-1] \tag{4.1}$$

$$f(C, h) \text{ is monotonically increasing in } h \text{ for } h \in [0, n-1] \tag{4.2}$$

$$f(C, n-1) > f(D, 0) \tag{4.3}$$

Okada (1991) provides interpretations for these conditions. Condition (4.1) ensures the dominance of the defect strategy in the one-shot game. Condition (4.2) demonstrates that the return to cooperation increases in the number of co-operators. Condition (4.3) shows that the NE where all N players defect is Pareto dominated by the case where all N players cooperate.

One common application of this NPPD is as a model of the tragedy of the commons problem. The tragedy of the commons problem is first described by Hardin (1968). The problem can be summarised as follow. Imagine there is a public pasture that anybody can use to let their animals graze. People get extra utility out of letting more animals graze. The more animals that graze, the more depleted the land becomes. Each person maximises their own utility, too many animals graze, and the land becomes depleted such that no animals can graze (Hardin, 1968). It is straightforward to see how the description of this problem fits within the formulation of this NPPD. People have the choice to either not let another animal graze (cooperate), or to let another animal graze (defect). Each person receives a higher payoff from letting another animal graze as long as sufficiently few other people also choose to let an additional animal graze. They are all better off if nobody lets an additional animal graze than if they all let an additional animal graze. The tragedy of the commons problem serves as an example of how the NPPD can model a real social dilemma.

Now we will concentrate on spatial NPPD where agents are located on a graph (Fosco and Mengel, 2011). The spatial NPPD possesses the following setup. First, let $N = \{1, 2, ..., n\}$ be the set of players. Let $A = \{cooperate, defect\}$ be the set of possible actions. Each player $i \in N$ chooses an action $a \in A$ for each opponent he is linked to. Players play the two-person Prisoner's Dilemma with each linked neighbour. They receive the standard two-person Prisoner's Dilemma payoffs for each game they play. Spatial NPPD are interesting because they allow for the possibility of local dynamics. Researchers studying spatial NPPD are usually interested in understanding how clusters of cooperative behaviour form (Ifti et al., 2004). The specifications of spatial NPPD problems are intuitively modelled using the ACE methodology. Analysis of these models normally explicitly concentrates on the system formed by interacting agents.

4.2. ACE studies of the Prisoner's Dilemma

Different versions of the Prisoner's Dilemma have been studied in the ACE literature using boundedly rational agents. The purpose of these studies is generally to determine the extent to which agents can learn rational NE or sub-game perfect equilibrium (SPE) outcomes without imposing any assumptions of rationality on their behaviour. Alternatively, they hope to see if the boundedly rational assumptions can support outcomes which are Pareto superior to the one-shot NE.

Miller (1996) studies the iterated, two-person Prisoner's Dilemma using agents evolved through genetic algorithms. Consider that Folk theorems generally state any feasible and individually rational outcome can be supported as a sub-game perfect equilibrium (SPE) in an infinitely repeated game with sufficient discounting (-e.g. (Friedman, 1971, Abreu et al., 1994)). Miller (1996) is interested in whether the assumption that agents are adaptive is sufficient to support SPE outcomes which include cooperative strategies. Agents use information about the previous moves of their opponent in order to decide which action to play in each period. Miller (1996) utilises the possibility of misreported moves to further constrain agents, and test how their behaviour adapts to different conditions on the information set. He simulates agents' behaviour under the conditions where no moves are misreported, 1% of moves are misreported, and 5% of moves are misreported. In general, he finds that agents play the one-shot NE in every subgame more often when the information provided is noisier. His interpretation of this is that agents are more likely to punish their opponent in a noisy environment, due to false reports of defection (Miller, 1996). Assuming agents are able to adapt enough to approximate trigger strategies, this explanation seems reasonable enough. In environments where moves are not frequently misreported, his results are less clear; cooperation can sometimes be observed as a sub-game outcome over many iterations of sub-games, but not always.

