FNSPE CTU

Dynamic Decision Making

Seminar Paper

Minority Game
From the Dynamic Decision Making Perspective

Author
Vladislav Belov

April 16, 2018

1 Introduction

In the following seminar paper the Minority Game is going to be discussed. In the next section we will define the general mathematical model of the Minority Game. Afterwards, agent Q-learning and Roth-Erev learning will be introduced. Within the scope of this work the implementation was performed and its results will be presented in some section.

The system the Minority Game describes originates from the El Farol Bar Problem which was introduced by the economist W.B. Arthur [ref]. It goes as follows: every Thursday the population of Santa Fe has a desire to visit the bar: if more of 60 % of people come to the bar, then it is considered to be overcrowded, so it is no fun there; if less, then the visit is pleasant and those who stayed at home are at a loss. Therefore, it can be easily seen, that the minority wins in games of El Farol Bar Problem type, that is the reason why the model has such a name. Nowadays this model is used frequently in Finance, Network Analysis, Biology, etc.

2 Mathematical Model of the Minority Game

Firstly, consider an odd $N = 2k - 1$, $k = 1, 2, \dots$, number of agents participating in the game. At each time step $t = 1, 2, \dots$ each agent has to make a decision whether to perform an action $+1$ (e.g. go to the bar, sell an asset on the market) or -1 (e.g. stay at home, buy an asset on the market). Formally designated $\forall i \in \{1, 2, \dots, N\}$:

$$a_i(t) = \pm 1 \tag{1}$$

In order to study game dynamics a special parameter called *total action* was introduced:

$$A(t) = \sum_{i=1}^{N} a_i(t), \forall t \in \{1, 2, \dots\}, \tag{2}$$

which is basically the sum of actions performed by every agent at a given game round.

After each game round the outcome is disclosed to each of the agents: as soon as the round is finished, everyone gets to know what the winning action was. This action $W(t+1)$ ($t+1$ is used to specify, that this information is available at a time step $t+1$) is determined by a simple rule:

- $A(t) > 0 \implies$ the action -1 was victorious, $W(t+1) = -1$;
- $A(t) < 0 \implies$ the action $+1$ was victorious, $W(t+1) = 1$;
- $A(t) = 0$ will never occur due to oddness of N.

In other words, $W(t+1) = -\text{sign}\big(A(t)\big)$.

Minority Game is a system of agents with bounded memory: each agent remembers $m \in \{1, 2, \dots\}$ recent game outcomes. In accordance with its memory. This *recent history* can be represented two different ways:

- As an m-tuple of most recent game outcomes: $\bar{\mu}(t) = \big(W(t-m+1), W(t-m+2), \dots, W(t)\big)$;
- As a single decimal number $\mu(t)$ with binary representation equal to $\bar{\mu}(t)$.[1]

[1] E.g.: let $m = 3$, then for $\bar{\mu}(t) = (-1, -1, 1)$ its decimal representation (obtained by applying the transformation $-1 \to 0$) is $\mu(t) = 1$, because $001_2 = 1_{10}$.

Recent History, $\bar{\mu}(t)$	$\mu(t)$	Action/Decision
000	0	-1
001	1	$+1$
010	2	$+1$
011	3	-1
100	4	$+1$
101	5	-1
110	6	$+1$
111	7	$+1$

Table 1: Strategy example for $m = 3$.

It can be easily seen, that in total 2^m possible recent histories exist, therefore, $\mu(t) \in \{1, 2, \ldots, 2^m - 1\}$.

At the beginning of the game each agent gets a fixed number of strategies. For any fixed m a *strategy* is a mapping $\pi : (0, 1, \ldots, 2^m - 1) \mapsto \{-1, 1\}^{2^m}$ (all possible outcomes are mapped to respective actions, e.g. see Table 1). The construction of such mapping implies, that agents are able to get from 0 to 2^{2^m} strategies. The set of strategies available to agent $i \in \{1, 2, \ldots, N\}$ will be called agent's strategic portfolio and denoted as $\mathcal{P}_i$. The action/decision of agent i with respect to recent history $\mu(t)$ using the strategy π will be denoted as $\pi_i(\mu(t))$, e.g. if agent i uses the strategy from Table 1 when recent history is 101, then $\pi_i(5) = -1$.

Not the question arises: how to model the process of decision making? In the following section we will discuss the original learning mechanism and some other.

3 Learning in the Minority Game

To begin, we will try to describe the Minority Game in a more general way. It is a complex system where agents need memory to make an optimal decision and not all relevant portions of the environment can be observed - we are dealing with partially observable, stochastic, sequential, dynamic, discrete, multi-agent environment. Depending on the learning mechanism, the set of states can be defined differently. Here basic ideas of the MG representation as MDP will be highlighted and some other SHIT DONE.

Let $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ be a Markov Decision Process representing the Minority Game. Obviously, both the transition model $\mathcal{T}$ and the reward model $\mathcal{R}$ are not available, so the agents have to learn "along the way".

