Cooperation without exploitation between self-interested agents

Steven Damer and Maria Gini

Abstract We study how two self-interested agents that play a sequence of randomly generated normal form games, each game played once, can achieve cooperation without being exploited. The agent learns if the opponent is willing to cooperate by tracking the attitude of its opponent, which tells how much the opponent values its own payoff relative to the agent's payoff. We present experimental results obtained against different types of non-stationary opponents. The results show that a small number of games is sufficient to achieve cooperation.

1 Introduction

We study cooperation between two self-interested agents, where an agent may be hostile but may also be willing to give up part of its expected payoff to provide a benefit to its opponent. Following game theory an agent should select the action that provides its own highest expected payoff, without regard for the opponent's outcome. However, many forms of cooperation are observed in evolution [13], and in iterated games [2]. Social Value Orientation theory [11] recognizes that people's behaviors depend on their personalities, and that people with a prosocial orientation highly regard the payoffs of others they interact with.

We study agents that play a sequence of non-zero-sum normal form games, each game played only once by the same two players. Playing against the same opponent enables the agents to observe each other, but since the game changes each time, it is harder to detect if the opponent is cooperative. Our setting is similar to stochastic games [15], but to simplify the learning process the payoff distribution is known to both players and each game is independent of the previous state and agents actions.

The main contributions of this paper are the use of a regularized particle filter to learn the willingness of the opponent to cooperate and experimental results against

Department of Computer Science and Engineering, University of Minnesota, 200 Union St SE, Minneapolis, MN 55455, USA e-mail: damer@cs.umn.edu,,gini@cs.umn.edu

different types of non-stationary opponents. We show that an agent can predict the behavior of its opponent within the limits implied by the rate at which the opponent changes, and achieve a cooperative outcome without risking significant exploitation.

2 Background on cooperation model

We use the model presented in [6] and extend the work in [7] to non-stationary opponents. In this model an agent adopts an *attitude* towards its opponent, which determines how much weight it attaches to its opponent's payoff relative to its own payoff. An attitude is a real number in the range [-1, 1]. An attitude of 1 means that the player wants to maximize social welfare, 0 that the agent is indifferent to the opponent's payoff, and -1 that the agent is spiteful. The attitude of the opponent is private information and must be learned. This model is functionally equivalent to Social Value Orientation theory [11] with a different parametrization, using attitude values instead of an angle representing the ratio of utility of the agent's payoff and the opponent's payoff.

Let's call the agents x and y, and their attitudes A^x and A^y respectively. To select its action, each agent computes a modified game. In the modified game agent x has a new payoff function P'^x defined as $P'^x_{ij} = P^x_{ij} + A^x \times P^y_{ij}$, where P^x_{ij} is the payoff in the original game for player x and P^y_{ij} is the payoff for the opponent when they choose respectively actions i and j. Similarly agent y computes a modified payoff function using its attitude A^y. Each agent selects an action which maximizes its score in the modified game, but receives its payoff from the original game.

An agent acting according to this model uses three parameters, the agent's *attitude*, an estimate of the opponent's attitude, which we call *belief*, and a *method* of choosing an action in the modified game. For simplicity we assume that agents play a strategy which is part of a Nash equilibrium. This is not the only choice, but it is convenient since it limits the method to a discrete set. In this context, method is simply the choice of which Nash equilibrium is used.

To indicate its willingness to cooperate, an agent first has to learn the attitude used by the opponent and then sets its own attitude to be higher than the estimated attitude of the opponent by a *reciprocation* value. Specifically, a reciprocating agent x sets its own attitude A^x to be equal to B^x, its estimate of the opponent's attitude, plus a reciprocation level R, $A^x = B^x + R$. If this results in a value below 0 or above 1, the value is set to 0 or 1. The value is not allowed to drop below 0 because attempting to take revenge on an opponent will reduce the agent's score. The value is not allowed to increase beyond 1 because two agents with attitudes higher than one can result in inefficiencies as each agent attempts to force its opponent to take a higher share of the payoff.