Fosco and Mengel (2011) examine agent behaviour in a spatial NPPD where homogeneous agents interact according to stylised rules. These rules essentially specify the strategy all agents will follow. In their model, agents make both linking and action decisions in each period. They make these choices using imitation rules; they copy the choices of neighbouring agents who achieve relatively high average payoffs. Agents are constrained in that they are

only allowed to have up to a specified maximum number of links at any given time. Their model is interesting because agents form a dynamic network. Their analysis primarily focuses on the coevolution between choice of action and the topology of the network. They find that there are two interesting outcomes. In the first outcome, agents essentially become fully segregated by choice of action. In the second outcome, most agents cooperate and agents who defect are marginalised to the fringes of the network (Fosco and Mengel, 2011). These results demonstrate that cooperation can be achieved when agents choose their interactions based on an imitation strategy. Furthermore, it shows that the imitation assumptions are strong enough to isolate agents who choose to defect.

This sub-section has reviewed two ACE studies of the Prisoner's Dilemma. In conjunction, these studies serve as good background for the ACE model of the spatial NPPD proposed in the next sub-section.

4.3. Model

This sub-section will detail the spatial NPPD model used in this paper. This will be done in two parts. First the agents and set up of the game will be described. Then limitations on the rationality of agents will be discussed.

All agents in this model are homogeneous. They are specified using two deep neural networks. The structure of these networks is available in Appendix A.1. Agents update their beliefs using reinforcement learning. More technically, the algorithm they use is known as Double Deep Q-learning (DDQN). This algorithm emerges from relatively recent work in the field of deep reinforcement learning (Hasselt et al., 2016). The aim of this field is to combine recent advances in deep neural networks with reinforcement learning techniques in an attempt to create more intelligent agents. Intuitively, DDQN agents are neural network approximations of values associated with choosing actions in a particular state. The DDQN algorithm is outlined here, following the work of Hasselt et al. (2016). First, start by initialising two neural networks. Both of these networks estimate values over the set of actions for the particular state. The first network chooses the action of the agent; this is the action in the set with the highest estimated value. The estimated value is the agent's prediction of the payoff he will receive in the next state (or the value associated with being in the current

state). The second network independently estimates the value for the action chosen by the first network. This estimate is used in setting a target value for the first network. The target is given by Hasselt et al. (2016) as follows:

$$Y_t^{DDQN} \equiv R_t + \gamma Q(S_{t+1}, \underset{a}{\text{argmax}}\, Q(S_{t+1}, a; \theta_t), \theta_t^-) \tag{4.4}$$

In equation (4.4), S_{t+1} is the state in period $t+1$, R_t is the observed reward in period t, γ is the discount factor, θ_t are the weights of the first network (action network), $Q(S_{t+1}, a, \theta_t)$ is the estimate by the action network, θ_t^- are the weights of the second network (target network), $Q(S_{t+1}, a, \theta_t^-)$ is the estimate by the target network. The specification of this target combines a real component, the actual reward observed in the current period, with a discounted future estimate of the reward in the next period. Action network weights θ_t are updated iteratively using:

$$\theta_{t+1} = \theta_t + \alpha(Y_t^{DDQN} - Q(s_t, a, \theta_t))\nabla_{\theta_t} Q(s_t, a, \theta_t) \tag{4.5}$$

Target network weights θ_t^- are updated iteratively using

$$\theta_{t+1}^- = \theta_t^- + \alpha(Y_t^{DDQN} - Q(s_t, a, \theta_t^-))\nabla_{\theta_t^-} Q(s_t, a, \theta_t^-) \tag{4.6}$$

In (4.5), θ_{t+1} are the neural network weights in the next period, θ_t are the neural network weights in the current period, α is the learning rate, $\alpha(Y_t^{DDQN} - Q(s_t, a, \theta_t))\nabla_{\theta_t} Q(s_t, a, \theta_t)$ is the gradient of the error term with respect to each weight. Equation (4.6) is the counterpart for the target network weights θ_t^-.

