A Cognitively Founded Model Of the Emergence Of Social Conventions

Umberto Gostoli

Department of Economics
Università Politecnica delle Marche
Piazzale R. Martelli, 8
60121 Ancona - Italy
ugostoli@hotmail.com

ABSTRACT. This paper presents a model of learning in the context of a relatively large population interacting through random bilateral matching to play a bilateral game in strategic form. While the theory of learning in games commonly assumes that the players can observe only the strategies chosen by their opponents, this paper introduces the additional assumption that the players are characterized by phenotypic traits observable by the other players with whom they interact. The extension of the traditional framework allows to introduce a more sophisticated and cognitively plausible formation model expectations than the ones proposed so far. In particular, this paper proposes a new model of the induction process through which the agents build mental models that take the form of lexicographically structured decision trees.

KEY WORDS: Theory of learning; Lexicographic categorization; Social stereotyping; Fast and frugal heuristic theory.
1. Introduction

Any dynamic or repeated game is characterized by a particular information structure that defines what the players know before the game starts and what they can observe during the stages, or periods, of the game. Moreover, any game is, implicitly or explicitly, characterized by a certain information processing algorithm that defines and represents the players’ cognitive skills. The equilibria reached, as well as the dynamics that leads to them, depend on the information structure of the game and on the decision process through which the players compute the information at their disposal.

It is commonly assumed that the players know the payoff matrix of the game, can observe the action of their opponent in each period and, moreover, that all the players know that the other players have the same prior knowledge and observational skills, that all the players know that the other players know and so on. These are the so-called perfect and common knowledge assumptions that come from the traditional static game theory, whose formal application to the structure of the game can lead to technical and philosophical problems (Binmore, 1997).

While the information available to the players has been considered by means of formal assumptions, less attention has been paid to the explicit formal definition of the cognitive process through which this information is computed by the players in order to reach a decision of what action to undertake. However, as the behavioural game theory shows, the differences between the outcomes of experiments and the predictions of game theory are often due to the unrealistic computational power and rationality assumptions of the latter (Camerer, 2003).

The theory of learning in games is the branch of the literature that introduced models of players less than perfectly rational and that explicitly formalized, together with the information structure of the game, the decision process that underlies the player’s action. While the game theory tells us which Nash equilibria a particular game has, the theory of learning in games gives us models that determine the path through which a certain equilibrium is reached. In these models, the equilibrium is the outcome of a process in which less than fully rational players grope for optimality over time (Fudenberg and Levine, 1999).

The approaches to learning in games can be divided in two large classes: reinforcing and forecasting learning models. The first class includes models where the players choose a strategy on the basis of its past performance. The imitation models belong to this class: the players are endowed with the unsophisticated but not trivial cognitive skills to assess their neighbours’ success and to associate this performance to a particular strategy. The second class is represented by models where the players are endowed with the cognitive skills to develop a forecasting model of the behaviour of other players and, so, to choose the strategy that is the best reply to their opponent’s expected action.

The fictitious play models belong to this last class. In general, in these models players are endowed with the basic skill to observe and memorize their opponent’s action. This cognitive skill allows the agents to keep track of the relative frequencies with which each strategy is played. These relative frequencies represent the players’ expectation about the strategies of distribution of the population.
In the model proposed in this paper the information structure of the game is extended to include the players’ skill to observe their opponent’s phenotype, represented by a string of three binary attributes. Given the information processing skills that characterize the fictitious play models, this extension would allow the players to develop expectations conditioned by their opponents’ phenotypes.

Of course, this extension is not new in the field of the theory of learning in games. Previous papers have introduced models where the agents are endowed with a visible tag composed by one binary attribute (Axtell, Epstein and Peyton Young, 1999) or by three binary attributes (Hoffmann, 2006). These works demonstrated that the agents’ capacity to develop expectations conditional on their opponents’ type leads to the formation of social classes. In other words, in these models we can observe the endogenous emergence of social stereotyping. The main difference between the present work and the two papers mentioned above is represented by the cognitive process through which the agents process the information at their disposal.

In fact, in this paper, the extension of the information structure of the game is not important in itself but is functional to the introduction of a new model of information processing algorithm, a model inspired by a recently introduced decision process theory: the Fast and Frugal Heuristic Theory (FFHT).

