Uncertainty, Possible Worlds and Coupled Dynamical Recognizers

Takashi Ikegami *
Sony Computer Science Laboratory,
6, rue Amyot, 75005 Paris
The Graduate School of Arts and Sciences,
University of Tokyo, 3-8-1 Komaba, Meguro-ku,Tokyo 153
and
Makoto Taiji †
Institute of Statistical Mathematics,
4-6-7 Minami-Azabu, Minato-ku, Tokyo 106

Abstract

We propose a new way of studying social dilemmas and the epistemic structures of agents. Here we study the iterated prisoner's dilemma game as played by cognitive players, where each player optimizes his or her own future actions by making an internal model of the opponent's behavior. A kind of recurrent neural network called a dynamical recognizer(DR) is used to make these internal models. The internal model of each player's behavior is constructed from a finite history. That is, many internal models are equally accurate in mimicking the opponent's behavior. If the optimized future action varies depending on which of the models is chosen, we construct branches in the world line to represent several possible future worlds. Depending on the game situation (e.g. the payoff structures, the length of past sequences to be considered, the uncertainty level in choosing models, etc.), the structures of the branching of world lines (i.e., of possible worlds) will vary. In some situations, the world line is surrounded by many possible worlds, each with different behaviors. In particular, we focus on how players can attain mutual cooperation in some of the world lines.

1 Introduction

We sometimes recognize that unavoidable uncertainty is pervasive in our lives, and we nevertheless communicate with each other and understand intended meanings. How is this possible? There is a conversation between Chuangtse and Hueitse that addresses this problem.

“Chuangtse and Hueitse had strolled on to the bridge over the Hao, when the former observed, ”See how the small fish are darting about! That is the happiness of the fish.”

”You not being a fish yourself,” said Hueitse, ”how can you know the happiness of the fish?””And you not being I,” retorted Chuangtse, ”how can you know that I do not know?”

”If I, not being you, cannot know what you know,” urged Hueitse, ”it follows that you, not being a fish, cannot know the happiness of the fish.”

”Let us go back to your original question,” said Chuangtse. ”You asked me how I knew the happiness of the fish. Your very question shows that you knew that I knew. I knew it (from my own feelings) on this bridge. ” (Chuangtse, B.C.4th C.).

We would like to modify the last part of Chuangtse's phrase as follows: “I know the happiness of the fish because I am not the fish, as you are not me.” We here study uncertainty in game-playing situations, but we are not going to eliminate uncertainties. Nor we are going to discuss strategies which counteract uncertainties. Our main message in this paper is, rather, that because of uncertainty we have notions of autonomy and insights into life itself.

Under conditions of uncertainty, we cannot directly communicate with each other. Any communication has to be made via interfaces. We consider such an interface to be comparable to a language system or our physical constraints(e.g. forms, the number of hands, etc.). These interfaces are autonomous in the sense that they have their own dynamics. Differences that are generated between a real object and its *model*

*ikeg@sacral.c.u-tokyo.ac.jp
†taiji@yamane.ism.ac.jp

are attributed to the nature of the interfaces through which the model is communicated. If such differences remain small, we say that the situation is stable, but if they increase, we say that it is unstable. When such instability increases, we may call the object autonomous, in the sense that we cannot generate a model which can perfectly predict its behavior. Agents achieve true autonomy when they obtain subjective judgement. Thus, it is important to study them in a situation in which the objective and subjective worlds differ. In other words, the difference between the real world and its internal models in agents is essential for autonomy.

In the present paper, we discuss the role of uncertainty in playing games. The uncertainty comes into play when we make models of opponents' behavior on the basis of a finite set of data.

2 Models of Players and Possible Worlds Simulation

2.1 Prisoner's dilemma game

The playing of a simple game involves one of the most straightforward interfaces possible. Here, we study the iterated prisoner's dilemma(IPD) game as an example. This game has been extensively studied in the past two decades(see e.g. Axelrod, 1984). In the 2 person IPD game, each player has to play to "defect" or to "cooperate" repeatedly. In the present paper, we represent COOPERATION by 0 and DEFECTION by 1. The payoff matrix to be studied is given as follows.

