UNIVERSITY OF MANCHESTER

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

POSTGRADUATE DIPLOMA PROJECT:

SIMULATION OF AGENT LEARNING WOLF ALGORITHM

SUBMITTED BY:TIMUR JUMANAZAROV

SUPERVISOR: MARTIN BROWN

ABSTRACT

The aim of this work is to develop a software, which will help simulate an environment of machine learning to analyze the WOLF algorithm. The WOLF stands for "Win or learn fast" algorithm, which is mainly based on principle of variable learning rate. The implementation of algorithm was realized on MatLAB software, which is a very powerful tool in iterative processes. Multiagent learning algorithm is used to present best decision making strategy which will meet minimal requirements. Since the optimal strategy is dependent on the play of agents opponent the and they follow the same algorithm, there is big need in simulation of off-line games. Also provided some conclusions of how well desired properties are reached.

**Table of Contents**

**Chapter 1: Introduction**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

**Chapter 2: Game theory** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.1 Type of games . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.2 Strategies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2.3Nash and convergence. . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**Chapter 3: Multiagent learning**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

**Chapter 4: Gradient Ascent**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

4.1 Iterative games. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

**Chapter 5: WOLF**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

5.1 Win or learn fast principle . . . . . . . . . . . . . . . . . . . . . . . . 13

5.2 Analysis of WOLF . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

5.3 Empirical results. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

**Chapter 6: Conclusion** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

**List of references** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

INTRODUCTION

A Modern world brought a very wide area of new challenges. So most of nowadays problems are been solved by means of new technology. So there is big dependency on highly intelligent machines, which will able to solve problems a human being can not. A type of scientific discipline that deals with adapting computers to be able to carry out that kind of problem domains is "Machine learning", which is part of Artificial Intelligence. It understood that presents a system which develops in some environment and it has sensors that helps to get data from outside an analyze it. A unit which learn in environment collects data and makes decisions is called an agent. Machine learning has a situation of agent learning, and when in same environment n-number of agent can cooperate is called as multi-agent learning. Multiagent for the present period of time has wide area of problems to solve.

In 2nd chapter is introduced a game theory, whose problem domains brought new challenges. So to solve these problems a machine learning is developed.

In 3rd chapter description of this branch is given. Some requirements of agent learning are explained, which are directed to find and develop algorithms that will be rational and meet expectations. Rationality is very crucial aspect in agent learning processes. Simply it may be understood that an agent would be clever enough to make the best decision in particular situation.

4th chapter is partially a review of previous work in gradient based algorithm. In this chapter an analysis of the algorithm is held and some outcomes are made.

Further in 5th chapter is a WOLF algorithm is presented. So researcher is focused on simulation of an environment where the agent can learn and coexist at the same time with other agents.

2 GAME THEORY

Game theory came from economics research to understand how different subject or object can cooperate. One of the main ideas of the game theory is to maximize the reward of agent or player. So there comes a very important property of nay game is how rational an agent can be in choosing an action. It means the agent has to choose the best action according to the choice of action is presented in particular game. Simultaneously his opponent can be also rational one , and will be choosing action according to his rationality. As it was mentioned if the aim is to maximize expected reward there comes an issue of adapting to other players strategy. Roughly speaking it considers that an agent has to exploit other players strategy while giving minimal chances to opponent to be recognize his own strategy. And a player can change a choice of selected action in order to make sure that his strategy is right according to his condition. In this paper we will mainly focus on 2 agent learning games. There are many different type of games which are used as an example to show the main principles of game theory. The author chooses two games which are "Battle of sexes" and penalty kicking.