The FFHT proposes a lexicographic decision process, that is, a process through which the attributes are looked up in a particular order of validity by the means of a decision tree. Figure 1 shows a lexicographic decision tree with which is possible to classify instances, identified by three binary attributes, in the two classes A and B.

---

**Figure 1.** A lexicographic decision tree that classifies instances characterized by three binary cues. The numbers in the ovals identify the cue while the letters in the rectangles represent the two classes to which each instance can belong.

Faced with the situation 0-1-1, the agent would classify it as an instance of class A: the third attribute (identified with the number 2) has value 1, so he has to consider the second node, containing the first attribute (identified with the number
0). His decision process does not need to proceed further because, as this attribute’s value is equal to 0, the decision tree ends with a leaf, containing in this case the class A. The model presented in this paper is based on the assumption that the agent’s mental model is structured as a decision tree.

The FFHT is based on the bounded rationality paradigm: because of the mind’s limitations, real decision makers have to use approximate methods to handle most tasks (Simon, 1955). The function of these methods, that we will call heuristics, is to make reasonable, adaptive inferences about the social and physical world given limited time and knowledge. In these situations, indeed, real agents cannot perform the cognitively demanding multiple linear regression, but, more plausibly, they look up only few available cues in a fixed order of validity (Gigerenzer and Goldstain, 1996).

The cues relevant in the decision-making process and the order in which they have to be considered can be learned in three main ways: they could be genetically coded, they can be learned through cultural transmission or they can be learned from the agent’s experience. In the model presented in this paper, the number and the kind of cues the agents can detect are genetically determined but the decision tree by which they are hierarchically ordered to form their mental model is the result of an induction process that the agents perform on the database of their experiences.

According to FFHT, in an evolutionary context, decision-making strategies are selected for their accuracy, frugality and speed, measures that relate the decision process to the environment in which the decision has to be taken. In fact, the FFHT introduces a new concept of rationality called ecological rationality: a heuristics, or mental model, is rational from an ecological point of view if it allows the agents to make effective decisions in the environment in which the agents live. This means that in order for a given heuristics to be rational it has to match the information structure of the environment in which that particular heuristics is used.

In a social context, however, the environment of each agent is represented by other agents. So, the mental model of each agent is rational from an ecological point of view if it matches the mental model of the other agents with whom it interacts. It is evident that, in a social system, each agent represent part of the environment for all the others and, so, in this context, the concept of rationality is necessarily socially determined.

The main aim of this paper is to find out if a social system formed by agents that make heuristic inferences about their opponent’s behaviour can reach an equilibrium, that is, a situation where the mental model that each agent develops through his experiences allows him to make ecologically rational decisions given the mental models developed by the other agents with whom he interacts. The model presented in this paper is a model of co-evolution of mental models with which we try to find out if the system ever reaches the steady state in which each agent’s mental model is confirmed by, and because of, the other agents’ mental models.

It is important to make clear that the term co-evolution in this model has nothing to do with the mutation and natural selection processes that characterize the evolutionary game theory models: here the evolution of mental models is entirely a cognitive process that takes place within each agent. In other words, in this model,
the agent’s behaviour changes not because of a casual mutation of his DNA but because the experiences he undergoes change his mental model.

2. The Model

As this is an agent-based model, it has to be described at two levels: the population and the agent level. The former specifies the kind of interaction that takes place among the agents who compose the population, while the latter specifies the decision process that take place within each single agent.

2.1. The structure of agents’ interactions

The model presented in this paper considers a population of $N$ agents randomly paired in each period to play the Stag Hunt Game, whose payoff matrix is shown in Figure 2. This game has two Nash equilibria: the socially optimal equilibrium S – S and the risk-dominant equilibrium H – H.

While in the evolutionary game theory the agent has no choice but to play the strategy to which he is genetically or culturally associated, in this model the agent has to make a decision in each period about which strategy to play. Of course, this decision is based on the expectation the agent has about his opponent’s action: he will play S if he expects his opponent to play S, otherwise he will play H.

