How Soft Games Can Be Played

Philippe De Wilde
Department of Electrical and Electronic Engineering
Imperial College of Science, Technology and Medicine
London, UK
p.dewilde@ic.ac.uk

ABSTRACT: This paper shows how games can be played that involve natural language, learning, and evolutionary fictitious play. This is implemented using respectively fuzzy logic, neural networks, and evolutionary programming. Because of this combination, they can be called soft games. The algorithm is described, and placed in the context of related research.

KEYWORDS: intelligent agents, games, soft computing, learning, replicator dynamics, evolutionary dynamics

1 Intelligent agents and game theory

An agent should not operate on its own. Communication is essential in a modern, distributed environment. Moreover, collaboration can have an effect greater than the sum of the individual contributions. No single ant, however sophisticated, can build a termite nest. In software terms this means that many relatively simple agents may be more powerful than a few sophisticated agents.

This should not be taken to extremes. If agents become too simple, they will only be able to do straightforward operations on bitstreams. Such a network of switches may operate efficiently, but does not show any intelligence. This will be sub-optimal. Communications can be routed so much more efficiently, if something is know about their content, rather than just a probability distribution of their symbols.

For intelligent networks, composed of intelligent agents, it is thus best to choose moderately sophisticated agents. How should their communication be organized? Rosenschein and Zlotkin have made a powerful argument for basing the communication of intelligent agents on game theory (Rosenschein and Zlotkin, 1994). Their emphasis is on negotiation. Coordination of agents by games has more recently been advocated in (Ossowski, 1999).

Negotiation is part of game theory. Should agents be restricted to negotiations only? Agents act on behalf of humans (Mamdani and Pitt, 1998). They should be able to establish the full range of human behaviour. They should be able to play any game with a certain degree of success. The ability to play a game is a necessary requirement for any agent.

Game theory has developed a solid theoretical framework for analyzing games. The methods used facilitate mathematical analysis. This is right if the players are rational, try to maximize their utility, and are able to do analysis off-line during the game. If the game playing agents are to mirror their human owners, they need special properties.

Firstly, they need to be able to deal with language, expressing moves in natural language. A proven way to compute with words is to use fuzzy logic (Zadeh, 1994). Making fuzzy moves will be essential in a theory of games for intelligent agents.

Secondly, agents need to be able to learn, because their human owners learn. A theory of learning in games has been presented in (Fudenberg and Levine, 1998). It is quite different from human learning in that the models that are optimized are mathematical models, chosen for their tractability rather than their closeness to human learning. How humans learn is a matter for neurobiology, and, some would claim, psychology. There is a strong argument in favour of connectionist learning (Rumelhart et al., 1986), a view that is adopted in this paper. Agents that learn using neural networks will resemble human learning to some extent. They learn to classify patterns from examples.

Finally, agents need to compete in an evolutionary environment, because humans do so. This leads us into the domain of evolutionary game theory (Weibull, 1995). However, the number of agents is fixed in many realizations of agent networks. Our agents will each have a population of agents, subject to evolutionary dynamics, that helps them to make moves.

As the proposed agents contain fuzzy, neural, and evolutionary attributes, they will be called soft agents. During the game, the three attributes will interact, they are more than just modules, they are integrated. The following three sections describe the attributes of the agents, explain how they are used in a game, and discuss their relationship with current concepts from game theory.

2 Game with fuzzy information

Combinations of fuzzy systems and games have been explored before. Coalitions can be fuzzy (Aubin, 1981). The Nash equilibrium can be fuzzy, because of fuzziness in the payoffs (Ponsard, 1987). Generally, in any decision making process, preferences can be fuzzy (Kacprzyk, 1996). A variant of game theory is the theory of moves, fuzzified into a theory of fuzzy moves (Kandel and Zhang, 1998).

The soft agent will express its move by a linguistic variable, e.g. "high" or "low". These moves are defined by a membership function. These membership functions are either known to the other players, or not.

We will now define the extensive form of this game, following (Mas-Colell et al., 1995). Many standard conventions are omitted. The game played by the agents consists of the following items

1. A finite set of possible actions $\mathcal{A}$ and a finite set of players $\{1, \ldots, I\}$.

2. The usual game tree with decision nodes and terminal nodes.

3. An action leads to a node from its immediate predecessor in the tree. This action is unique. It is only known to the player who takes the action. The other players only know a fuzzy subset of $\mathcal{A}$. If they only know the name or symbol of the subset, we call the game a *game with linguistic*

	s_1	s_2	s_3
t_1	-3, 4	-2, 1	-3, -3
t_2	1, 1	2, -3	-3, -3

Figure 1: The normal form of a two-player fuzzy game. Player 1 can choose between strategies t_1 and t_2. Player 2 has strategies s_1, s_2 and s_3. To every strategy corresponds a single action. The players do not know each other's actions, but either a linguistic variable describing a fuzzy subset of the actions (game with linguistic information), or the membership function itself (game with membership information).

information. If they know the membership function, we will talk about a *game with membership information*.