The technical interpretation of equations (4.5) and (4.6) is that the weights of the neural networks are updated by backpropagating the error between the target value and the network predicted values. Intuitively, agents revise their internal beliefs about the value of being in a particular state based on observed outcomes. The expectation is that the neural network approximations should converge to reasonable value estimates over many iterations.

The spatial NPPD is implemented as follows. Agents are located on a torus, and play the two-person Prisoner's Dilemma with their neighbours. Each agent has four neighbours. This is visualised in the following Figure 4.1. In Figure 4.1, red, green, blue, purple represent

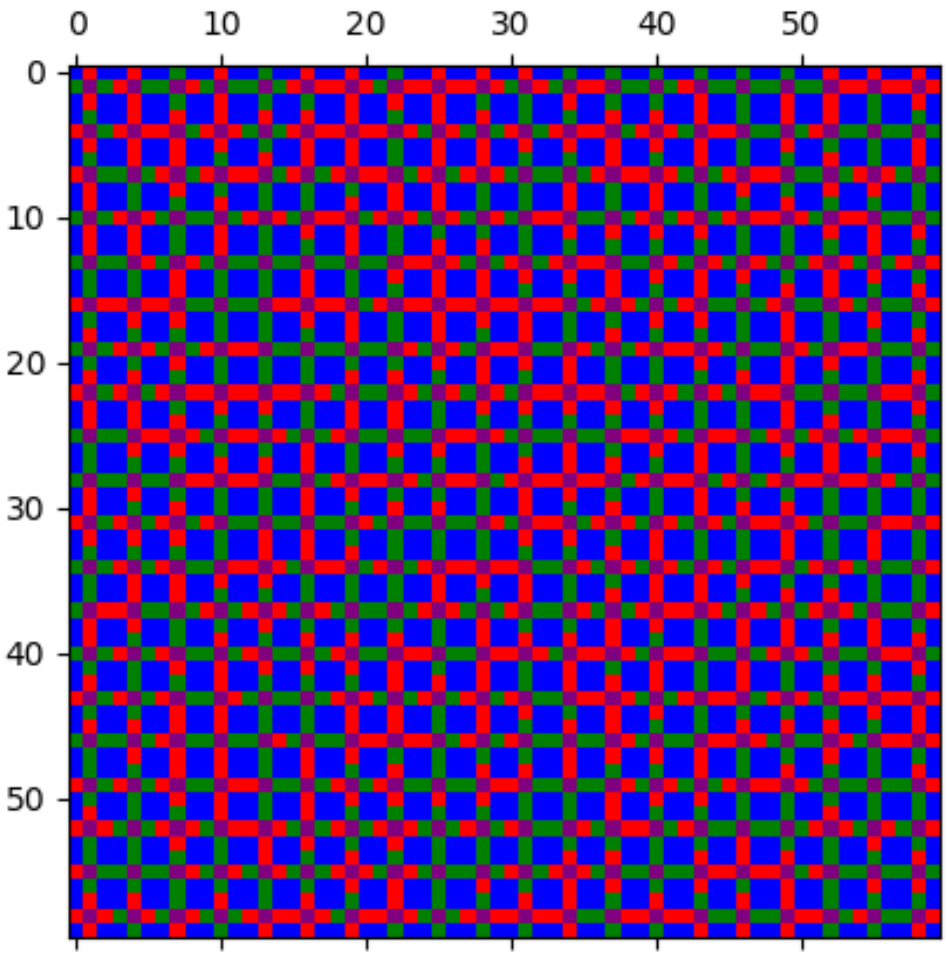

Figure 4.1: Toroidal grid, $\{red = defect, green = cooperate, blue = noise, purple = agent\}$

defect, cooperate, noise, and agents respectively. Noise is included so as to preserve the spatial structure, and also to decrease the signal provided to agents. The values for noise, and agents are drawn randomly from $U(0, 1)$. The value for defect is 0. The value for cooperate is 1. Each period, each agent observes the 6x6 section of the grid which directly surrounds him. More technically, this 6x6 section is the input into the deep neural networks. It includes the observed moves of other agents within this vicinity, as well as noise. Agents use this section to jointly choose an action a for each neighbour from the set of all possible actions $A = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$.