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2,2</td>
<td>1,0</td>
</tr>
<tr>
<td>H</td>
<td>0,1</td>
<td>1,1</td>
</tr>
</tbody>
</table>

Figure 2. The Stag Hunt Game

2.2. The agent’s model

To understand how expectations form and evolve, we have to describe the model at the level of the agent. Regarding the agents’ prior knowledge, this model adopts the limited knowledge paradigm that characterizes the models adopted by the theory of learning in games: the agent knows only his own payoff matrix and does not know what his opponent gets from the interaction.

Each agent is characterized by phenotypic traits represented by three binary attributes. This means that the population is composed by eight different phenotypes: 000, 001, 010, 011, 100, 101, 110, 111. Moreover, each agent can observe the phenotypic traits of his opponent, together with the action his opponent performs. So, after each game, the phenotypic traits and the action of the opponent is stored in the agent’s memory $M$, its size being a parameter of the model. An
example of an agent’s experiences database could be the matrix shown in Figure 3. This matrix shows that in his last game, the agent met an opponent whose phenotype was 001 and who played strategy S. Of course his own phenotype and action have been stored at the same time in his opponent’s database of experiences.

We call the agents’ memory size $s$, that is, the number of experiences the agent can store in his memory. When the number of periods $p = s$, the agent’s memory is full. New experiences are stored according to the FIFO system: the oldest experience in the database is discarded to make room for the latest experience.

Having defined the agent’s gathering and storing cognitive skills, we come to the information processing algorithm that underlies the agent’s decision. We call $l$ the number of periods that constitutes the agent’s learning period, its length being another parameter of the model. For the first $l$ periods the agent has not yet developed a mental model that allows him to forecast his opponent’s move so he makes a random forecast and chooses the action that is the best response to this forecast. We have to notice that, in general, to best respond to a random expectation is not the same as making a random choice: in the first case the agent never chooses strongly dominated strategies, an event that, instead, is possible in the second case.

<table>
<thead>
<tr>
<th></th>
<th>Attribute 1</th>
<th>Attribute 2</th>
<th>Attribute 3</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>game 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>S</td>
</tr>
<tr>
<td>game 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>H</td>
</tr>
<tr>
<td>game 3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>H</td>
</tr>
<tr>
<td>game 4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>S</td>
</tr>
<tr>
<td>game 5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>H</td>
</tr>
</tbody>
</table>

Table 1. An agent’s experiences database after 5 games

At the end of the learning period the agent builds, through an induction process performed in his experience database, a mental model that allows him to forecast, from his opponent’s phenotype, the strategy his opponent will play. An example of an agent’s mental model can be the decision tree of Figure 1, interpreting the letters A and B in the rectangles as, respectively, the two strategies S and H. If an agent with this mental model meets an opponent whose phenotype is, for example, 011, the agent’s forecast of his opponent’s move would be S. The mental model represents, in fact, a set of hierarchically organized behavioural rules of the kind If/Then that maps each phenotype to a strategy to be chosen when meeting an agent with that phenotype.

From the cognitive point of view, we have to distinguish two different processes: the induction process through which the agent develops his mental model on the basis of his experiences, and the decision process, that takes place when the agent forecasts, with his mental model, his opponent’s move. In other words, with the induction process the agent makes the mental model, with the decision process he puts it into use. While the decision process takes place in each match, the induction
process can also be performed once every given number of periods \( r \), this number being another parameter of the model.

2.2. The induction process

While the decision process is quite straightforward, it is necessary to specify in a detailed way the induction process through which the agent develops his mental model from the database of his experiences. In this model, this cognitive process is modelled by the means of data mining techniques.

Data mining differs from other forecasting techniques, like the multiple linear regression, by two characteristics that make this technique particularly suitable for the construction of a model of the decision making process in the context of this model. First of all, the variables that compose the database on which the data mining techniques are applied are discrete. Even when the variables are continuous, they have to be transformed in discrete variables through the introduction of appropriate ranges. In the context of this model, where the social interaction takes the form of a game, this is an advantage because in a game, typically, the agent’s strategy space is discrete and, moreover, it is assumed that the agents’ phenotype is composed by attributes that have discrete values.