4. A collection of information sets $\mathcal{H}$, and a function assigning each decision node to an information set. All decision nodes assigned to a single information set have the same choices available.

5. A function assigning each information set to the player who moves at the decision nodes in that set. Remember that all players have full knowledge of the nodes in their information set, but only know fuzzy subsets of the actions that led to the nodes, if these actions were taken by other players.

6. All players are equally likely to be allowed to make the first move.

7. A collection of payoff functions $u = \{u_1(\cdot), \ldots, u_I(\cdot)\}$ assigning utilities to the players for each terminal node that can be reached.

The structure of the game is common knowledge, except that the other player's actions are only known from fuzzy descriptions. The concept of strategy is still based on the information set, but the latter contains fuzzy subsets of $\mathcal{A}$ instead of player's actions. Strictly speaking, a strategy is a function from the player's information set $\mathcal{H}$ to the set of actions $\mathcal{A}$. The normal form representation does not deviate from the standard one, as the utility functions are standard. The payoff matrix has the player's strategies in the column and row headers, but the players have only fuzzy information about the actions resulting from each other's strategies. For an example, see figure 1.

The game with fuzzy information defined above is a game with incomplete information (Aumann, 1989; Mas-Colell et al., 1995). Is fuzzy information mathematically equivalent to incomplete information? In a game of incomplete information, or a Bayesian game, each agent has a type, a random variable only observed by the agent itself. The distribution of the type is common knowledge among the players. The strategy depends on the player's type. If we identify the strategy with the action it leads to, and if the membership function describing the fuzzy action is a probability distribution, then an equivalence is possible. Indeed, in our game with membership information, all players know the membership function, now a probability distribution, of what the other players are going to do. What they actually do, the element of the fuzzy set, is then equivalent to the realization of the random variable of which the probability distribution is known. The fuzzy set element is equivalent to the type.

However, a membership function is not necessarily a probability distribution, and this is one of the strengths of fuzzy systems (Zadeh, 1993). We could use

	pl.1 "medium"				pl.2 "medium"				pl.3 "high"			
linguistic	2				2				3			
membership	2	5	5	2	2	5	5	2	1	2	4	5

Figure 2: The pattern of actions taken by three players for a game with linguistic information (top row) and membership information (bottom row). The linguistic variable can take three values, "low", "medium", and "high", corresponding to 1, 2, and 3 in the pattern. For the "linguistic" row, $Q = 3$. The membership functions are defined on four points, and can take five values. For the "membership" row, $Q = 5$.

possibility theory to define the type of the players, but we will not do this at this stage. Just observe that a game with membership information is more general than a game with incomplete information when the membership functions are not probability distributions.

In a game with linguistic information, the membership functions themselves are unknown. It is as if the players get an identifier of a probability distribution, but not the distribution itself. However, they may be able to approximate the distribution from observing the players. This leads us to the next topic, learning in soft games.

3 Learning in soft games

Learning has only recently gained importance in the game theory community (Fudenberg and Levine, 1998). The emphasis is on fictitious play and, to a lesser extent, on reinforcement learning (Erev and Rapoport, 1998). There are many more powerful learning algorithms to be explored, and we use recurrent or Hopfield neural networks.

In a game with membership information or linguistic information, the players have incomplete information, and hence they will gain from learning. In fictitious play, the players exploit statistics about each other's strategies. This only works well if the strategies or corresponding actions are known explicitly.

Linguistic information or membership information can be seen as *patterns*. In a game with fuzzy information, a pattern is a vector $\mathbf{x}(t) = (x_1(t), \ldots, x_I(t))$ containing all the information that a player has about the actions of the other players at time t. If there are Q classes ("high", "low", etc.) in the information about player i, then x_i will be one of Q values, $1, 2, \ldots, Q$.