Figure 1 shows how different reciprocation levels affect the payoffs. We can see that any non negative reciprocation level produces cooperation and higher payoffs. The reciprocation level we use in our experiments is .1, since it limits the potential loss but is sufficient to lead to full cooperation.

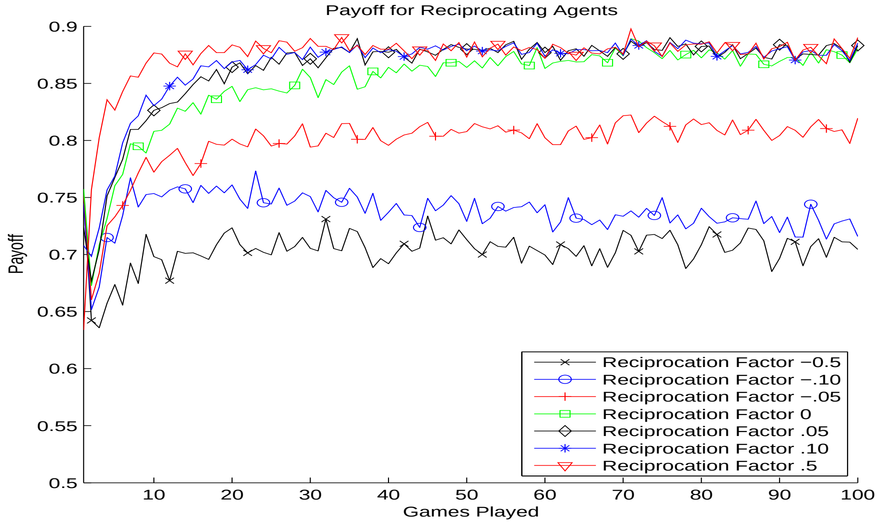

Fig. 1 Payoffs for various choices of R when two reciprocating agents play each other.

3 Learning

In every round the agent observes the payoff matrix of the game, chooses its own action, and observes the action chosen by its opponent. From that information, it needs to learn a probability distribution over the attitude, belief, and method of the opponent. Due to the complex interactions between those parameters and the game being played, it is not possible to do this analytically.

Instead we use a particle filter, which represents a probability distribution with a number of samples drawn from it (see Algorithm 1). The distribution represented by the particles is a discrete distribution with probability of each particle proportional to its weight. When an observation is made, each particle's weight is updated by multiplying it by the probability assigned to the observation by that particle.

We use a regularized particle filter [12], which resamples from a continuous instead of a discrete distribution. As observations are made, the relative probability of the particles changes. At the extreme, if one particle has all the weight, the distribution is effectively represented by a single particle. To avoid this, when the effective number of particles drops below a threshold, a new set of particles is drawn by sampling from the existing distribution and adding noise.

Noise is drawn from a Gaussian distribution with 0 mean and standard deviation equal to $N^{-1/6}$ times the standard deviation of the particle set, where N is the number of particles. This is an improvement over the approach in [6] because it doesn't require knowledge of the distribution from which games are drawn and increases accuracy. Method is a discrete value, so it cannot be perturbed with Gaussian noise. Instead, with some probability we change it to a random new method. The optimal probability is found using a technique called Leave-One-Out, where we select the

Algorithm 1 *RegularizedParticleFilter*

```
Generate initial set $P$ of $N$ particles
for particle $p \in$ P do
    Assign attitude $att_p$, belief $bel_p$, and method $method_p$ of particle $p$ from prior
    Assign weight $weight_p = 1/N$
end for
while presented with data do
    Observe opponent's action $M$ in game $G$
    Compute effective number of particles
    $N_{eff} = 1/[\sum_{p \in P} weight_p{}^2]$
    if $N_{eff} >$ threshold then
        for particle $p \in$ P do
            Compute probability of opponent's action $M$ in game $G$, $prob_p$, given $att_p$, $bel_p$, and
            $method_p$
            Update $weight_p$=$weight_p \times prob_p$
        end for
    else
        Compute standard deviation $Std_{att}$ of $att_p$ and standard deviation $Std_{bel}$ of $bel_p$
        $h = N^{-1/6}$
        Compute perturbation probability $pp$ for $method_p$
        while accepted particles < total particles do
            Select particle $p$ from $P$ with probability proportional to $weight_p$ and create new
            particle $p'$
            Assign attitude and belief adding Gaussian noise
            $att_{p'} = att_p + h \times N(0, Std_{att})$; $bel_{p'} = bel_p + h \times N(0, Std_{bel})$
            With probability $pp$, $method_{p'}$ = random method else $method_{p'} = method_p$
            Compute $prob_{p'}$= probability of opponent's action $M$ in $G$ given $att_{p'}$, $bel_{p'}$, and
            $method_{p'}$
            Accept $p'$ with probability $prob_{p'}$
        end while
    end if
end while
```