Secondly, the regression techniques are compensatory, that is, all the variables are considered without any particular order: what we do is simply to add all the variables’ values after having multiplied them by the appropriate weights. The data mining techniques, instead, are lexicographic: they classify the different variables according to their relevance in the classification process. Indeed, while the output of a regression analysis is a linear equation, the outcome of the data mining analysis is, typically, a decision tree that orders the independent variables hierarchically from the most to the least relevant, a more cognitively plausible mental model structure according to the FFHT.

The best way to describe the information processing algorithm that represents the agent’s induction process is to look closely at an example. Let’s consider the database of Table 2, containing 20 instances. The database is formed by three independent variables that define each instance (\( V_1, V_2, V_3 \)) and one dependent variable representing the class to which the instance belongs (\( C \)). The first and the third independent variables are binary variables: they can have either value 0 or 1. The second independent variable can have three values: 0, 1 or 2. The instances can belong either to class 0 or to class 1.

The first step of the data mining process is to find out the first independent variable of the decision tree. In order to do so, we have to compute the average information value for each independent variable. According to the information theory, the information value of a database is 1 minus the entropy of the database. The database’s entropy is defined as the number of bits required to specify the class of an instance given that it belongs to the database. The entropy value can go from 0, if the information that an instance belongs to the database is all we need to classify
the instance, to 1, if the fact that the instance belongs to the database does not gives us any useful information about its class.

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2. A database

If we call $N$ the total number of instances in a database composed by a variable that can be classified in $r$ classes and $n_i$ the number of instances belonging to class $i$, the entropy $E$ of a database is given by [1]:

$$E = - \frac{n_0}{N} \log_2 \frac{n_0}{N} - \frac{n_1}{N} \log_2 \frac{n_1}{N} - \cdots - \frac{n_i}{N} \log_2 \frac{n_i}{N} - \cdots - \frac{n_r}{N} \log_2 \frac{n_r}{N}$$ [1]

For example, if we look at the dependent variable column (C) of the database of Table 2, we see that there are 8 instances belonging to class 0 and 12 instances belonging to class 1. The entropy of this database is given by:

$$E = - \frac{8}{20} \log_2 \frac{8}{20} - \frac{12}{20} \log_2 \frac{12}{20}$$
So, the entropy of the database is 0.971. It represents the additional number of bits we need in order to classify an instance given that it belongs to the database. Therefore, the information value of the database \( \text{info}(D) \) is:

\[
\text{info}(D) = 1 - 0.971 = 0.029
\]

If the \( C \) column was composed by the same number of 0’s and 1’s, the database entropy would be 1 and the its information value 0. This makes sense because, intuitively, the fact that an instance belongs to the database, in this case, would not give us any useful information regarding its class. Conversely, if the \( C \) column was composed, for example, entirely by 1’s, the database entropy would be 0 and its information value 1: the fact that an instance belongs to the database would give us all the information we need to classify the instance as an instance of class 1.

With these basic information theory concepts we are now able to compute the average information value of each independent variable. For each independent variable we can identify as many subsets of the whole database as the number of different values the variable can have. For example, let’s consider the variable \( V1 \). It can have value 0 or value 1, so we can build two subsets from the initial database of Table 2: the subset of the classes of the instances where \( V1 \) is 0, that we call \( C(V1=0) \), and the subset of the classes of the instances where \( V1 \) is 1, that we call \( C(V1=1) \) (Table 3 and Table 4).

<table>
<thead>
<tr>
<th>( C(V1=0) )</th>
<th>( C(V1=1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3

Table 4

Now we can calculate the entropy of each database:

\[
E[C(V1=0)] = - \frac{2}{10} \log_2 \left( \frac{2}{10} \right) - \frac{8}{10} \log_2 \left( \frac{8}{10} \right) = 0.722
\]

\[
E[C(V1=1)] = - \frac{6}{10} \log_2 \left( \frac{6}{10} \right) - \frac{4}{10} \log_2 \left( \frac{4}{10} \right) = 0.971
\]
Being the dimension of both the subsets 10, the average entropy of variable \( V_1 \), that we call \( E(V_1) \), is:

\[
    E(V_1) = \frac{10}{20} \times 0.722 + \frac{10}{20} \times 0.971 = 0.846
\]

For the information theory, it means that if we know the value the variable \( V_1 \) has in a particular instance, to classify correctly that instance we need, on average, 0.846 additional bits. The average information value of \( V_1 \), \( \text{info}(V_1) \), is, consequently:

\[
    \text{info}(V_1) = 1 - 0.846 = 0.154
\]

We now repeat the procedure for the second independent variable \( V_2 \). First, we have to build the subsets of the original database. Being \( V_2 \) a variable that can have three values, we will identify three subsets, that are represented below (Table 5, Table 6 and Table 7).