This set represents all 16 possible combinations of actions the agent can choose as 4-bit strings. The first bit is the action for the neighbour to the left, second bit is the action for the neighbour to the right, third bit is the action for the neighbour above, and fourth bit is the action for the neighbour below. A more concise explanation is that the action network maps the input section to one of these 16 elements in A. The payoff matrix is specified so that it satisfies the conditions necessary for this problem to be the Prisoner's Dilemma. It is shown in the following table:

		Player 2	
		C	D
Player 1	C	$(5, 5)$	$(0, 10)$
	D	$(10, 0)$	$(1, 1)$

Table 4.1: Payoff Matrix

In Table 4.1, Player 1's payoff is listed first and Player 2's payoff is listed second. Agents observe the total payoffs achieved from all four neighbours they play. The code used to implement the agents and setup of this model has been made publicly available [2].

This agent implementation was chosen primarily because it makes agent behaviour adaptive; agent behaviour changes in response to experience. A different adaptive algorithm could have been chosen (e.g. genetic algorithms), but this one is interesting because it draws on recent interdisciplinary research between neural networks and reinforcement learning. Agents are given virtually no information about the game, they are only allowed to observe achieved payoffs and very noisy reports of the moves of other agents in their vicinity. They are attempting to optimise their choices using a very weak, and noisy signal. The purpose of this model is to see if, under these conditions, the estimated value of being in any state converges to the value agents would achieve if they all played the always defect strategy (i.e. the algorithm converges to the one-shot NE).

[2] `https://github.com/mattliston/undergraduate_dissertation/blob/master/model.py`

5. Results

Before analysing the results of the model, it is first important to note some practical details regarding how it was fit. The model was fit using 80 million random samples (80M 6x6 sections). The hyperparameters α, γ were specified as $0.0001, 0.1$ respectively.

The analysis in this section will centre on estimates of mean cooperation, mean predicted values, and how they change as the model fits more samples. Starting with mean cooperation, results are depicted in 5.1.

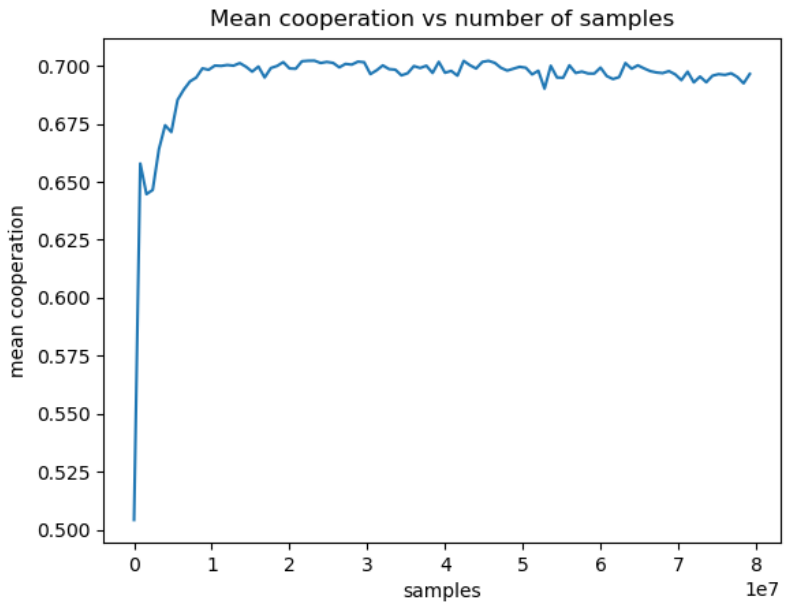

Figure 5.1:

Mean cooperation is computed as follows. First, 100 60x60 random sample states are initialised (see Figure 4.1). For each of these states, the agents' responses are computed. The agents' responses are used to compute the next state. Sample mean cooperation is computed as the number of cooperative actions / total number of actions in this next state. This is an average measure as each state contains many agents. This is done 100 times to form a sampling distribution of mean cooperation. Since the model parameters were saved every 800,000 samples, this process is repeated every 800,000 samples to see how cooperation changes as more samples are fit. Figure 5.1 records the mean of the sampling distribution every 800,000 samples. The standard error of the mean was computed for all sampling

distributions, but is very small in all cases. For each sampling distribution, the standard error of the mean is available in Appendix A.2.