In a game with membership information, a pattern is a vector $\mathbf{x}(t) = (x_1(t), \ldots, x_I(t))$, where x_i is now a vector itself, containing the membership function. If every membership function is discretely defined on d points, the patterns in a game with membership information with be d times longer than the patterns in a game with linguistic information. To simplify the learning algorithm, the membership function can take on Q values, $1, 2, \ldots, Q$, instead of the usual values in $[0, 1]$. For an illustration, see figure 2.

The players learn by storing patterns of play in which they won in a neural network. They use this neural network to decide which action to take, when presented with the fuzzy information of the actions that the other players are taking. The neural network learning algorithm is that of Potts-glasses (Kanter, 1988). Denote the winning patterns by k_i^μ. They have either I or dI components, dependent on whether the game is with linguistic or membership information. Call this number of components N. The neural network has a weight tensor $J_{ij}^{kl}; i,j = 1,\ldots,N; k,l = 1,\ldots,Q$. This tensor is an accumulation of functions of the p winning patterns as follows

$$J_{ij}^{kl} = \frac{1}{Q^2 N} \sum_{\mu=1}^{p} m_{k_i^\mu,k} m_{k_j^\mu,l}, \quad i,j = 1,\ldots,N; k,l = 1,\ldots,Q, \tag{1}$$

with

$$m_{\sigma\tau} = Q\delta_{\sigma\tau} - 1, \tag{2}$$

and $\delta_{\sigma\tau} = 1$ for $\sigma = \tau$ and 0 otherwise. Every agent has such a tensor J, and when a new winning pattern is encountered, a new term is added to the sum in (1).

When an agent has to take an action, it calculates

$$h_{\sigma_i} = -\sum_{j \neq i}^{N} \sum_{k,l=1}^{Q} J_{ij}^{kl} m_{\sigma_i,k} m_{\sigma_j,l} \tag{3}$$

for all $\sigma_i \in \{1,\ldots,Q\}$. The σ_j are the components of the pattern of the actions that the players intend to take. The state of the neuron i is the σ_i that minimizes (3). This way, a new pattern vector is obtained. This is iterated until convergence.

In a game with linguistic information, agent i inspects the state of the neuron i. This number indicates what move to make, via the correspondence between the linguistic variable and the numbers $1,\ldots,Q$. In a game with membership information, the agent inspects the state of a group of d neurons, starting at neuron di. The numbers observed give the membership function of the optimal action to take. In both cases, a defuzzification is necessary.

In short, agents store successful patterns of actions taken by other agents, in a neural network. They use the currently observed pattern of actions as input to the neural network (3). After convergence, they obtain a suggestion for an action to take. They take this action, and if it has increased their utility, add it to the store of successful patterns (1).

Will this work? For a given input, the neural network will converge to its nearest stored pattern, provided not too many patterns are stored (Kanter, 1988). The storage rule (3) is not unlike fictitious play, it remembers past moves. On top of this, it can deal well with correlation between players. It is related to reinforcement learning in that it stores only patterns that were successful.

The convergence of learning is non-trivial (Foster and Young, 1998). However, the convergence and learning of neural networks is well understood (De Wilde, 1997) and this may help in the analysis.

4 Every agent its population of virtual agents

If agents act on behalf of humans, their number cannot be constant. Not only does the population of humans using agents change, humans constantly join and leave societies, environments. Companies come and go, interests and opportunities grow and wane.

Wherever populations are changing, evolution is a good model for population dynamics and adaptation. If a human interacts with a population via agents, it will be advantageous for this human to have a population of virtual agents as a model of the real population. Some of the virtual agents will be copies of the human's agents.

Our agents are engaged in game playing, and evolutionary game theory (Weibull, 1995) will give an insight in the population dynamics. We will integrate evolution in the neuro-fuzzy game playing described so far. Every agent will have a population of virtual agents playing the game. After some evolution, the fittest agent will be selected to make the actual move.

In our setup of the game, although the information the players receive about each other's actions is fuzzy, the utility they derive from playing is known to them. An example is a sealed-bid auction where fuzzy information about the bids is leaked among the players. Any strategic business decision would give another example. The utility is known, and fuzzy information will be known about the actions of the other players, either leaked, obtained via insider information, or analysis.

The utility was used by the neural network to classify what are the winning patterns for a particular player. It will be used again here in the replicator dynamics for the population of virtual agents.