probability that gives the highest likelihood of resampling the current distribution from a distribution created by removing one particle from the current set.

We use 400 particles with attitude and belief drawn from a Gaussian distribution centered at 0 with the identity matrix as a covariance matrix, and method drawn from a uniform distribution over the list of methods under consideration. We assign each particle a weight of .0025 and resample if the effective number of particles goes below 200.

4 Experimental Results

We now present experimental results obtained against several classes of non-stationary opponents. Experiments against a stationary opponent and in self-play have been reported in [7]. We measure *model accuracy*, i.e. the Euclidean distance between the estimate of attitude and belief of the opponent and their true values, and *prediction* *accuracy*, i.e. the Jensen/Shannon divergence between the prediction and the actual probability distribution used by the opponent.

We use randomly generated normal form games with 16 actions per player, and payoffs uniformly distributed between 0 and 1, as in [6]. We have found experimentally that 16 actions provide a good balance between model and prediction accuracy. Model accuracy increases with the number of actions, since each action is more informative as more alternatives are rejected, but prediction accuracy decreases because the space of the predicted distribution increases.

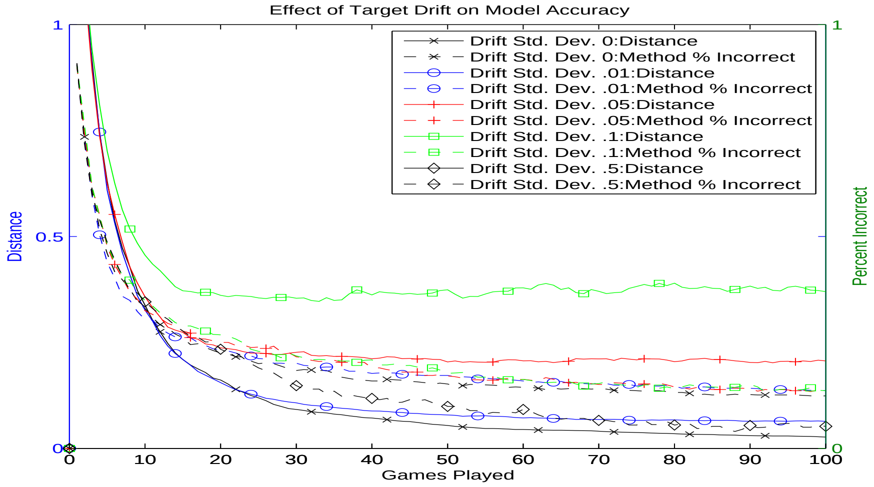

Fig. 2 Model accuracy when learning a randomly drifting target. Results aggregated over 100 sequences of games. Learning targets drawn from a 0 mean Gaussian.

We have explored two types of non-stationary opponents, both separately and in combination. The first type changes its values for attitude and belief according to *random drift* with a Gaussian distribution. This models an agent which gradually adjusts its strategy. The second type changes by *redrawing* values from the prior with a fixed probability. This models an agent which changes according to some threshold, or which may be replaced without notice. The third type combines *random drift and redrawing*.