Figure 5.1 shows that after many sampling iterations, about 70% of actions chosen by agents are cooperative. This implies that agents are not learning to play the one-shot NE strategy. They are, however, learning to play the one-shot game. This is clear as agents are just trying to learn whether there is more value in playing cooperate or defect for each neighbour in response to some minimal information set. They are not playing the iterated game with their neighbours because they learn from randomly generated states, not using moves explicitly generated by other agents. They are not learning to play the game sequentially. They are essentially just trying to learn what the best response is for the one-shot Prisoner's Dilemma.

Now we examine the value estimates agents make. Figure 5.2 shows the mean value estimates which govern the actions of agents.

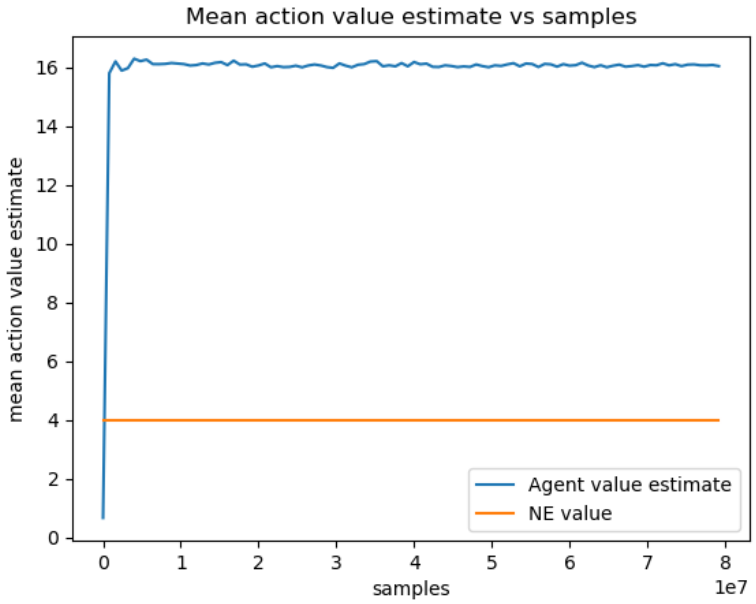

Figure 5.2:

These mean value estimates are computed using the same sampling distribution process which was used to compute mean cooperation. The means of the sampling distributions are what is depicted. For each sampling distribution, the standard error of the mean is very small. These estimates are available in Appendix A.2. Figure 5.2 shows that on average

agents estimate the value of being in any particular state is four times the value they would achieve if they played the rational one-shot NE strategy. It is not surprising that they have high value estimates given that there are high levels of mean cooperation. It is more difficult to tell why agent estimates converge to this exact value. It could just be that estimates fall into a positive reinforcement loop. If this is the case, the model could have been made more interesting by introducing some degree of heterogeneity into agents' behaviour. It could also be overcome by randomly misreporting achieved payoffs in a manner similar to Miller (1996); this would make it much more difficult for agents to make value estimates.

6. Limitations

The results of this experiment could have been made more robust if more models were fit as an ensemble. For example, if 100 independent models were trained on 80M samples. This was not feasible due to constraints on time and computational resources. In retrospect, these constraints could have been overcome by reducing the number of samples per model, and increasing the number of models. The issue this would overcome is the criticism that the convergence of the model is difficult to predict, and that it could converge vastly differently depending on the initial weights of the neural networks.