The database of Table 5 contains the classes of the instances where \( V_2 \) has value 0, the database of Table 6 contains the classes of the instances where \( V_2 \) has value 1 and the database of Table 7 contains the classes of the instances where \( V_2 \) has value 2.

<table>
<thead>
<tr>
<th>( C(V_2=0) )</th>
<th>( C(V_2=1) )</th>
<th>( C(V_2=2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5  Table 6  Table 7

We calculate the databases' entropies:

\[
    E[C(V_2=0)] = - \frac{2}{5}\log_2 \left( \frac{2}{5} \right) - \frac{3}{5} \log_2 \left( \frac{3}{5} \right) = 0.971
\]

\[
    E[C(V_2=1)] = - \frac{5}{9}\log_2 \left( \frac{5}{9} \right) - \frac{4}{9} \log_2 \left( \frac{4}{9} \right) = 0.994
\]

\[
    E[C(V_2=2)] = - \frac{1}{6}\log_2 \left( \frac{1}{6} \right) - \frac{5}{6} \log_2 \left( \frac{5}{6} \right) = 0.649
\]
Given the databases’ dimensions, the average entropy of variable V2 is:

$$E(V2) = (5/20) \times 0.971 + (9/20) \times 0.994 + (6/20) \times 0.649 = 0.885$$

The average information value of V2 is:

$$\text{info}(V2) = 1 - 0.885 = 0.115$$

Following the same procedure, we find the average information value of variable V3, that is:

$$\text{info}(V3) = 1 - 0.912 = 0.088$$

The most informative variable, that is the variable with the highest average information value, is variable V1 and so this is the first variable, or the root, of the decision tree. Being variable V1 a binary variable, the first node of the decision tree will have two branches, one for each possible value of variable V1.

The next step is to find out the variable with the highest average information value for each branch, excluding variable V1 that has been already chosen as root of the decision tree. This means building from the original database, a subset for each branch: a subset of instances where V1 has value 0 and a subset where V1 has value 1 (respectively the right branch and the left branch of the tree shown in Figure 3).
For each database we will compute the average information value of V2 and V3, choosing for each node the variable with the highest average information value. Having determined the second node for each branch, in this particular case, the process terminates, because the third variable will be the one that is left after the first two have been chosen. We have to notice that not necessarily the decision tree has to contain all the independent variables of the initial database: it could be that the data-mining analysis identifies independent variables that have no informational value.

The decision tree we get at the end of this data mining algorithm will be the decision tree that maximizes the number of instances that are correctly classified or, in other words, that minimizes the number of the exceptions. This decision tree can also be defined as the theory that best describes the database or, alternatively, the most likely theory given the database.

From a general point of view, the model described above is a feed-back process: the agents’ experiences determine the agents’ mental model which determine the agents’ behaviour which determines the agents’ experiences and so on. The main aim of the simulations, whose results will be presented in the following section, is to find out if this process ever reaches the steady state where the agents’ behaviour produces experiences that confirm the agents’ mental model.

It is important to point out that the equilibrium we are talking about here is not the usual *strategy equilibrium* considered by both the classic and evolutionary game theory: in this model the equilibrium we are mainly interested in is the *mental model equilibrium*. Of course, once we reach the mental model equilibrium we will also have a strategy equilibrium as a direct consequence of the former.

3. The Simulation

In this section are presented the results of a particular simulation that will serve us as example, results that are, however, representative of the outcome of simulations based on the social and cognitive model presented above. In fact, making the agents in their learning period random forecasts, different simulations will produce results that are different in their details, but, nevertheless, we can see that they share general statistical features.