Player 1's score	Player 2's score	Player 1's move	Player 2's move
1	1	0	0
0	p	0	1
p	0	1	0
q	q	1	1

To preserve the condition of the prisoner's dilemma, p and q should hold such that $1 < p < 2$ and $0 < q < 1$. In Axelrod's original game, these parameters were $p = 5/3$ and $q = 1/3$. We know that effective strategies for playing the game that are selected through evolutionary dynamics change drastically depending on these parameters (Lindgren and Nordahl 1994, Matsushima and Ikegami 1997). In our simulations below, we choose values of p and q so as to perturb the game playing situations.

2.2 Game player as Dynamical Recognizer

A dynamical recognizer was first discussed by Pollack (Pollack 1991, Kolen 1994), and was independently used for studying dynamical aspects of language by Elman (1991). It is a kind of two-layed recurrent neural network, in which the recurrent outputs are fed back to the weights of the function networks rather than to the input layer. Therefore, network connections can temporarily be updated while the network is receiving input signals.

For the purposes of the present study, the recurrent outputs record the opponent's current status, and the context network converts these outputs into weights in the function network in order to predict the next action. Here and below we will refer to the space constructed by the outputs from the function network (including both recurrent and network outputs) as the context space. The output is taken from a node of the function network. In this study only one input and one output node are necessary since the IPD game has only two actions, cooperation and defection. The output is rounded off to 0 (cooperation) and 1 (defection). The network is expressed by the following equations.

$$z_i(n) = g(\sum_{j=0}^{M} w_{ij} y_i(n)), \quad (1)$$

$$w_{ij} = \sum_{j=1}^{N} u_{ijk} z_k(n-1). \quad (2)$$

Here g(x) is given a sigmoid function $(e^{-\beta x} + 1)^{-1}$, $y_i(n)(i = 0 \cdots M)$ denote input and $z_i(n)((i = 1 \cdots N)$ denote output states. We name the net weights as function network (w_{ij}) and context network (u_{ijk}).

In the equations above, nonlinearity exists only in the sigmoid function. We can control the degree of instability by changing the parameter β (through this paper, we set $\beta = 6.0$). In practice, we use two input neurons and three output neurons. One of the input neurons is fixed in its state as a biasing network (through this paper, we set $y_1 = 0.3$), while two output neurons are called recurrent outputs and are used recurrently to determine the function network.

To train a dynamical recognizer to mimic an opponent's behavior, a simple back propagation is usually used (Pollack 1994, Taiji and Ikegami 1998); that is, the connections are changed in proportion to the amplitude of the distance between the ideal and current output values. In practice, the error $E(n)$ after the n-th game is computed by

$$E(n) = \sum_{k=1}^{n} \lambda^{n-k}(z_0(k) - d(k))^2, \quad (3)$$

where $d(k)$ is the target, i.e. the actual opponent's action in the k-th game, $z_0(k)$ is the predicted action by the network, and λ is a parameter which controls forgetfulness. For most simulations, $\lambda = 0.95$ was used. The derivative of this error was propagated in the context network.

In the present study, instead of using back propagation, we quantify the strength of the connections u_{ijk} as either 0 or 1. Therefore we can exhaustively search all structures to arrive at those that best mimic the opponent's behavior. Since the context networks no longer have continuous values, the supposed language class is severely limited. However, we find that not only finite automata but also many other non-finite automata are mimicked by those networks.

