Progress in Nonlinear Dynamics and Chaos
Vol. 1, 2013, 47-53
ISSN: 2321 – 9238 *(online)*
Published on 3 June 2013
www.researchmathsci.org

Progress in
Nonlinear
Dynamics and Chaos

Role of Iterated Prisoner's Dilemma in Genetic Based Machine Learning

[1]M.Thiyagarajan* and *[2]J.Rajkumar

[1]Dean Research, NGI
Nehru Institute of Engineering and Technology
Coimbatore – 641 105, Tamilnadu
m_thiyagarajan@yahoo.com

[2]Department of Master of Computer Applications
Nehru Institute of Engineering and Technology
Coimbatore – 641 105, Tamilnadu
rajkumar.kj@gmail.com

Received 25 May 2013; accepted 2 June 2013

Abstract. Several strategies have been followed by most of earlier researchers in the field of machine learning. Agarwal has connected Machine Learning with Iterated Prisoner's Dilemma Problem [IPD]. Holland has proposed basic directions to explore goal of genetic operators in the study of machine learning. Axelrod connected Genetic Algorithm with IPD. We integrate these basic approaches to give a novel setup for Genetic based Machine Learning the particular decision analysis on game theory.

Keywords: Genetic Based Machine Learning, Genetic Algorithms, Iterated Prisoner's Dilemma, Decision under uncertainty.

1. Introduction

The interaction of learning and evolution is a topic of great interest in evolutionary computation. It plays an important role in application areas such as multi-agent systems, economics, politics, and biological modeling. When learning and evolution interact in an evolutionary game scenario. The simulation environment is similar to Axelrod's [1] well-known IPD simulation study environment.

1.1. Basic Concepts and Results [4]

Definition 1. GA and Optimizations Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics.

They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. Random search algorithms have achieved increasing popularity as

researchers have recognized the short comings of calculus-based and enumerative schemes.

Four differences separate genetic algorithms from more conventional optimization techniques: Direct manipulation of a coding, search from a population, not a single point, search via sampling, a blind search using stochastic operators, not deterministic rules. Although there is still much to learn about the theory and application of genetic algorithms, the diversity and effectiveness of genetic algorithms in many problem areas is cause for optimism.

Definition 2. (*Classifier Systems*) Classifier systems are a form of Genetic – Based Machine Learning (**GBML**) system, combining a simple, parallel production system based on string rules, an appointment of credit algorithm modeled after an information – based service economy, and genetic algorithms. Classifier systems and their derivatives are finding increasing application in science, engineering, and business circles.

In the most general classifier systems, classifiers send messages that are placed on a message list, thereby activating other classifiers or action triggers called effectors. The presence of a central message list provides a centralized communication channel. Since space is limited on the message list, some method must exist for choosing among competing messages.

ML (Machine Learning) and IPD (Iterated Prisoner's Dilemma) [1]

In the iterated game, player strategies are rules that determine a player's next move in any given game situation (which can include the history of the game to that point). Each player's aim is to maximize his total payoff over the series. If you know how many times you are to play, then one can argue that the game can be reduced to a one-shot Prisoner's Dilemma [8]. The argument is based on the observation that you, as a rational player will defect on the last iteration – that is the sensible thing to do because you are in effect playing a single iteration. The same logic applies to your opponent. Knowing that your opponent will therefore defect on the last iteration, it is sensible for you to defect on the second to last one, as your action will not affect his next play. Your opponent will make the same deduction. This logic can be applied all the way back to the first iteration. Thus, both players inevitably lock into a sequence of mutual defections. One way to avoid this situation is to use a regime in which the players do not know when the game will end. Nature could toss a (possibly biased) coin to decide. Different Nash equilibria are possible, where both players play the same strategy. Some well-known examples are:

(1) **Tit-for-tat:** cooperate on the first move, and play the opponents previous move after that;

(2) **Grim**: cooperate on the first move, and keep cooperating unless the opponent defects, in which case, defect forever;

(3) **Pavlov**: cooperate on the first move, and on subsequent moves, switch strategies if you were punished on the previous move.