2.1 Types of games

Every game representation has a form to understand the form of a game. A very general form is Normal Game. So the structure of game is defined as:

N – is denoted as number of players involved in game, and every player is marked as

P={P1, P2, ...PN}. Minimum realization of multi-agent learning is 2 players, and called as Bimatrix games. N=2 is very popular because it carries out examples of two competing sides such as companies producing same product

A- is showing a set of action available for i- player

R- and playoff junction which shows the rewards according to selected action

Games can be presented in form of matrices, which will help to analyze them easier. General sum games are called strictly collaborative. Also games can be zero-sum games which sometimes called as strictly competitive games. Crucial aspect of zero- sum is that the total reward an agent can gain is fully equal to the total loss of 2nd player. Matching pennies can be considered as an example, because whenever a person guesses a side of coin he gets unit of money, so as his opponent looses exactly same amount. So it rises one of the main properties of multi-agent learning is *rationality.* It means, when agent is learning in particular environment while choosing any action in a set he has to take into account that the other players strategy.

Rationality itself is closely connected with so called Dominant strategy, which is the helpful in finding optimal strategy. To understand this concept we can take a closer look at an example. Consider we have a 2x2 playoff matrix

(2.1)

2.2 Strategies

Strategies are basic concept of every game and choosing the best one is the goal that have to be achieved, machine learning is the branch that focused on finding solutions for decision making. Simply we can say that a player has some strategies to play which could be best or worse. And in every chosen strategy must be a set of action available. In this thesis examples are in format of 2 by 2 matrix which means, they have couple strategies and in every strategy 2-3 available actions. So the main question is how to recognize the strategy that will be the best. In general form in agent learning presents an identification of dominant strategy.

Dominant strategy is the strategy that is best in the choice of strategies. A player chooses best one by comparing playoffs. Technically speaking option of choosing one of the elements is playoff matrices arises two concept of dominance. One of them is *strict* dominancy, when choosing one has better outcome than other. And *weak* dominancy is when one agent's playoff matrices element are not fully dominant to others. According to the probabilistic environment of most games selecting an action are divided in *pure strategy* and *mixed strategy*. An example of "penalty kicking", which was chosen as a good demonstrative game uses mixed strategy because the selection of action occur under probabilistic distribution. In that example a player has 50/50 shot deciding to kick right or left side.

2.3 Nash and convergence

It happens that there exist not only one dominant strategy or both agents has strictly dominant strategies but in case when they play against each other these strategies will not be best for every game played iteratively. So agents meets the situation called Nash equilibrium presented in 50s (Nash 1950).

Definition of Nash equilibrium says that no player can do better by maximizing playoff dependent on his own strategy. Generally it means that if the agent chooses a playoff is same with the other agent would chose in terms of your decision. And on the other hand the 1st agents choice will be exact as 2nd players best choice. So if in two presented cases the joint playoff will be the same it means they are in Nash equilibria. In Matlab realization it can be done by comparing the maximum element. Then as biggest elements found and for both matrices they are the same it means the exists Nash.

%playoff matrices

>> A=[0 3; 1 2];

%identify best response

>>A1=A(:,1);

>>A2=A(:,2);

>>[a11 a12]=max(A1);

>>[a21 a22]=max(A2);

In simulation process of gradient based algorithm it is important to find out equilibrium in order to analyze the convergent point. So as it was mentioned before one more crucial aspect of agent learning will be reached. Because if both players use one algorithm and learning process flows in same way equilibrium will be reached.

3 MULTIAGENT LEARNING

As it will be considered below a many examples of game theory are been solved when strategy is selected several times, which means iterative play. The aim of repetitive play is same as game theory and pursues the aim of maximizing the playoff reward in every game. Agents that learn and cooperate in same environment called as Multi-agent system[x2]. Key point in this branch of Artificial Intelligence is how to make an agent that is learning in some environment to behave rationally. Simply a performance of any algorithm can be seen by comparing to the goals that must be achieved. This branch became popular in 90's while scientist where dealing with robotics(x4). As a result it became a crossing area between machine learning and multi-agent systems.

multi-agent learning

multi-agent systems

machine learning

fig3.1. Multi-agent learning system

Usually learning process for agent goes on in 2 regimes. One is online learning for instance modern neural networks. And the other is off-line learning algorithms and considered to be applied in a particular period of time.