First of all we have to set the parameters of the model. These parameters, presented in the previous section, are: the number of agents, $N$; the agents’ memory size, $M$; the length of the learning period, $l$; the number of periods after which each agent updates its mental model, $r$. In the present simulation the value of these four parameters have been set to:

- $N = 1000$
- $M = N (40, 4)$
- $l = M/4$
- $r = 1$

So, the population is formed by 1000 agents who, in each period, are randomly paired to form 500 couples to play a one-shot stag hunt game. At their birth the
agents are assigned a randomly generated phenotype and a memory, which is randomly drawn from a normal density function having average 40 and variance 4. The learning period for each agent is ¼ of her memory. This means that each agent will form a mental model after its memory has been filled for ¼ of its size. Then, the mental model will then be updated after every game (τ = 1).

In order to see if the dynamics of the simulation leads to an equilibrium we are interested in tracking the population forecasting success, that is, the number of agents who make the right forecast of their opponent’s action. From the beginning of the game till the end of the learning period this performance will be around 50%, that means that half of the population has a correct expectation about the opponent’s behaviour, because of the random guess the agents make during the learning period. Then, as the agents develop and update their mental model, we expect this performance to increase until it reaches, eventually, the equilibrium, a state where 100% of the agents have correct expectations about their opponent’s strategy.

The graph of Figure 4 shows the path of the average forecasting performance in the particular simulation we are considering: as we can see, at around period 50, the forecasting ability of the agents begins to grow at increasing speed until period 180 when about 90% of the agents guess correctly their opponent’s choice. Then, the performance growth tends to slow down, until it reaches 100% around period 350.

![Figure 4. Population forecasting success](image)

Even if the exact timing of the average forecasting performance growth changes from one simulation to the other, the common feature of all simulations is that it always reaches 100%. This means that the system always reaches the mental model equilibrium, where the expectations of each agent of the population consistently match the other agents’ expectations. After period 350, the experiences the agents make are consistent with their mental model and, consequently, the evolution of their mental models reaches a steady state. At this point, for each agent its mental...
model is the ‘true’ mental model because it allows it to forecast perfectly its opponent’s behaviour in every game.

The mental models that characterize the equilibrium are a peculiar character of each simulation because they depend on stochastic fluctuations that take place during the learning period, when each agent makes random forecasts. The eight decision trees below show the mental models of a sample of eight agents, one for each phenotype, after the system has reached the mental model equilibrium.

The agents use these mental models to forecast its opponent’s behaviour. To see how these mental models determine the agents’ behaviour, let’s consider the decision tree of the agent whose phenotype is 000. It tells us that the first thing this agent looks at is the third phenotypic trait (P3) of its opponent. If the value of this phenotypic trait is 0, then it looks to the second phenotypic trait (P2) and, if it is 0, the decision process terminates: the agent will forecast that its opponent will play H. If the value of P3 is 1, this agent only needs to look at the first phenotypic trait of its opponent (P1): if it is 0 then the forecast will be H, otherwise the forecast will be S. If the agent 000 meets the agent 001, we can see that the outcome will be H – H, so, in this case, we can say that the players’ expectations confirm each other, as we should expect at the equilibrium.
With the mental models shown above, the outcome of the 36 possible matches is shown in Table 8. First of all, we can notice that all the matches show a perfect correspondence that characterizes, in every match, the two players’ expectations. In other words, all the mental models are consistently confirmed by the outcome of the interaction: we have reached the mental model equilibrium. Secondly, 19 of the 36 matches are characterized by the S – S strategy equilibrium, while the remaining 17 matches are characterized by the H – H equilibrium. This means that, at the population level, in each period around 53% of the agents play the strategy S and around 47% play the strategy H. This represents the strategy equilibrium that determine the overall efficiency reached by the system in this particular simulation.

We can also notice that this equilibrium is characterized by a particular average payoff distribution: while the agent with the phenotype 111 plays the strategy S in 6 of the eight possible matches it can have, with an average payoff of 1.75, the agent with the phenotype 110 plays S only in 2 matches, with an average payoff of 1.22.