In our version of the problem we assume that K is large, so that the opponents have enough opportunities to learn from each other. Also, the payoff matrix is given by table-1:

(C,C) ! (3; 3) (C,D) ! (0; 5) and (D,C) ! (5; 0) (D,D) ! (1; 1),

where (C;C) ! (3; 3) means that if both cooperate, both get a payoff of 3, and so on. In 1980, Robert Axelrod staged two round-robin 'tournaments' between computer programs designed by participants to play IPD. Many sophisticated programs were submitted. In each case, the winner was Anil Rapaports submission, a program that simply played Tit-for-Tat. In 1987, Axelrod carried out computer simulations using a genetic algorithm (nowadays it would be called a co-evolutionary simulation) to evolve populations of strategies playing the IPD against each other. In these simulations, tit-for-tat-like strategies often arose as the best but it was proved that they were not optimal. In fact, Axelrod used this to illustrate that there is no 'best' strategy for playing the IPD in such an evolving population, because success depends on the mix of other strategies present in the population. Axelrod's simulations illustrate a different approach to studying the IPD one in which the players are not perfectly rational, and solutions evolve rather than being deduced.

GA (Genetic Algorithm) and IPD (Iterated Prisoner's Dilemma) [6]

In its simplest form, each of two players has a choice of cooperating with the other [or] defeating. Depending on the two players' decisions, each receives payoff according to a payoff matrix similar to the one shown in this figure. A typical payoff matrix in the Prisoner's dilemma problem

		Player 2	
	Decision	Cooperate	Defect
Player1	Cooperate	(R=6,R = 6)	(S=0,T=10)
	Defect	(T=10,S=0)	(P=2, P=2)

Table 1. R: REWARD, S: SUCKER, T: TEMPTATION, P: PENALTY

When both players cooperate they are both rewarded at an equal intermediate level (The returned, R). When only one player defects, he receives the highest level payoff (the temptation, T) while the other player gets the sucker's just deserts (the sucker, S), when both players defect they each receive an intermediate penalty (the penalty, P).

Axelrod allowed decision rules to depend upon the behavior of both parties during the previous three moves. On each of those moves there are, of course, four possibilities. Players can cooperate (CC [or] R for reward), the other player can defect (CD [or] S for sucker), the first player can defect (DC or T for temptation), or both player can defect (DD [or] P for penalty). To code a particular strategy, Axelrod first coded the particular behavioral sequence as a three – letter string. For example RRR would represent the sequence where the first player was played for a sucker twice and finally defected. The three – letter sequence was then used to generate a number between 0 and 63 by treating the code as an integer base 4 where the behavioral alphabet is decoded in the following way:

CC= R = 0; DC = T = 1; CD =S =2; DD =P=3

Illustration 1. Three mutual defections (PPP) would decode to a 63 using this coding, Axelrod then defined a particular strategy (over the past three moves) as a 64-bit binary string of C's (Co operate) and D's (defect) where the ith C or D corresponds to the ith behavioral sequence. Using this scheme, for example, a Din position 0 would be decoded as a rule of the form **RRR -> D** and a rule of the form **RRP -> C.** In an earlier version Axelrod assumed initial mutual cooperation, but he found that early game behavior was very important to developing strategies that could beat tit –for- tat. Together each of the 70-bit strings thus represented a particular strategy with 64- bits for the rules and six bits for the premises. Each of the string strategies in a population of size 20 played each of the eight opponents in a game of 151 moves. A fitness measure was calculated by taking a weighted average of the point scores against each of the eight opponents where the weights were chosen to closely match tournament conditions. From a random start, the genetic algorithms discovered strategies that beat the overall performance of tit- for- tat.

Classifier systems are a form of genetics- based machine learning (GBML) system, combining a simple, parallel production system based on string rules, an apportionment of credit algorithm modeled after an information – based service economy, and genetic algorithms.

Classifier systems and their derivatives are finding increasing application in science, engineering and business circles. In the most general classifier systems, classifier send messages that are placed on a message list, thereby activating other classifiers or action triggers called effectors. The presence of a central message list provides a centralized communication channel. Since space is limited on the message list, some method must exist for choosing among competing messages.