The populations of virtual agents do not interact, the players do not exchange information at this level. Concentrate on a single population of virtual agents here, belonging to one player. The standard replicator dynamics in a homogeneous population is defined as follows (Weibull, 1995; Fudenberg and Levine, 1998). Suppose that all agents use pure strategies. If $\theta_t(s)$ is the fraction of players using pure strategy s at time t, then

$$\dot{\theta}_t(s) = \theta_t(s)[u_t(s) - \bar{u}_t], \tag{4}$$

where

$$u_t(s) = \sum_{s'} \theta_t(s') u(s, s') \tag{5}$$

is the expected payoff to using pure strategy s at time t, and $u(s, s')$ is the utility from playing strategy s against s'. If the game is a two-player game, then s' is the strategy of the single adversary playing against the player using s, and if the play is a multi-player game, s' is the combined strategy of the adversaries. Finally,

$$\bar{u}_t = \sum_{s} \theta_t(s) u_t(s) \tag{6}$$

is the average expected payoff in the population.

This continuous time dynamics increases the fraction of players having above average utility. It is easy to define a more realistic, discrete time version of these dynamics (Weibull, 1995; Fudenberg and Levine, 1998). Moreover, equation (4) will not be used directly. The strategy of an agent is a direct consequence of the weight matrix J, see (1), of the neural network used by the agent. The weight matrix defines, for a given fuzzy information set, the action the agent will take. When the virtual agents interact, those whose weight matrices give rise to an above average utility will duplicate, and those below average will be removed from the population. This is our replicator dynamics in practice.

The combination of neural learning and evolutionary dynamics is potentially very powerful. The fitness or utility depends on how well the neural network performs. The successful neural networks get duplicated. This means that what is learned is passed on to the offspring. This does not happen in the natural world, where evolution is Darwinian (Smith, 1989). The soft game described here tries to out-smart nature.

5 Conclusion

This paper uses fuzzy systems, neural learning, and evolutionary computing to construct agents that have a human-like intelligence for playing games. They can use language to express moves, learn patterns, and survive in a competitive environment.

The algorithm has been described, and placed in the context of related research. A simplified version of this algorithm has been successfully applied to an auction. This will be reported elsewhere. Further analysis is necessary, especially about the convergence properties of the learning.

The fuzzy, neural, and evolutionary algorithms are really integrated, not just modules in a library. As far as I am aware, this is the first application of all aspects of soft computing to game theory and intelligent agents.

I would like to thank N. Azarmi and B. Azvine for their continuing interest in soft agents.

References

Aubin, J.-P. (1981). Cooperative fuzzy games. *Mathematics of Operations Research*, 6(1):1–13.

Aumann, R. J. (1989). Game theory. In Eatwell, J., Milgate, M., and Newman, P., editors, *Game Theory*, pages 1–67. Norton, New York.

De Wilde, P. (1997). *Neural Network Models, second expanded edition.* Springer Verlag, London.

Erev, I. and Rapoport, A. (1998). Coordination, "magic", and reinforcement learning in a market entry game. *Games and Economic Behavior*, 23:146–175.

Foster, D. P. and Young, H. P. (1998). On the nonconvergence of fictitious play in coordination games. *Games and Economic Behaviour*, 25:79–96.

Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games.* MIT Press, Cambridge, Massashusetts.

Kacprzyk, J. (1996). Supporting consensus reaching under fuzziness via ordered weighted averaging operators. In Chen, Y. Y., Hirota, K., and Yen, J. Y., editors, *Soft Computing in Intelligent Systems and Information Processing*, pages 453–458, Piscataway. IEEE.

Kandel, A. and Zhang, Y.-Q. (1998). Fuzzy moves. *Fuzzy Sets and Systems*, 99:159–177.

Kanter, I. (1988). Potts-glass models of neural networks. *Physical Review A*, 37(7):2739–2742.

Mamdani, A. and Pitt, J. (1998). The growing population of software agents: A manifesto. Preprint.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, Oxford.

Ossowski, S. (1999). *Co-ordination in Artificial Agent Societies*. Springer, Berlin.

Ponsard, C. (1987). Fuzzy mathematical models in economics. *Fuzzy Sets and Systems*, 28:273–283.

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge, Massachusetts.

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1, Foundations*. MIT Press, Cambridge, Massachusetts.

Smith, J. M. (1989). *Evolutionary Genetics*. Oxford University Press, Oxford.

Weibull, J. W. (1995). *Evolutionary Game Theory*. MIT Press, Cambridge, Massachusetts.

Zadeh, L. A. (1993). Possibility theory and soft data analysis. In Dubois, D., Prade, H., and Yager, R. R., editors, *Fuzzy Sets for Intelligent Systems*, pages 478–508. Morgan Kaufmann, San Mateo, California. Reprinted.

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3):77–83.