Figure 2 and Figure 3 show the effect of various levels of drift on model and prediction accuracy. Unsurprisingly, as drift increases, accuracy decreases. An odd effect is that the accuracy for the method increases for a high drift. This occurs because there are regions in the model space in which method does not affect the agent's actions. With a high drift rate, the opportunity to get out of those regions outweighs the difficulty caused by the rapid change in values. The model error increases as drift increases, but still provides a reasonable estimate of the opponent's intentions. With a .5 drift, the prediction accuracy is equivalent to being able to correctly identify one action out of sixteen 55% of the time.

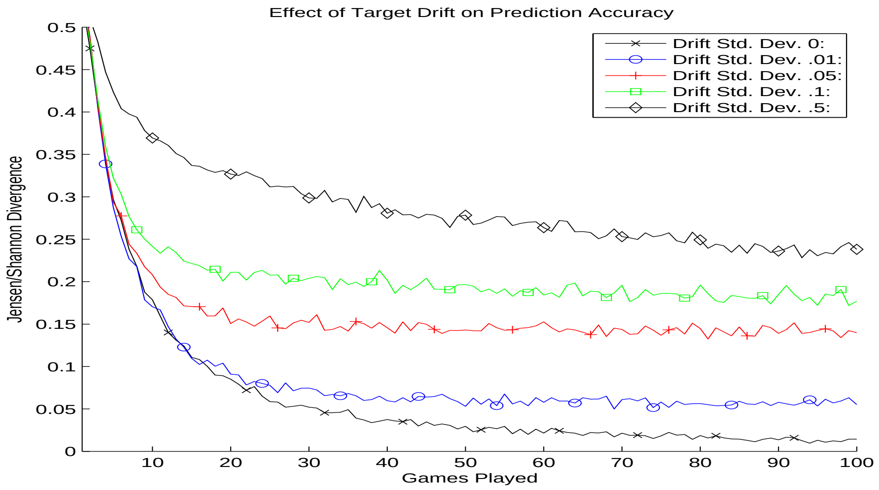

Fig. 3 Prediction accuracy when learning a randomly drifting target. Results aggregated over 100 sequences of games. Learning targets drawn from a 0 mean Gaussian.

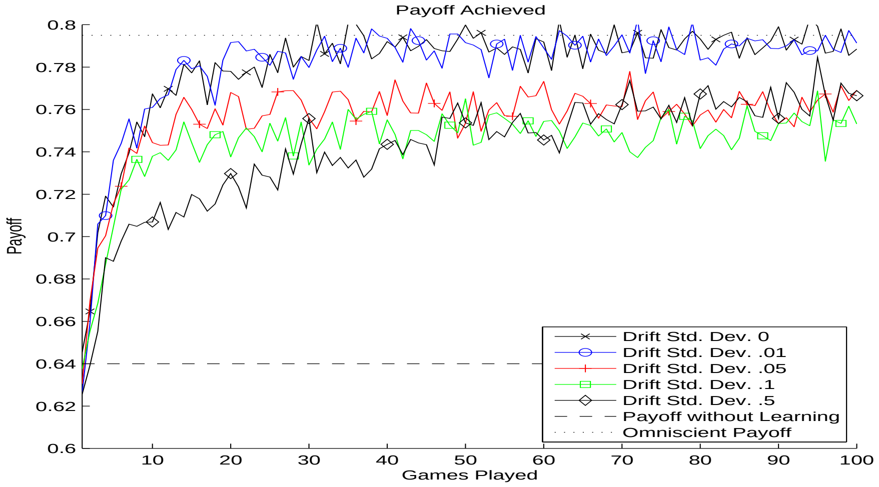

Fig. 4 Payoffs of agent against a randomly drifting target.

Figure 4 shows the payoffs, which remain high even with a large drift. Omniscient payoff is the expected payoff when the agent knows the opponent's attitude, belief, and method. Payoff without learning is the expected payoff when the agent best responds to the prior over the opponent's attitude, belief, and method. Both values were found empirically.