As a part of simulation process we will consider that the agents are learning offline. The study of agent learning brought researchers (Conitzer and Sandholm,2003) to couple least desirable properties which are:

* learning against other player when he is stationary to converge to best response by playing best response strategy
* when players are learning using one algorithm to converge to Nash equilibrium.

These properties are useful when the agent doesn't know very well which strategy the opponent going to play. Therefore it can be concluded that if both players use best response algorithm (Alonso and Kudenko,1999), they will anyway converge to some point which is mostly Nash. Previously in multi-agent learning some algorithms were introduced, such as Q-learning, Awesome, opponent modelling and gradient based algorithms. These algorithms mainly have same purposes and been created recently.

This project will deal with one that is gradient based and introduces a new concept in scalable learning rate. The difference is in so called variable stepsizes that allows to a player change strategy according to how well the current strategy is doing. In next chapters when algorithm will be shown we will deal with several well known examples in machine learning and try to analyze the dynamics of strategy update, and compare with previous ones.

The realization of algorithm in software is divided into 2 main parts:

1. is to understand which of the present tools could be used to model multi-agent learning. Matlab is very powerful that is able to realize simple operations on optimization, probabilistic issues and decision-making part.
2. design of algorithm that is game solver

So as an output we can say that the software must be capable in supporting agent to choose best response. For instance if we have a case as in famous game "battle of sexes"(Osborne 1994), and a husband as a first player must choose the solution by choosing whether going to opera or football. Additionally we can say that the algorithm must be flexible in case of appearance of new information of other agent and act according to the present best response.

4 Gradient Ascent

As it has been mentioned above multi-agent learning algorithm may use gradient in order to reach expected playoff (Singh, Kerns and Mansour, 2000). Better it can be considered on repetitive form games.

On the example of bimatrix games further will be shown how the dynamics of agent learning change over iterations.

Firstly let's take an example of penalty kicking. We will have two playoff matrices which will represent their actions and rewards they get according to the selection. A footballer who will kick the penalty will be player number one and denoted as P1. And a goalkeeper as P2. Below a is given representation of the game in matrices.

In this simple game a field player kicks whether right or left, and the aim of the goalkeeper is to catch a ball by jumping to one of the sides. One crucial aspect in this game is every round players doesn't know which action other one going to choose. So the learning is mainly based on the previous games, so we can say that there is an update of strategy at each stage. In this example the reward for P2 when he misses a side to jump is given as 0. It could be given as -1, but as it has been decided the analysis of algorithm would be more satisfying if the game was given not as zero-sum.

The selection of action is distributed over probability [0;1] and denoted by alpha(α) and

betta (β) respectively. For the first player selecting any action has equal cases so α =0.5 and

1-α=0.5. Of course in reality a player if he is right handed, will kick more likely to left side, but we can skip it as a fact that he uses both legs very well. On the other side strategy for P2 is similar that is why β=0.5 and 1-β =0.5.

So afterward the expected playoff is calculated:

(2.1)

where and are the selected available actions from matrix for both players.

The Matlab software allows easily calculate these playoffs which been found as -2 and 2.

The joint function for both strategies lies inside unit square area. So if strategy will move out of constrained space the gradient will project it back to probability space.

2.1 Iterative games

We now will apply gradient based method on repetitive games and the goal is to maximize reward by updating the current strategy by taking into account previous strategy of player and his opponent. In literature it was found that Singh (2000) introduced so called IGA(Infinitesimal Gradient Ascent) algorithm. The idea is to decay the strategy at each step with some step size which will decay the strategy over time.

So the update will look like:

(2.2)

-2

2

In Matlab was created a programme which allows to simulate the update process and analyze the dynamics of changes. In this algorithm a partial derivative was used as an update process, so it can be concluded that there must be a point with no gradient at all.

We can rewrite this equation in format of discrete linear state space form which will be more useful in analysis.