<table>
<thead>
<tr>
<th>Match</th>
<th>Payoff</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 – 110</td>
<td>000 – 111</td>
</tr>
<tr>
<td>001 – 001</td>
<td>001 – 010</td>
</tr>
<tr>
<td>010 – 100</td>
<td>010 – 100</td>
</tr>
</tbody>
</table>

Table 8.

4. Conclusions

While the specific results of the simulation presented in the previous section depend on variables and parameters determined through stochastic algorithms and, consequently, change from one simulation to the other, we can point out some general characteristics of the dynamics of the social and cognitive model presented in this paper.
First of all, the mental model equilibrium, and the consequent strategy equilibrium, are reached in all the simulations, after around 300 periods. This means that, given the assumptions of the model, the emergence of a social stereotyping system, that is, a set of socially formed and evolved beliefs that tend to confirm and strengthen each other, is statistically a very likely phenomenon. These beliefs are not true from an absolute point of view: at the beginning of the simulation the phenotypes attached to each agent do not have any influence on the agent’s behaviour. However, by playing repeatedly the game, the agents develop beliefs about their opponents’ behaviour that become true from a social point of view: they are true because the agent’s opponent holds beliefs that make them true. In other words, the model presented in this paper suggests a mechanism for the endogenous emergence of social conventions, defined as the equilibrium reached by a system of beliefs that evolves until it reaches a state of internal coherence.

We have to notice that, even if the particular social convention that the system develops depends on stochastic events, or historical accidents, nevertheless, once it gets established it is in the interest of each single agent to follow it. In other words, even if would be in the interest of each agent to change the convention to reach the socially optimal equilibrium, the situation where all the agents choose S, an agent that would decide not to follow the convention would be worse-off in a system where the other agents follow it. This makes the convention a steady state from which it is almost impossible to escape, unless subsets of the population decide collectively to change it.

Secondly, the strategy equilibrium that characterizes the social convention is not the social optimum but it is not the worst social outcome, represented by the risk-dominant equilibrium, neither. In fact, the simulations show that the proportion of interactions S – S goes from 40 to 60% of the total. However, we have seen that each equilibrium is characterized also by an unequal average payoff distribution among the various phenotypes. This fact would have important consequences if we include in the model a differential reproductive process of the agents’ phenotypes. In this case, the phenotypes having the higher average payoff would tend to grow in the population relatively to the phenotypes with the lower average payoff. This demographic dynamics leads to the important result that the socially optimal outcome, where all the agents play Stag, is always reached.

Finally, from a general point of view, these simulations show that, if we give up the perfect and common knowledge paradigm that characterizes the classical game theory to embrace the bounded rationality paradigm of the theory of learning in games, the dynamics and the equilibrium eventually reached by the system depend crucially on the assumptions about the cognitive skills of the agents, cognitive skills that, in order to build models that have some positive or normative value, need to be empirically justified. One of the aim of this paper has been the proposal of an agent that is one step closer to the cognitive sophistication of real agents than the agent that have populated the fictitious play models so far. The additional assumption adopted in this paper is that the agents’ behaviour is based on the mental model that takes the form of a decision tree through which the agents analyse the input in a lexicographic way. This assumption appears to be, in many cases, cognitively more plausible than the compensatory techniques that are typically used in statistical
analysis and that represent, so far, the more widely adopted model of the agents’ processing skills.

From the methodological point of view, introducing a more realistic model of the agent’s cognitive skills may require us to substitute mathematics with the programming languages as tool to formalize models: the cost of this substitution is the necessity to abandon general deductive prepositions for conclusions based on the statistical analysis of a more or less lengthy series of simulations.

The author would like to thank his PhD supervisor Prof. Roberto Scanzieri for his constant support during all the stages of the research. I am grateful to Prof. Nigel Gilbert who, during the three-month research period I spent at the University of Surrey, gave me precious advices that helped me to reconsider critically my work, allowing me to amend some of its weak points. I wish to thank Prof. Pietro Terna and his PhD students for the invaluable help they gave me regarding the technical aspects of the agent-based simulation. Finally, I am grateful to XJ Technologies for the free supply of AnyLogic 5.2, the software used to develop the model presented in this paper.

4. References


Short title of article  20