Figure 5 shows the effect of randomly redrawing the target from the prior. Random resets make learning considerably more difficult. No learning is possible with a

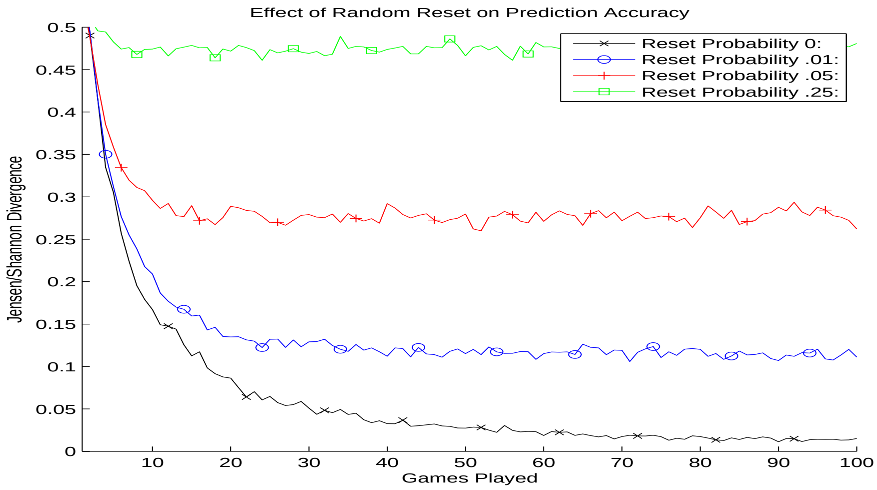

Fig. 5 Prediction accuracy when learning a randomly resetting target. Results aggregated over 100 sequences of games.

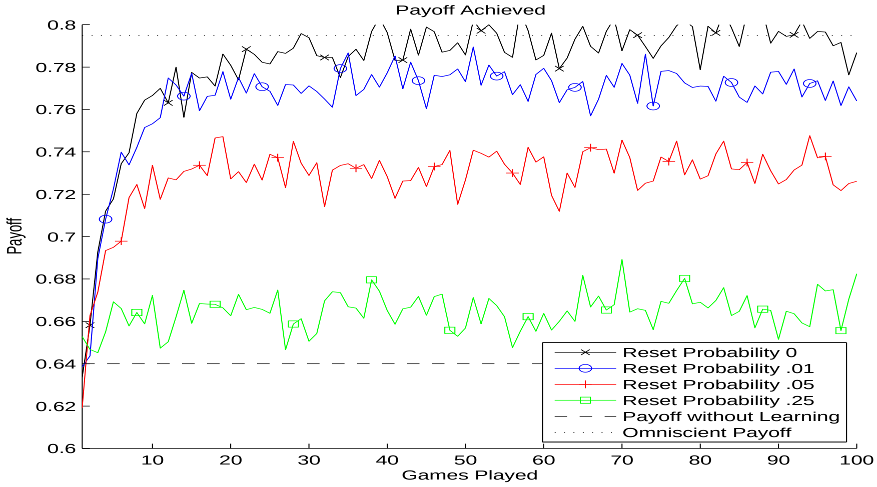

Fig. 6 Payoffs of agent against a randomly resetting target.

reset probability greater than .05. This is because the agent does not have sufficient time to learn between resets. For example, with a reset probability of .25, on average there are 4 games between resets, which is not sufficient to fully learn the target. Figure 6 shows the corresponding payoffs. As expected, higher reset probabilities result in lower payoffs.

Figure 7 shows the effect of combining both types of nonstationarity. Unsurprisingly, it is much harder to learn when the parameters are changing slightly in ev-