(2.3)

So as an output of this simulation we have a graph of strategy update at every game played by players. As it can be seen from the figure below there are two strategies for P1 and P2. They start learning with some step size.

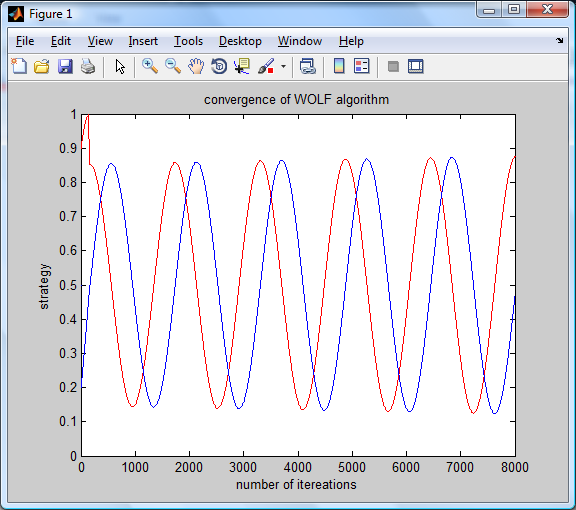


fig 4.1 – Gradient Ascent algorithm

So as it can be seen from the graph we can notice that this algorithm fails to converge. The speed of change is dependent on the stepsize, so if it is chose big the dynamics will vary very fast. And on the contrary the smaller stepsize the slower convergence. Figure 4.1 shows that it has 8000 updates and the learning rate is chosen as 0.002. And the initial conditions are alpha=0.9 and betta=0.2