Another potential limitation is that the input into the neural networks might have been too noisy. The state agents observe could have been constructed with a higher signal. Rather than observing 2D input, agents could have observed a collapsed 1D report of actions in their vicinity. This would have been a more flexible implementation, because it allows the amount of noise agents observe to be varied more easily. It is not reasonable to expect agents to meaningfully map states that include large amounts of noise.

7. Conclusion

Much of economic theory focuses on modelling choices or decisions under some assumption of rationality. One major reason behind this is that assumptions of rationality generally increase the tractability of the problem being studied. This is due to the fact that they allow models to be evaluated using well understood optimisation techniques. The issue is that experimental and empirical evidence often shows that assumptions of rationality are not

realistic. However, when these assumptions are removed, models can become hard to understand or predictions become difficult to evaluate. Agent-based computational economics is a framework that in principle allows problems to remain tractable even when all assumptions of rationality are removed. Models are guaranteed to be deterministically evaluated due to the nature of computation. They can be understood in terms of both the behavioural assumptions programmed into agents, and the extent to which predictions differ from the rational equilibrium. Furthermore, this paper presents a working ACE model of an N-person Prisoner's Dilemma. It tests the extent to which agents learn to play the one-shot Nash equilibrium strategy under adaptive behavioural assumptions and severe information constraints. It finds that under these conditions, agents are unable to identify the dominant, rational strategy.

References

L. Tesfatsion, Introduction to the special issue on agent-based computational economics, Journal of Economic Dynamics and Control (2001). doi:10.1016/S0165-1889(00)00027-0.

S. H. Chen, Varieties of agents in agent-based computational economics: A historical and an interdisciplinary perspective, 2012. doi:10.1016/j.jedc.2011.09.003.

L. Tesfatsion, Chapter 16 Agent-Based Computational Economics: A Constructive Approach to Economic Theory, 2006. doi:10.1016/S1574-0021(05)02016-2.

F. Hayek, The use of knowledge in society, in: The Economic Nature of the Firm: A Reader, Third Edition, 1945. doi:10.1017/CBO9780511817410.007. arXiv:arXiv:1011.1669v3.

J. H. Holland, J. H. Miller, Artificial Adaptive Agents in Economic Theory, The American Economic Review (1991).

N. J. Vriend, Was Hayek an Ace?, 2000. doi:10.2139/ssrn.185650.

T. C. Schelling, Models of Segregation, The American Economic Review (1969). doi:10.1126/science.151.3712.867-a. arXiv:00028282.

G. W. Flake, The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation, 2000. doi:10.2307/2589369.

M. Levy, Agent based computational economics, in: Computational Complexity: Theory, Techniques, and Applications, 2012. doi:10.1007/978-1-4614-1800-9_2.

J. Arifovic, Genetic algorithm learning and the cobweb model, Journal of Economic Dynamics and Control (1994). doi:10.1016/0165-1889(94)90067-1.

D. K. Gode, S. Sunder, Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality, Journal of Political Economy (2002). doi:10.1086/261868.

R. E. Lucas, Econometric policy evaluation: A critique, Carnegie-Rochester Confer. Series on Public Policy (1976). doi:10.1016/S0167-2231(76)80003-6.

R. E. Lucas, Adaptive Behavior and Economic Theory Adaptive Behavior and Economic Theory*, Source: The Journal of Business The Behavioral Foundations of Economic Theory (1986).

J. Holland, Robust algorithms for adaptation set in a general formal framework, 1970. doi:10.1109/sap.1970.270009.

M. Mitchell, An Introduction to Genetic Algorithms (Complex Adaptive Systems), The MIT Press (1998). doi:10.1016/S0898-1221(96)90227-8.

V. P. Crawford, An "evolutionary" interpretation of Van Huyck, Battalio, and Beil's experimental results on coordination, Games and Economic Behavior (1991). doi:10.1016/0899-8256(91)90004-X.