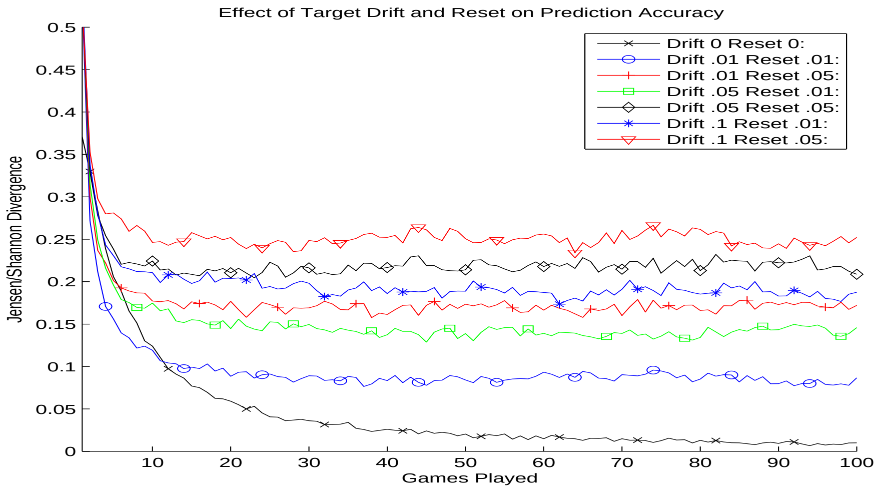

Fig. 7 Prediction accuracy when learning a randomly drifting and resetting target. Results aggregated over 50 sequences of games.

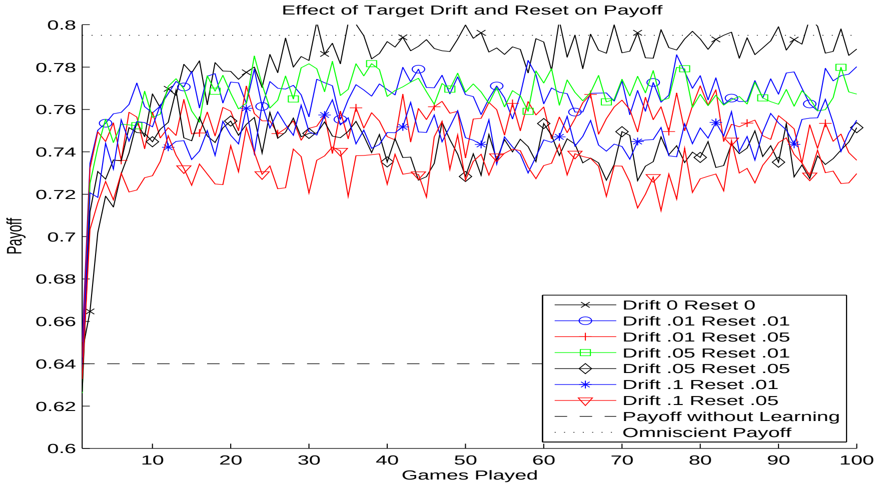

Fig. 8 Payoffs against a randomly drifting and resetting target.

ery game, and occasionally radically. However, it still possible to make reasonable predictions. This is particularly valuable because it implies the ability to track the behavior of much more sophisticated agents by modelling their changes in attitude and belief as random behavior. Figure 8 shows the corresponding payoffs. Despite the large prediction error, the payoffs of the agents are higher than if no learning had occurred.

5 Related Work

Our model of cooperation is based on models developed to explain human cooperation in normal form games. Valavanis [16] proposed a modification of a normal form game to reflect an agent's preferences over its opponent's utility. Frohlich [9] pointed out that this can lead to an ill-defined utility function, and proposed restricting an agent's preferences to its opponent's consumption instead of utility. Fitzgerald [8] introduced a utility which is linear in the opponent's payoff, and pointed out that positive attitudes will not necessarily reduce the level of contention between agents. An overview of some of the many different models which have been used to explain human behavior can be found in [3]

Reciprocation is recognized as an effective way to motivate an opponent to cooperate, as demonstrated in the Tit-for-Tat strategy for iterated prisoner's dilemma [2]. However, Tit-for-Tat does not perform well in noisy environments. One way to handle that [1] is to track deviations from a learned opponent policy, resetting when it becomes apparent that the learned policy is inaccurate. This problem is similar to the one we deal with, where the environment is the source of noise in the observations, instead of nonstationarity of the opponent.