2.3 Both Players generate Internal Models of each Other

A simulation cycle goes as follows. First, a pair of initial strings is given, a set of initial moves of 0 and 1. They are either given randomly or in specific patterns. Each player then generates their opponent's model on the basis of those initial patterns. Using the model, each player predicts his opponent's expected future moves by giving all possible combinations of inputs in strings up to the length of 10. Then the players choose the input string that is expected to bring the highest value and play the first element of that string. Then, having one new bit of information concerning their opponents' strategy, players can update their models. The next simulation cycle is then performed in the same way, and successive iterations of the cycle follow. Games between autonomous optimizers with such simulation capabilities are first discussed by O.Rössler in 1994.

By introducing uncertainty, represented by ϵ, a number of internal models are degenerated. It means that there exist an equivalent class of models which can mimic the opponent with the accuracy ϵ, but the models have different net structures from each other. Any model of the index k which satisfies the error condition,

$$E_{optimal} < E_k < E_{optimal} + \epsilon \quad (4)$$

can form an equivalent class and therefore each model can be a candidate for the internal model.

This error condition may not cause any problems if all models that have degenerated propose the same decision as their next move. If they happen to propose different moves, however, the game dynamics will depend on which model is chosen. Any criterion of model selection can be introduced here to let players continue the game, but rather than proceed in that way, we branch a world line instead. A world line is defined as a sequence of cooperations and defections. When there are more than 2 equivalent internal models and thier proposed decisions are different, we bifurcate the world line. The players in the different world line have different choice of moves. We thus have two parallel world lines, i.e., two parallel possible worlds. Sometimes the internal models of both players degenerate as well. In such cases, we have 4 possible worlds, including the *real* orbit. No two world lines can interact with each other.

3 Simulation of Possible Worlds in Model Space

3.1 World Branchings

When there is no uncertainty in the optimization process, mutually defecting attractors are always present at the end, except when some special initial configurations are used. We obtained the same results in our previous simulations with continuously valued context networks (Taiji and Ikegami, 1998). As higher uncertainty levels are allowed, more internal models can be degenerated. There exists a threshold value in the uncertainty level, above which the world lines begin to branch.

Fig. 1 shows an example of a branching pattern, where the payoff matrix is given that is close to Axelrod's original matrix. We set the uncertainty level to $\epsilon = 0.01$. All branchings that occurred before time step 80 are depicted in this figure. As shown in the figure, most of these branchings are attracted to mutually defecting (MD) states, although they can have different internal models, as will be shown below. Only two of the world lines are attracted to non-MD states. Non-MD states include mutually cooperating states or transient states that last until at least 200 time steps have passed.

How are the dynamics of the two world lines which are attracted to non-MD states different from those of other world lines? Suppose we simulate a game

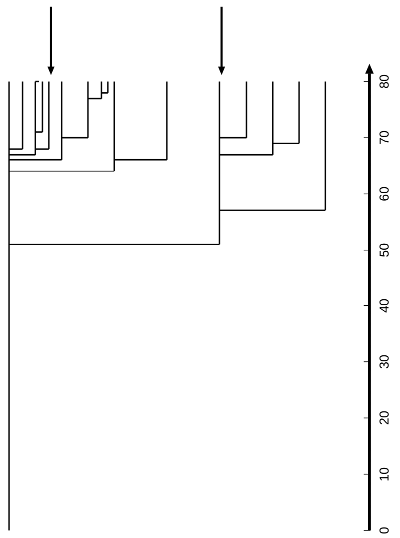

Figure 1: An example of the branchings of world lines. This figure can be read as an evolutionary tree with a single root. In this example, two branches, which are indicated by arrows, are attracted to non-MD states. The payoff matrix with $p = 1.7$ and $q = 0.4$ is used.