M. Ezekiel, The Cobweb Theorem, The Quarterly Journal of Economics (1938). doi:10.2307/1881734.

J. F. Muth, Rational Expectations and the Theory of Price Movements, Econometrica (1961). doi:10.2307/1909635. arXiv:arXiv:1011.1669v3.

J. Andreoni, J. H. Miller, Auctions With Adaptive Artificially Intelligent Agents (1990).

J. H. Kagel, D. Levin, The winner's curse and public information, American Economic Review (1986).

R. S. Sutton, A. G. Barto, Sutton & Barto Book: Reinforcement Learning: An Introduction, Technical Report, 1998. doi:10.1109/TNN.1998.712192. arXiv:1603.02199.

W. B. Arthur, On designing economic agents that behave like human agents, Journal of Evolutionary Economics (1993). doi:10.1007/BF01199986.

M. Rothschild, A two-armed bandit theory of market pricing, Journal of Economic Theory (1974). doi:10.1016/0022-0531(74)90066-0.

J. Schmidhuber, Deep Learning in neural networks: An overview, 2015. doi:10.1016/j.neunet.2014.09.003.

S. Van der Hoog, Deep Learning in Agent-Based Models: A Prospectus, 2016. doi:10.2139/ssrn.2711216.

S. H. Chen, C. L. Chang, Y. R. Du, Agent-based economic models and econometrics, 2012. doi:10.1017/S0269888912000136. arXiv:arXiv:1312.0049v1.

D. McFadden, A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration, Econometrica (1989). doi:10.2307/1913621.

C. GräBner, Agent-based computational models-a formal heuristic for institutionalist pattern modelling?, Journal of Institutional Economics (2016). doi:10.1017/S1744137415000193.

M. Richiardi, The promises and perils of agent-based computational economics, LABORatorio R. Revelli, Centre for Employment ... (2003).

R. Leombruni, M. Richiardi, Why are economists sceptical about agent-based simulations?, in: Physica A: Statistical Mechanics and its Applications, 2005. doi:10.1016/j.physa.2005.02.072.

G. Hardin, The Tragedy of the Commons Author: Garrett Hardin Published by : American Association for the Advancement of Science Stable URL : http://www.jstor.org/stable/1724745, Science (1968).

C. Fosco, F. Mengel, Cooperation through imitation and exclusion in networks, Journal of Economic Dynamics and Control (2011). doi:10.1016/j.jedc.2010.12.002.

M. Ifti, T. Killingback, M. Doebeli, Effects of neighbourhood size and connectivity on the spatial Continuous Prisoner's Dilemma, Journal of Theoretical Biology (2004). doi:10.1016/j.jtbi.2004.06.003. arXiv:0405018.

J. H. Miller, The coevolution of automata in the repeated prisoner's dilemma, Journal of Economic Behavior and Organization (1996). doi:10.1016/0167-2681(95)00052-6.

J. W. Friedman, A Non-cooperative Equilibrium for Supergames, The Review of Economic Studies (1971). doi:10.2307/2296617.

D. Abreu, P. K. Dutta, L. Smith, The Folk Theorem for Repeated Games: A Neu Condition, Econometrica (1994). doi:10.2307/2951739.

H. V. Hasselt, A. Guez, D. Silver, Double DQN.pdf, in: AAAI, 2016. arXiv:1606.04615.

Appendix A.

Appendix A.1. DQQN Deep Model Architecture

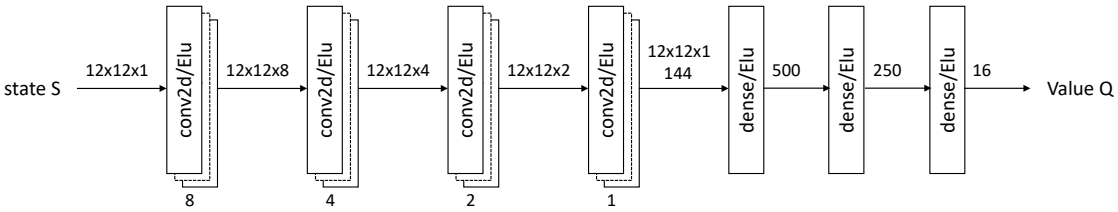

Figure A.1: Q network

Appendix A.2. Results, Standard Error (SE) of Mean Cooperation, Value Action

Standard error of the mean is calculated using: $SEM = \frac{\sigma}{\sqrt{N}}$ where σ = standard deviation of the sampling distribution, N = number of samples