Most research on learning for agents which play normal form games has focused on repeated play of a single game against a stationary opponent with the goal of finding either an equilibrium or a Pareto-optimal outcome in self-play. Fudenberg [10] provides a good overview of fictitious play, which explores the effects when agents attempt to learn their opponents actions and then choose the best response. Reinforcement learning is a popular technique for dealing with normal form games. Unlike the techniques presented in this paper, it does not require a complete description of the game, however it requires repeated interactions within a single game in order to learn the optimal actions. M-Qube [5] is a reinforcement learning algorithm which balances best response, cautious learning to bound losses, and optimistic learning by looking for strategies with potentially high returns even if risky. The algorithm provably bounds losses in repeated games but it requires playing thousands of times.

AWESOME [4] is the first algorithm guaranteed to learn to play optimally against stationary opponents and to converge to a Nash equilibrium in self play. It also learns to play optimally against opponents that eventually become stationary. To guarantee convergence in self-play, it assumes all agents play the same Nash equilibrium. By limiting the history the opponent can use, the algorithm described in [14] learns against non-stationary opponents by playing thousands of repeated games. We are interested in methods that learn much more rapidly.

6 Conclusions and Future Work

We have described a regularized particle filter to learn model parameters and we have evaluated the performance of the learning algorithm against non-stationary opponents. The model is effective for achieving cooperation in situations where cooperative actions are not obvious.

This paper shows results for random non-stationary opponents. It would be interesting and useful to examine the performance of this model when applied to other types of non-stationary opponents. Another point to investigate is a method to compute appropriate reciprocation levels from the game and from the probability distribution over the model parameters of the opponent, instead of using a preset level.

References

1. Au, T.C., Kraus, S., Nau, D.: Symbolic noise detection in the noisy iterated chicken game and the noisy iterated battle of the sexes. In: Proceeding of the First International Conference on Computational Cultural Dynamics (ICCCD-2007) (2007)
2. Axelrod, R.M.: The evolution of cooperation. Basic Books (1984)
3. Camerer, C.F.: Progress in behavioral game theory. The Journal of Economic Perspectives **11**(4), 167–188 (1997)
4. Conitzer, V., Sandholm, T.: AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. Machine Learning **67**(1–2), 23–43 (2007)
5. Crandall, J.W., Goodrich, M.A.: Learning to compete, cooperate, and compromise using reinforcement learning. Machine Learning (2010)
6. Damer, S., Gini, M.: Achieving cooperation in a minimally constrained environment. In: Proc. of the Nat'l Conf. on Artificial Intelligence, pp. 57–62 (2008)
7. Damer, S., Gini, M.: Extended abstract: Friend or foe? detecting an opponent's attitude in normal form games. In: Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems (2011)
8. Fitzgerald, B.D.: Self-interest or altruism. Journal of Conflict Resolution **19**, 462–479 (1975)
9. Frohlich, N.: Self-Interest or Altruism, What Difference? Journal of Conflict Resolution **18**(1), 55–73 (1974)
10. Fudenberg, D., Levine, D.K.: The Theory of Learning in Games. MIT Press (1998)
11. McClintock, C.G., Allison, S.T.: Social value orientation and helping behavior. Journal of Applied Social Psychology **19**(4), 353–362 (1989)
12. Musso, C., Oudjane, N., Legland, F.: Improving regularized particle filters. In: A. Doucet, N. de Freitas, N. Gordon (eds.) Sequential Monte Carlo Methods in Practice, pp. 247–271. Springer-Verlag, New York (2001). URL citeseer.ist.psu.edu/musso01improving.html
13. Nowak, M.A.: Five rules for the evolution of cooperation. Science **314**, 1560–1563 (2006)
14. Powers, R., Shoham, Y., Vu, T.: A general criterion and an algorithmic framework for learning in multi-agent systems. Machine Learning **67**(1–2), 45–76 (2007)
15. Shapley, L.S.: Stochastic games. Proceedings of the NAS **39**, 1095–1100 (1953)
16. Valavanis, S.: The resolution of conflict when utilities interact. The Journal of Conflict Resolution **2**(2), 156–169 (1958). URL http://www.jstor.org/stable/172973