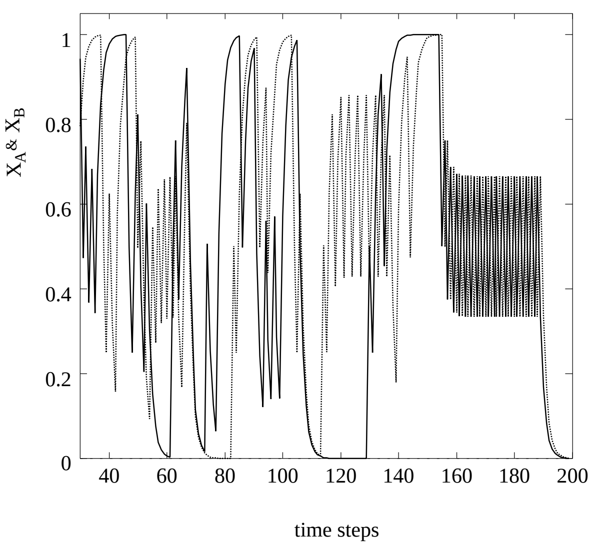

Figure 2: Synchronization of two variables. When they synchronize in the period of 2 with the inverse phase, mutual cooperation is attained. Otherwise the variables stay in transient states or go to MD states. Rigid line represents x^A and dotted line x^B.

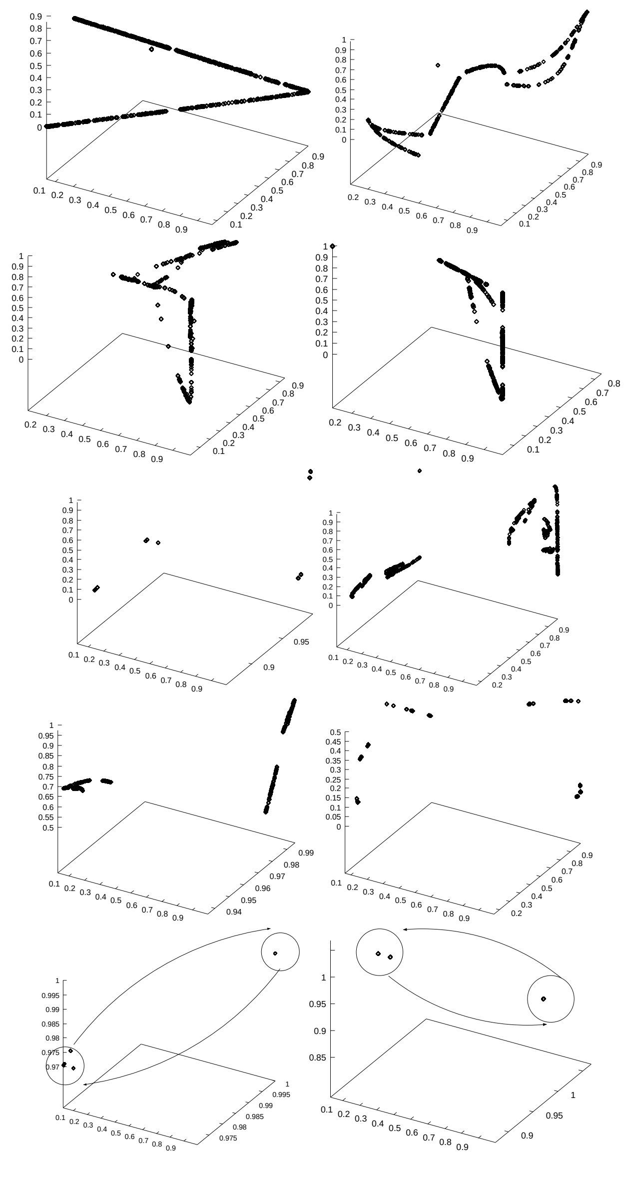

Figure 3: Context space plots of DR against all input bit patterns up to the length 10 are depicted for both players A and B. A pair of internal models are sequentially listed along the time steps. The model images in the right column represent the internal model of player B by player A. Those in the left column represent the model of player A held by B. Both sets of images are tending toward Tit fot Tat images, which are represented by two nodes and two arcs in the last picture.

between player A and B. By converting their sequences of moves ($\vec{x^A}$ and $\vec{x^B}$) into decimal numbers, we can obtain the time evolution of the two state variables x^A and x^B. This can be achieved simply by putting decimal points at the heads of the strings.