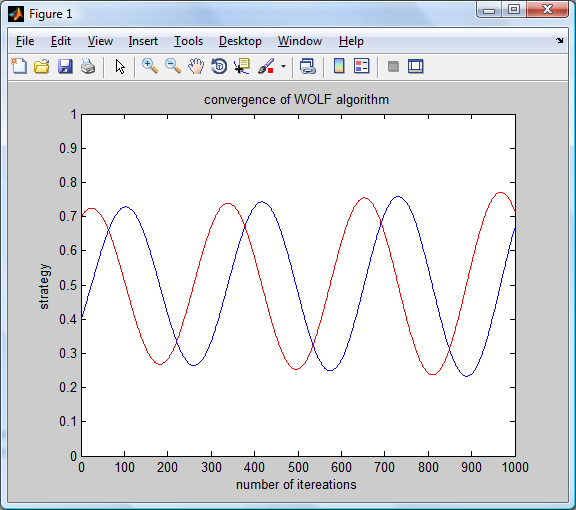


fig 4.2. Gradient Ascent with smaller learning rate

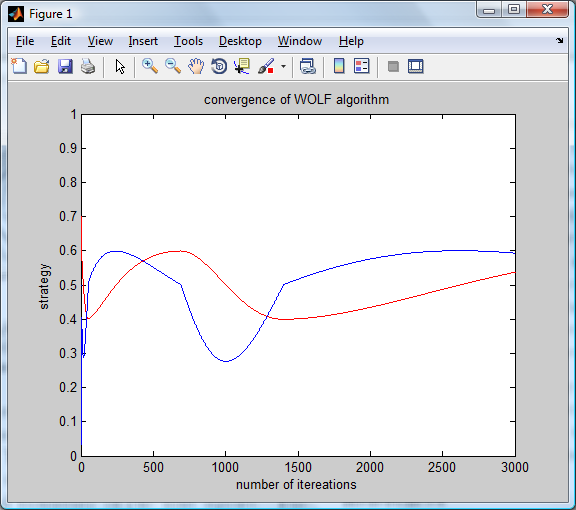


fig 4.3 Penalty Kick game

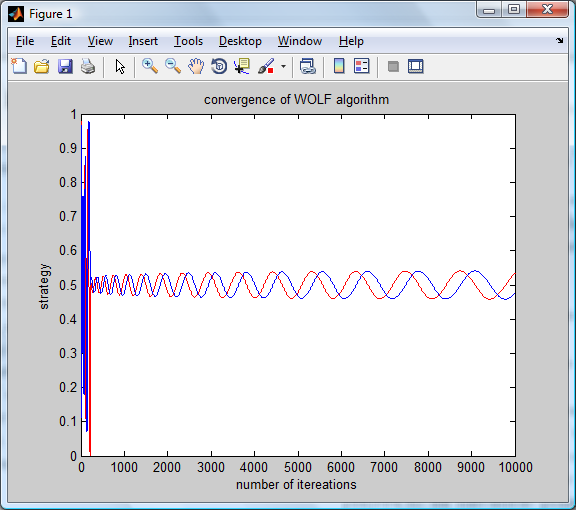


fig 4.4 penalty kick in form of zero-sum game

Above are given 3 cases

1. when the game is strictly collaborative, where the playoff has same structure as in example of "penalty kicking". (figure 4.3) graph has bigger learning rate so it affect the speed of strategy update. But the problem is that even if you vary gradient it fails to converge.
2. is case when the playoff matrix is presented in form zero-sum game. As it can be seen from figure 4.3 the dynamics of strategy change is very slow, and 1000 iterations are not enough to see the whole process.
3. the case when learning rate is decaying over the time and chosen as . The number of iterations are 10000, simply to show that by the time the update process becomes slower.

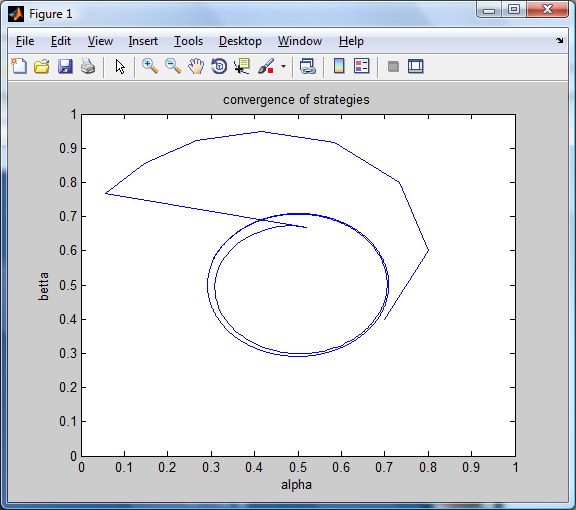


fig 4.4. the dynamics of both strategies over 5000 iterations.

In the figure above shown a case when the learning rate is decaying over each iteration. And it has some limit cycle where both *alpha* and *betta* oscillate around. As it can be concluded IGA is not convergent at each game and needs some regulations. So issue of modifying present algorithm has arised. In the next chapter new concept will be introduced.

5 WOLF

Bowling and Velosa (2003) studied previous algorithm and introduced new concept of variable learning rate. And simply called it as WOLF which stands for "Win or learn fast". The idea is to maximize the playoff but additionally it takes into account how well the agent is doing. So they proposed that there must be couple stepsizes that will vary according to the present conditions of the player. As it has been discussed previously the requirements for multi-agent learning algorithm is to be rational and converge to some point, which is mostly considered as Nash equilibrium.

5.1 Win or learn fast principle

The main idea of new algorithm is to look at your opponent strategy and behave according to your present conditions. Simply it means an agent will compare his current strategy with one that is in equilibrium. So we have two conditions according to comparison of conditions.

1st is when agent is doing better than expected to be in equilibrium. Graphically it may means that the strategy is moving away from the centre. But it arises a question of the P2 to be adapting to current strategy, it means by the time for instance our goalkeeper will learn that the kicker is always choosing to kick the right side and will start jumping and saving the gate. And in general formulation we can say that the P1 has to slow down his learning rate.

2nd is when the strategy that a player chose is not giving good result, and is playing worse than expected. In this situation it is obvious that the learning rate must be changed to bigger one in order to eliminate a lag.