2. Model of Genetic based Machine Learning of Holland [5]

The mathematical framework proposed here holds many elements in common with the mathematics used to study other adaptive systems such as economies, ecologies, physical systems far from equilibrium, immune systems, etc. In each fields, there are familiar topics, with mathematical treatments, that have counterparts in each of the other fields. Even an abbreviated list of such topics… is impressive: Niche exploitation, functional convergence and enforced diversity [ecology]. Competitive exclusion [ecology]. Symbiosis, parasitism, mimicry [ecology] Epistasis, linkage and revision and redefinition of "building blocks" [genetics]. Linkage and "hitchhiking" [genetics]. Multifunctionality of "building blocks" [genetics and comparative biology]. Polymorphism [genetics]. Assorative recombination ("triggering of operators) {genetics and immunology]. Hierarchical organization [phylogenetics, development biology, economics and AI]. Tagged clusters [biochemical genetics, immunogenesis and adaptive systems theory]. Adaptive radiation and the "founder" effect of generalists [ecology and phylogenetics]. Feedback from coupled procedures [biochemistry and biochemical genetics]. "Retained earnings" as a function of past success and current purchases [economics]. "taxation" as a control on efficiency [economics]. "exploitation" (production) vs "exploration" (research) [economics and adaptive systems theory]. "tracking" vs "averaging" [economics and adaptive systems theory]. Implicit evaluation of "building blocks" [adaptive systems theory]. "basins of attraction" and behavior far from equilibrium [physics]. Amplification of small biases submerged in noise on "slow" passage through a

critical point [physics]. Any complex system constructed from components interacting in a nonlinear fashion will, in one regime or another, exhibit all of these features. A general mathematical theory of such systems would explain both the pervasiveness of these features and the relations between them.

Decision under uncertainty [3]

Decision under uncertainty, as under risk, involves alternative actions whose payoffs depend on the (random) states of nature. Specifically, the payoff matrix of a decision problem with m alternative actions and *n* states of nature can be represented as [$v(a_i, s_j)$]

The element a_i represents action *i* and the element s_j represents state of nature *j*. the payoff or outcome associated with action a_i and state s_j is $v(a_i, s_j)$. The difference between making a decision under risk and under uncertainty is that in the case of uncertainty, the probability distribution associated with the sales S_j, j= 1,2,….*n*, either unknown or cannot be determined. This lack of information has led to the development of the following criteria for analyzing the decision problem:

1. Laplace, 2. Minimax, 3. Savage, 4. Hurwicz

The **Laplace** criterion is based on the **principle of insufficient reason.** Because the probability distributions are not known, there is no reason to believe that the probabilities associated with the states of nature are different. The alternatives are thus evaluated using the optimistic assumption that all states are equally likely to occur – that is, $P\{S_1\} = p_1$ $\{S_2\} = \ldots = p\{S_n\} = 1/n$, given that payoff $v(a_i, s_j)$ represents gain, the best alternative is the one that yields

$$\max_{a_i} \left\{ \frac{1}{n} \sum_{j=1}^{n} v\left(a_i, s_j\right) \right\}$$

If $v(a_i, s_j)$ represents loss, then minimization replaces maximization

The **maximin (minimax)** criterion is based on the conservative attitude of making the best of the worst possible conditions. If $v(a_i, sj)$ is loss, then we select the action that corresponds to the minimax criterion

$$\min_{a_i} \left\{ \max_{s_j} v\left(a_i, s_j\right) \right\}$$

If $v(a_i, s_j)$ is gain we use the maximin criterion given by

$$\max_{a_i} \left\{ \min_{s_j} v\left(a_i, s_j\right) \right\}$$

The **Savage regret** criterion aims at moderating conservatism in the minimax (maximin) criterion by replacing the (gain or loss) payoff matrix $v(a_i, s_j)$ with a loss (or regret) $r(a_i, s_j)$ matrix, using the following transformation:

$$r\left(a_i, s_j\right) = \begin{cases} v\left(a_k, s_j\right) - \min_{a_k} \left\{ v\left(a_k, s_j\right) \right\}, \\ \max_{a_k} \left\{ v\left(a_k, s_j\right) \right\} - v\left(a_k, s_j\right), \end{cases}$$

The last test to be considered is HurwicZ criterion, which is designed to reflect decision-making attitudes, ranging from the most optimistic to the most pessimistic (or

conservatitive). Define $0 \leq \alpha \leq 1$, and assume that $v(a_i, s_j)$ represents gain. Then the selected action must be associated with

$$\max \left\{ \alpha \max_{s_j} v\left(a_i, s_j\right) + (1-\alpha) \min_{s_j} v(a_i, s_j) \right\}$$