Iteration	SE,Cooperation	SE,Value Action	Iteration	SE,Cooperation	SE,Value Action
0	0.00034	0.0137	40000000	0.000649	0.788
800000	0.000396	0.741	40800000	0.000618	0.784
1600000	0.000693	0.76	41600000	0.000626	0.784
2400000	0.000863	0.758	42400000	0.000619	0.785
3200000	0.000724	0.772	43200000	0.000672	0.781
4000000	0.00069	0.776	44000000	0.000672	0.778
4800000	0.000697	0.772	44800000	0.000489	0.783
5600000	0.000598	0.781	45600000	0.00061	0.781
6400000	0.000647	0.777	46400000	0.000609	0.782
7200000	0.000583	0.782	47200000	0.000583	0.781

Iteration	SE,Cooperation	SE,Value Action	Iteration	SE,Cooperation	SE,Value Action
8000000	0.0006	0.777	48000000	0.000687	0.785
8800000	0.000627	0.782	48800000	0.000627	0.784
9600000	0.000652	0.777	49600000	0.000553	0.776
10400000	0.00063	0.781	50400000	0.000634	0.781
11200000	0.000626	0.777	51200000	0.000677	0.78
12000000	0.000563	0.781	52000000	0.000611	0.787
12800000	0.000618	0.777	52800000	0.000529	0.782
13600000	0.000601	0.776	53600000	0.000548	0.785
14400000	0.000624	0.779	54400000	0.00059	0.779
15200000	0.000533	0.78	55200000	0.000655	0.783
16000000	0.000595	0.784	56000000	0.000634	0.781
16800000	0.000565	0.784	56800000	0.000574	0.786
17600000	0.00058	0.784	57600000	0.000615	0.785
18400000	0.000646	0.782	58400000	0.0006	0.78
19200000	0.000605	0.785	59200000	0.000623	0.785
20000000	0.000643	0.783	60000000	0.000529	0.787
20800000	0.000609	0.784	60800000	0.000575	0.782
21600000	0.00062	0.776	61600000	0.000527	0.788
22400000	0.00063	0.78	62400000	0.00061	0.78
23200000	0.000596	0.78	63200000	0.000547	0.781
24000000	0.000708	0.777	64000000	0.00065	0.783
24800000	0.000593	0.782	64800000	0.000599	0.781
25600000	0.000577	0.781	65600000	0.000632	0.786
26400000	0.000604	0.78	66400000	0.000572	0.785
27200000	0.00057	0.779	67200000	0.000595	0.783
28000000	0.000611	0.78	68000000	0.000691	0.785
28800000	0.000703	0.782	68800000	0.000582	0.785
29600000	0.000574	0.781	69600000	0.00065	0.779
30400000	0.000536	0.783	70400000	0.000622	0.782
31200000	0.000641	0.779	71200000	0.000654	0.786
32000000	0.0006	0.777	72000000	0.000563	0.779

Iteration	SE,Cooperation	SE,Value Action	Iteration	SE,Cooperation	SE,Value Action
32800000	0.000543	0.786	72800000	0.000597	0.787
33600000	0.000604	0.784	73600000	0.000585	0.786
34400000	0.000582	0.782	74400000	0.000676	0.779
35200000	0.000658	0.784	75200000	0.000703	0.786
36000000	0.000619	0.783	76000000	0.000574	0.782
36800000	0.000596	0.784	76800000	0.000625	0.781
37600000	0.000638	0.782	77600000	0.000532	0.783
38400000	0.000611	0.784	78400000	0.000603	0.776
39200000	0.000576	0.784	79200000	0.000598	0.78