(5.1)

And now after the principle of variable learning rate was cleared it can be used in update process for each player. As an analogy in IGA we have same update parameters, except that now exists extra learning rate. Simply the algorithm is as presented by 5.2

(5.2)

where both scalable learning rates must be bigger than zero.

5.2 Analysis of WolF

To analyse the new gradient based algorithm we can rewrite (5.2) into state space form

(5.3)

so have a matrix, let denote is as a W

(5.4)

we can calculate eigenvalues, by

(5.5)

calculations will show that system has imaginary eigenvalues, and now we have 2 cases to analyze:

1st when learning rates are not constant what will be the dynamics of strategy change.

2nd is when update is supported by constant stepsizes.

Obviously the dynamics will be different. In lemma presented in work of Michael Bowling(2003) was given that a player is winning if he is moving away from the centre. But the condition of using both agents same algorithm will bring to the situation when no one is winning or losing. As a define of winning it must be concluded according to equilibrium which is said to be expected playoff a difference between two strategies must be greater than zero.

5.3 Empirical results

Now we some results will be presented. Below given a game, which have small learning rate. Key factor of simulations that is presented is that it that even an algorithm is considered to be rational and in fact must by the time converge to some point it hasn't reached it. As we can see there are 10000 iterations which are not enough to realize the whole dynamics. So next, some changes should be applied to matlab code.

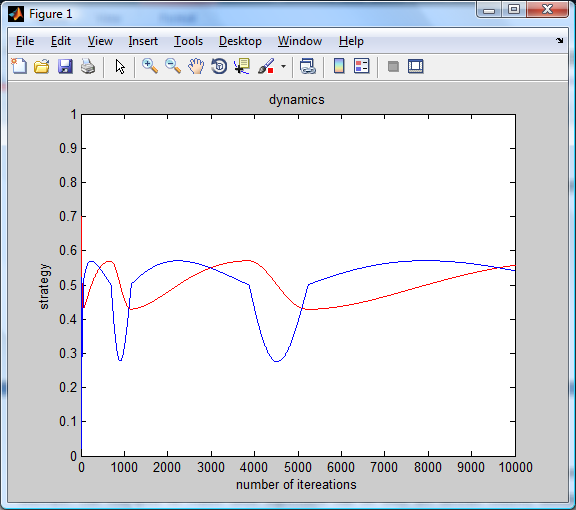


fig 5.1 Simulation of game using WOLF

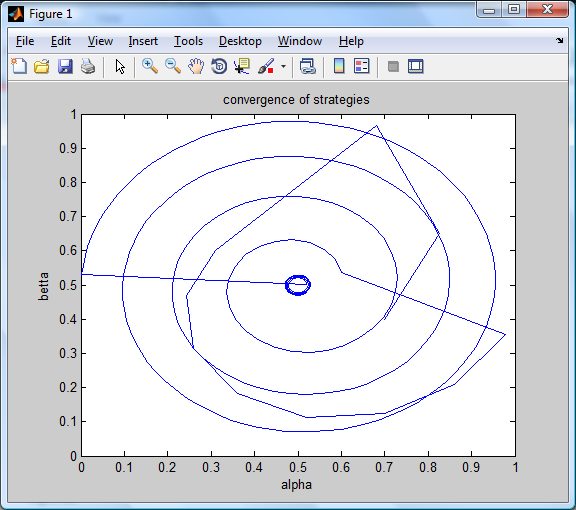


fig 5.2 Penalty kicking game using WOLF algorithm

One of the lemmas that was stated in feasibility study work says that in case when a W matrix has imaginary eigenvalues, a strategies form an elliptical orbit. In the figure(5.2) above is given an example of penalty kicking game. And it is noticeable that it forms ellipse at the first iterations of strategy update. Additionally strategies are moving towards centre which is inside unit square and is Nash itself.