Two variables oscillating synchronously in the period of 2 but with the inverse phase, will result in players cooperating with each other(see Fig.2). When the variables fail to synchronize, however, mutual cooperation is difficult to obtain. This finding is confirmed in many other cases with different p and q values. However, the inverse statement does not always hold; i.e., players can be cooperating each other without showing any synchronizations.

To correspond with Fig.2, we depict the time evolution of the internal models of the players in Fig.3. It is interesting to note that these internal models gradually conform to Tit for Tat images. Because this period 2 oscillation is one characteristic of the Tit for Tat strategy, the inverse phase of period 2 synchronization will induce players to have Tit for Tat images of each other. Once the players arrive at Tit for Tat images, the optimal strategy is to cooperate, but if they fail to arrive at Tit for Tat images, they begin to defect again. When players fail to have Tit for Tat images, complicated images, which cannot simply be made correspondence to finite automata, are obtained.

Fig.4 depicts the overall images that players ultimately have. We indexed the internal models by converting their context network connections to decimal numbers. Tit for Tat, for example, has the index 768. The two players do not always come to have the same internal models. While internal models in MD states have some common values among their connections , those in mutually cooperative states seem to be more constrained.

3.2 Phase Diagram of World Branchings

Here, we analyze the global structure of the branchings by changing the payoff matrix with the prisoner's dilemma game conditions. We will determine: 1)how many possible worlds can be generated, and 2) how many possible worlds are attracted to non-MD states.

World lines can branch when internal models degenerate and offer various possible futures at a given time. So when a world line branches, it means that players on that world line cannot decide which option to play. From the viewpoint of game, the decisions of the players are dependent on payoff matrix (p, q). Because players compute their future expectations by knowing the payoff matrix, they are likely to deviate from mutual cooperation in high p and high q regions.

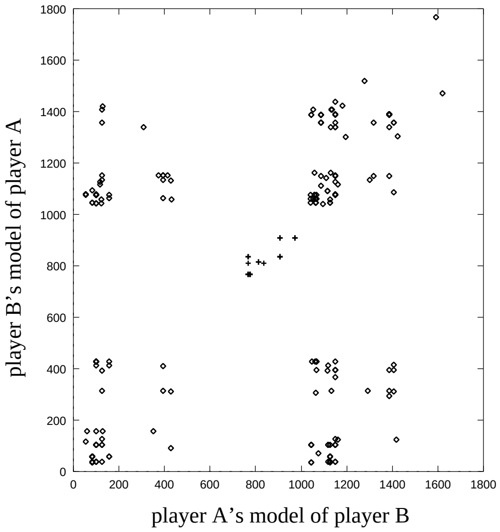

Figure 4: In the model space, MD states are represented by diamond marks, whereas mutually cooperative states are represented by crosses. The payoff matrix with $p = 1.7$ and $q = 0.5$ is used to draw this picture.

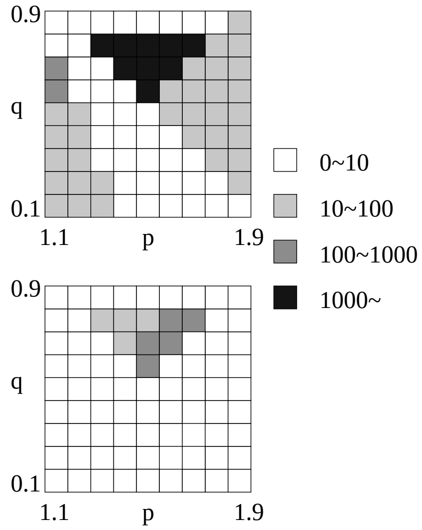

Figure 5: The number of branchings (a) and those attracted to non-MD states (b). Darker regions imply that there are more branchings (a) or more world lines tending toward non-MD states(b). Giving the initial random string of the length 30, we count the number of branchings until time setp 80. The accuracy $\epsilon = 0.01$ is used to draw the diagram.