In this paper we want to focus on comparison of different learning mechanism and what is their impact on the performance of a single agent. Let agent i has won $w_i(t)$ and lost $l_i(t)$, $w_i(t) + l_i(t) = t$, then its performance at time $t$is defined as:

$$p_i(t) = w_i(t) - l_i(t). \tag{3}$$

3.1 Original Learning Mechanism

The original learning mechanism is based on giving strategies scores which will be updated as the game progresses, in other words, if the strategy was successful during the recent round, then its score is increased, and vice versa. The update rule of the score for strategy π denoted as R^π is defined as follows:

$$R^\pi(t+1) = R^\pi(t) - \delta_{\mu(t),j} \cdot \pi(\mu(t)) \cdot g\big(A(t)\big) \tag{4}$$

where δ stands for Kronecker delta and g is any odd function with respect to total action $A(t)$.[2]

At each game round agents choose the best scoring strategy from the ones available for them:

$$\pi_i^{\text{opt}} = \arg\max_{\pi \in \mathcal{P}_i} R(t). \tag{5}$$

The strategy score can be in some sense interpreted as the reward function, but such an assumption is a bit misleading.

3.2 Q-learning Mechanism

The Q-learning approach is based on the update of Q-function which is an expected total reward from taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$. Its update rule is defined as:

$$Q_i(s,a) \leftarrow (1-\alpha) \cdot Q_i(s,a) + \alpha \cdot \Big(r(s) + \gamma \cdot \max_{a'} Q_i(s',a') \Big) \tag{6}$$

where s' is the next state, α is a learning parameter, γ is a discount factor and r is an immediate reward at state s.

Let $\mathcal{S} = \{\pi : \pi \text{ is a strategy}\}$, the set of possible actions will be defined as $\mathcal{A} = \{\text{switch from using strategy } \pi \text{ to } \tilde{\pi} : \pi, \tilde{\pi} \text{ are strategies }\}$. Now Q-learning may be applied on the minority game. The action-state function denoted as $Q_{i,\pi,\tilde{\pi}}$ has the following meaning:

- If $\pi = \tilde{\pi}$, then the agent will not change his current strategy to another one;
- If $\pi \neq \tilde{\pi}$, then the agent will switch from strategy π to $\tilde{\pi}$.

Agent $i \in \{1, 2, \dots, N\}$ will choose its optimal strategy at each time step by the ϵ-greedy rule[3]

$$\pi_i^{\text{opt}} = \arg\max_{\tilde{\pi} \in \mathcal{P}_i} Q_{i,\pi,\tilde{\pi}}. \tag{7}$$

Action-state function update rule which has to be evaluated at each game round will take form:

$$Q_{i,\pi,\tilde{\pi}} \leftarrow (1-\alpha) \cdot Q_{i,\pi,\tilde{\pi}} + \alpha \cdot \Big(r(\pi) + \gamma \cdot \max_{\pi'} Q_{i,\pi,\pi'} \Big) \tag{8}$$

where after using the strategy π agent's immediate reward is $r(\pi) = -\pi_i(\mu(t)) \cdot A(t)$. That means, in case π was successful it will be rewarded by $|A(t)|$ and $-|A(t)|$ in the opposite case.

3.3 Roth-Erev Learning Mechanism

Roth-Erev reinforcement learning is completely different from the techniques discussed in subsections 3.1 and 3.2. Here agents are not strategies according to the original model, decision making is based on *propensities* and their update rule. Although the state space $\mathcal{S}$ and the set of possible actions $\mathcal{A}$ are the same as in the subsection 3.1, the learning mechanism is as follows [**?**]:

- Agent $i \in \{1, 2, \dots, N\}$ has a propensity $q_i^a(t)$ to play action a, $\forall a \in \mathcal{A}$. Propensities are required to meet the condition $q > 0$ and they have to stay positive[4];

[2] Here for simplicity $g(\cdot) = \text{sign}(\cdot)$.

[3] Strategy π_i^{opt} will be chosen with probability $1 - \epsilon$.

[4] That can be achieved by defining a positive payoff function.

- Let $(y_i(t), 1 - y_i(t))$ represent agent's strategy at time t which means, that he will take action -1 with probability $y_i(t)$, hence, the action $+1$ will be taken with probability $1 - y_i(t)$. These probabilities are defined by a simple relation:

$$y_i(t) = Pr(a = +1) = \frac{q_i^{+1}(t)}{q_i^{+1}(t) + q_i^{-1}(t)}. \tag{9}$$

  It is simple to deduce, that $Pr(a = -1) = 1 - y_i(t)$.

- Propensity is updated only if the taken action was successful[5]:

$$q_i^a(t+1) \leftarrow q_i^a(t) + \frac{|A(t)|}{N}. \tag{10}$$

4 Simulations

[5] In the original text [?] the update is given as some general function. The following update was proposed to perform simulations.

References