To show the differences in dynamics even in one game one more simulation of P1 and P2 player is provided. The difference with the output above is that now decaying variable learning rate is used. Conditions are the same, but more iterations are calculated. The equilibrium point is found to be at point 0.5 and 0.5. So by looking at the figure it can be concluded that over 7000 iterations the adjoint strategy is very close to converge to nash equilibrium.

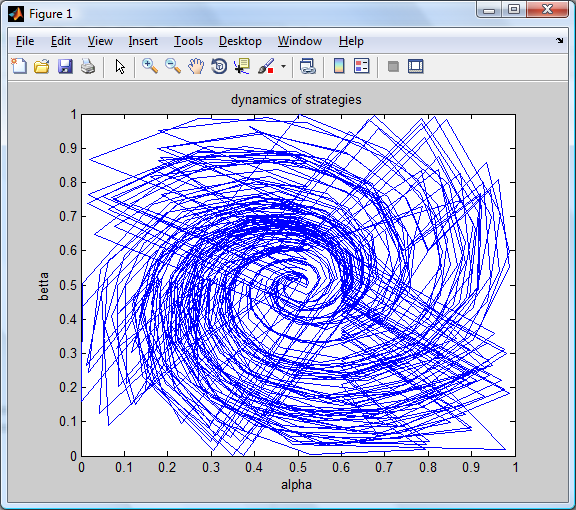


fig 5.3 Penalty kicking game with decaying learning rate

Also the game of "battle of sexes" is simulated. The playoff matrices are given as:

A= [ 0 3; 1 2 ];

B= [3 2 ; 0 1];

learning rate for winning and losing are nn=(k^(-1/5));nn2=(k^(-2/3)); respectively.

It can be clearly seen that at the end algorithm reaches the centre of (0.5,0.5) which is equilibrium point and reaches zero gradient conditions

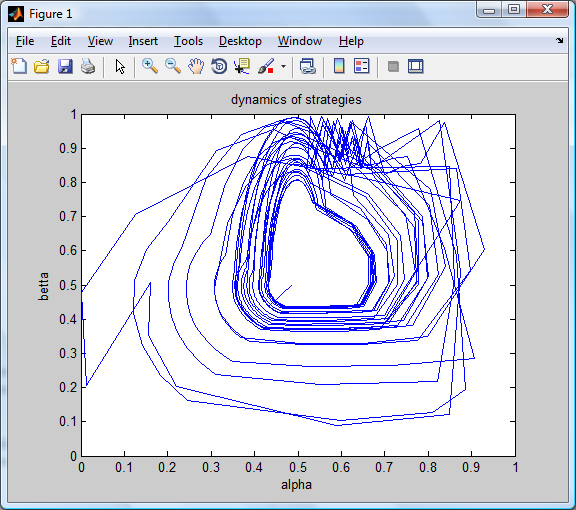


fig 5.4 Battle of sexes

Zero gradient point is considered to be coordinates on phase plane where a previous strategy is same as current strategy. For iterative algorithm we can say that it is a stop function. By simplifying the partial derivative and reforming it we have:

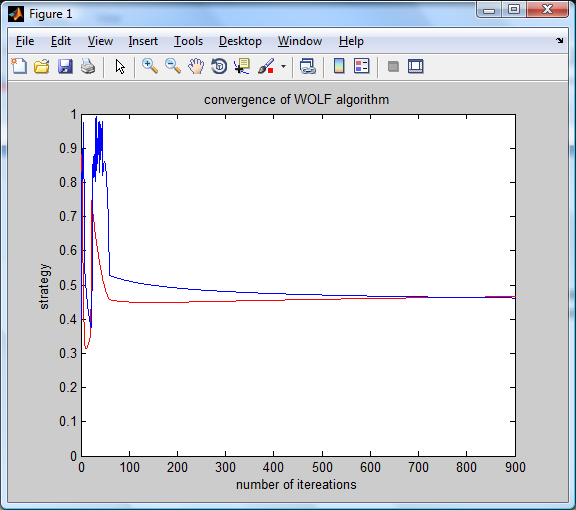


fig 5.5 An example of convergence of WOLF algorithm

In figure 5.5 can be clearly seen that the convergence of algorithm is reached very soon according to previous simulations and reaches Nash equilibrium at ~ 800 iteration. Obviously this example represents that the Win or learn principle is effective in reaching desired expectations.