If players can exploit others in the future without taking a big risk (i.e. high p and high q), they are likely to defect from others rather than cooperate. When they have Tit for Tat images of their opponents, however, instant defection is not a wise decision even in those high p, high q regions. This result is insured by the prisoner's dilemma condition. We thus can comparatively discuss the differences in branching patterns only for the same input strings (e.g. initial strings).

In Fig. 5, the number of branchings and the number of branches which go to non-MD states are computed as functions of the parameters p and q. From this phase diagram and the analysis of it we see the following: 1) Some (p, q) regions have exponentially increasing branchings, 2) Having many branchings does not imply the existence of many world lines that go to non-MD states, and, 3) Only some (p, q) regions have a high number of branchings with many world lines tending toward non-MD states. It is of interest that the frequently branching regions are separated from the others by straight lines in this diagram. This is reminiscent of the one-memory strategy diagram in the IPD game (Lindgren and Nordal, 1994). It is possible that some initial strings (given memory sequences) can induce only internal models with shorter memory capacities (for example small size finite automaton), while others induce models with longer memory capacities.

4 Discussions and Future Remarks

We have seen that states of mutual cooperation are reachable when players are allowed to use slightly less than optimal models. By computing the possible world structures around the real world line, we saw that many world lines that lead to cooperating states are embedded in it. This result is also a function of the payoff matrix.

We insist that our study is not merely intended to have philosophical implications. We believe we can contribute to the field of robotics, in which decision making and optimization, as addressed in our study, are major trademarks, both at the hardware and software levels(Tani, 1996). We believe that uncertainty in decision making should also be a central issue in robotics. We thus can propose one possible test for measuring the autonomy of a robot without looking inside the programs.

First, we define autonomy in robots as when they behave differently under the same experiment conditions. In our study, the coexistence of many branches with different attractors in the possible worlds is taken as a sign of autonomy. By preparing an environment which can be translated into the prisoner's dilemma game, we are able to look into the possible worlds of the behavior patterns of robots. Possible worlds can be realized by simply doing the same experiment many times. Then, based on the branching patterns that have appeared, we can measure the autonomy of the robot. This possible world analysis can be taken, so to speak, as a new kind of Turing test.

(This manuscript is a shorten version of our paper submitted to the proceedings of the 3rd international conference on Emergence (Helsinki, August/1998)).)

References

[1] R. Axelrod. *The Evolution of Cooperation*, (Basic Books, New York) 1984.

[2] Chuangtse, *Autumn Water* B.C. 4C. (translated by Zhuang Zi).

[3] J.Elman, Language as a Dynamical System in *Mind as Motion* (eds. R.E.Port a nd T.van Gelder, MITpress 1995) pp.195–226.

[4] G.E.Hughes and M.J.Cresswell, *An Introduction to Modal Logic* (Victoria univ. press, 1968)

[5] J.F.Kolen. Exploring the Computational Capabilities of Recurrent Neural Networks. Doctoral Dissertation, The Ohio State University, 1994.

[6] K.Lindgren and M.G.Nordahl. Evolutionary dynamics of spatial games. *Physica D* 75:292–309, 1994.

[7] M. Matsushima and T. Ikegami, Evolution of Strategies in the 3-person IPD game. (submitted to *J. Theor. Biol.* 1997)

[8] J. B. Pollack. The induction of dynamical recognizers. *Machine Learning*, 7:227–252, 1991.

[9] O.E.Rossler, Fraiberg-Lenneberg Speech, *Chaos, Soliton and Fractals* 4 (1994) pp.1 25–131.

[10] M. Taiji and T.Ikegami, Dynamics of Internal Models in Game Players. (submitted to *Physica D* 1998).

[11] J.Tani, Model-Based Learning for Mobile Robot Navigation from a Dynamical Systems Perspective. *IEEE Trans. System, Man and Ctbernetics Part B, Special issue on robot learning*, Vol. 26 No.3, 1996.