The parameter α is called the index of optimism. If α = 0, the criterion is conservative because it applies the regular minimax criterion. If α = 1, the criterion produces optimistic results because it seeks the best of the best conditions. We can adjust the degree of optimism (or Pessimism) through tegarding optimism and pessimism, α in the specified (0,1) range. In the absence of strong feeling regarding optimism and pessimism, α=5 may be an appropriate choice. If $v(a_i, s_j)$ represents loss, then the criterion is changed to

$$\min_{a_i} \left\{ \alpha \min_{s_j} v\left(a_i, s_j\right) + (1-\alpha) \max_{s_j} v(a_i, s_j) \right\}$$

3. Our Proposed Model

Based on the observation of Holland GBML components there is a need in revising the code for individuals in the population of search algorithm GA. This will be done using bucket brigade problem and other types of machine learning rules. Besides doing this change are to made on the other operated of GA with this modification Axelrod's approach to IPD may be used build strategies for GBML taking extra game theory strategies IPD.

Data analysis and Algorithms (Discussion):

Step 1: Creation for new induvals for selection operators [2].

Classifier systems are as the crow flies promote. So we set various tissues on the classifier system skeleton as we enlarge a Simple Classifier System (SCS) in the Pascal or Java Programming Language. The data declarations required to implement a population of classifiers and its environmental message in the SCS. The classifier type classtype is defined as a record containing a condition c, an action a and a number of scalar variables. The classifier type contains a number of variables of type real: strength and bid is self explanatory and ebid is classifier's effective bid (EB earlier) and also one additional variable is the Boolean variable matchflag. Matchflag is equal to true, when the classifier's condition is matched by the current environmental message. The condition type is defind as an array of type trit – a ternary digit, an integer between -1 and 1, where a – 1 is interpreted as the wildcard character and both o and 1 are interpreted as is. In the SCS the action type is taken as type bit. In the multiplexer problem, the classifier system is learning a Boolean function and must output a 1 or a 0. The datatype classarray as array of classifiers (an array of classtype) and this array of classifiers in population are recordtype, poptype (population of classifiers). We can create an array of classifiers (type classarray) called classifier and include integer variable are n classifier, n position, the number of positions in the condition respectively. And then include a number of real type population parameters in the population type poptype. Population type creates a single instance called population. Envmessage creates a single environmental message called envmessage. This envmessage to represent our environmental message. We also create an auxiliary data structure to record which classifiers are currently matched by the environmental message. This structure is of type class list and is called the matchlist.

With these we can finalize a module for the first step of the model.

Step 2: Modification cross over mutation operator of GA

After creating the induvals of the population a selection procedure is designed using probability rules. The second operator of GA namely the crossover will contain two or more bit positions of the member will be selected. After this the mutation operator is designed a need of machine learning (ML) process.

Step 3: Specific rules of GBML and relevance to IPD strategies

The common strategies adapted in machine learning (ML) are taken care of in Holland machine learning rules. We take the Tit-for-tat, Grim & Pavlov and connect with IPD strategies.

Step 4: Creation of GBML rules composed of revised operators and IPD.

Repeating steps 1, 2 and 3 above us can complete our model.

4. Conclusion and Future Enhancements

Machine Learning field has different facets suitable to select process in question. Our model has developed to include uncertainty in the decision of selection and collection of data sets to form a definite course. Exploring the foundation from operational research in the decision rules and uncertainty suggested by the four components like 1. Laplace, 2. Minimax, 3. Savage, 4. Hurwicz. We can modify the above approach to give a better machine learning process which will be of used the broader area of E-Learning.

REFERENCES

1. A Agrawal and D. Jaiswal, When machine learning meets ai and game theory, Stanford University, Machine Learning Final year Projects [pp 221 – 240, 1981]
2. D. E. Goldberg, *Genetic Algorithms for Search, Optimization and Machine Learning,* Reading, Addison-Wesley, Revised Edition, 2007.
3. H. A. Taha, *Operations Research: An Introduction*, 8th Edition, Pearson Education, 2008.
4. R. Axelrod, The evolution of strategies in the iterated prisoner's dilemma, in genetic algorithms and simulated annealing, L. Davis, Ed. Los Altos, CA: Morgan Kaufmann, 1987.
5. J.H.Holland, A mathematical framework for studying learning classifier systems. In D. Farmer, A. lapedes, N. Packard and B. Wendroff (Eds.), Evolution, games and learning (pp. 307 -317), Amsterdam, North-Holland, (Reprited from Physica, 22D, 307 -317), 1986.
6. S. Mittal and K. Deb, Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives, *IEEE Transaction on Evolutionary Computation,* 13(3) (2009) 554 – 565.