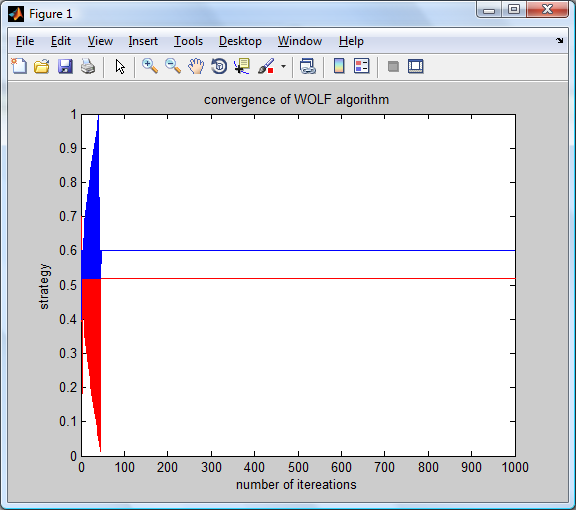


fig. 5.6 Zero gradient case

This is the last example fast convergence. After all simulations we can say that the response of WOLF algorithm depends on which type of game is been played, and choosing proper variable learning rates.

6 CONCLUSION

This thesis was dedicated to show how well the WOLF algorithm works. In this paper was presented results of simulation of couple games. The simulation process was accomplished in MAtlab software and seems helpful in understanding of dynamics of strategy update.

As a conclusion we can say that this algorithm met minimal requirements and the rationality of WOLF was shown. Also the simulations allowed observing the process of convergence and realizing dynamical behaviours of strategies of both agents. Of course the 2x2 matrix games are the minimal realization of general games, but it showed general concept of agent learning especially in case of WOLF algorithm.

List of references

E. Alonso, and D. Kudenko, 1999. Machine learning techniques for adaptive logic-based multi-agentsystems. *Proceedings of the Second Workshop of the UK Special Interest Group on Multi-Agent* *Systems (UKMAS-99)*, Hewlett-Packard Laboratories, Bristol.

Bowling M. and [Veloso](http://www.cs.cmu.edu/%7Emmv) M. (2001). Convergence of Gradient Dynamics with a Variable Learning Rate. In *Proceedings of ICML-2001*, pp. 27–34, Williams College, MA,

Bowling, M., and Veloso, M. (2002). Rational and convergent learning in stochastic games. In *Proceedings of* *the Seventeenth International Joint Conference on Artificial* *Intelligence,* 1021-1026

Bowling M. and [Veloso](http://www.cs.cmu.edu/%7Emmv) M. (2003). Multiagent Learning Using a Variable Learning Rate. *Artificial Intelligence*, 136:215–250

V. Conitzer and T. Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents, 2003.

J. F. Nash, Jr., Equilibrium points in n-person games, PNAS 36 (1950) 48-49, reprinted in (36)

Osborne, Rubinstein (1994). A course in game theory. *The MIT Press.*

Sahota, M. K., Mackworth, A. K., Barman, R. A., Kingdon, S. J. . Real-time control of soccer-playing robots using off-board vision: the dynamite testbed*\ In IEEE International Conference on Systems, Man, and Cybernetics, 3690-3663.*

S. Singh, J. Kearns, Y. Mansour, Nash convergence of gradient dynamics in general-sum games, in: *Proceedings of the Sixteenth Conference on Uncertainty in Articial Intelligence, Morgan Kaufman, 2000, pp. 541-548.*

M. Wooldridge. Multiagent Systems: A Modern Approach to Distributed Artificial

Intelligence. The MIT Press, Cambridge, Massachusetts, 1999.