Pattern Analysis in Social Networks with Dynamic Connections

Yu Wu[1] and Yu Zhang[2]

[1] Department of Computer Science
Standford University
ywu2@standford.edu
[2] Department of Computer Science
Trinity University
yzhang@trinity.edu

Abstract. In this paper, we explore how decentralized local interactions of autonomous agents in a network relate to collective behaviors. Most existing work in this area models social network in which agent relations are fixed; instead, we focus on dynamic social networks where agents can rationally adjust their neighborhoods based on their individual interests. We propose a new connection evaluation rule called the Highest Weighted Reward (HWR) rule, with which agents dynamically choose their neighbors in order to maximize their own utilities based on the rewards from previous interactions. Our experiments show that in the 2-action pure coordination game, our system will stabilize to a clustering state where all relationships in the network are rewarded with the optimal payoff. Our experiments also reveal additional interesting patterns in the network.

Keywords: Social Network, Dynamic Network, Pattern Analysis.

1 Introduction

Along with the advancement of modern technology, computer simulation is becoming a more and more important tool in today's researches. The simulation of large scale experiments which originally may take people months or even years can now be run within minutes. Computer simulation has not only saved researchers a great amount of time but also enabled them to study many macro topics that were impossible to study experimentally in the past. In this situation, Multi-Agent System (MAS) has emerged as a new area that facilitates the study of large scale social networks.

Researchers have invented many classic networks, such as random networks, scale-free networks (Albert and Barbási 2002), small world networks (Watts 1999), to list a few. Although these networks have successfully modeled many social structures, they all share the weakness of being static. In today's world, many important virtual networks, such as e-commerce networks, social network services, etc., have much less stable connections among agents and thus the network structures will constantly change. Undoubtedly, the classic networks will fail to capture the dynamics of these networks. Out of this concern, (Zhang and Leezer 2009) develops the network model HCR

J. Salerno et al. (Eds.): SBP 2011, LNCS 6589, pp. 163–171, 2011.

(Highest Cumulative Reward) that enables agents to update their connections through a rational strategy. In this way, any classic network can be easily transformed into a dynamic network through the HCR rule. In their model, agents evaluate their neighbors based on their performance history, and all interactions in history are equally weighted. However, people may argue that the evaluation function is not very realistic since in the real world recent events may have a greater influence on people than long past ones.

Motivated by this fact, we extend HCR to a new rule called HWR (Highest Weighted Reward). The difference is that HWR allows the agents to use a discount factor to devalue past interactions with their neighbors. The more recent the interaction is, the more heavily the corresponding reward is weighted in the evaluation function. In our research, we identified certain patterns from the simulation results and then demonstrated the existence of the pattern empirically.

2 Related Work

To our view, the social simulation conducted on MAS can go into two directions: One is to build a model as realistic as possible for a specific scenario, such as the moving crowds in an airport. The other one is to build an abstract model that captures the essence of human activities and can be used to model many different social networks of the same kind. Our work here belongs to the latter category.

Research in this area usually focuses on the study of social norms. One of the most basic questions that can be asked about the social norms is whether the agents in the social network will eventually agree on one solution. Will the whole network converge on one solution? The results vary drastically in different networks and under different algorithms. The agents can converge into various patterns and with different time period. Generally, there are two categories in this area studying the emergence of social norms. The first category studies social norms in static networks. The second studies the evolving or dynamic network.

In the first category of static network, one of the most significant findings was by (Shoham and Tennenholtz 1997). They proposed the HCR (Highest Current Reward) rule. In each timestep, an agent adopting HCR rule switches to a new action if and only if the total rewards obtained from that action in the last time step are greater than the rewards obtained from the currently chosen action in the same time period. The authors simulated various networks under different parameter configurations (memory update frequency, memory restart frequency, and both) to experiment on an agent's effectiveness in learning about their environment. They also theoretically proved that under HCR rule the network will always converge to one social norm.

Another important work done in the first category is the Generalized Simple Majority (GSM) rule proposed by (Delgado 2002). Agents obeying the GSM rule will change to an alternative strategy if they have observed more instances of it on other agents than their present action. This rule generalizes the simple majority rule. As the randomness $\beta \rightarrow \infty$, the change of state will occur as soon as more than half of the agents are playing a different action. In that case, GSM will behave exactly as the simple majority rule does.

The second category of research on agents' social behavior is the study of evolutionary networks or dynamic networks, such as (Borenstein 2003, Zimmermann 2005). Here researchers investigate the possible ways in which an agent population may converge onto a particular strategy. The strong assumptions usually make the experiments unrealistic. For example agents often have no control over their neighborhood. Also, agents do not employ rational selfish reward maximizing strategies but instead often imitate their neighbors. Lastly, agents are often able to see the actions and rewards of their neighbors, which is unrealistic in many social settings.

In the second category of the research, (Zhang and Leezer 2009) proposed the Highest Rewarding Neighborhood (HRN) rule. The HRN rule allows agents to learn from the environment and compete in networks. Unlike the agents in (Borenstein 2003), which can observe their neighbors' actions and imitate it, the HRN agents employ selfish reward maximizing decision making strategy and are able learn from the environment. Under the HRN rule, cooperative behavior emerges even though agents are selfish and attempt only to maximize their own utility. This arises because agents are able to break unrewarding relationships and therefore are able to maintain mutually beneficial relationships. This leads to a Pareto-optimum social convention, where all connections in the network are rewarding.

This paper proposes the HWR rule, which is extended from the HRN rule. In HRN rule, the agent values its neighbors based on the all rewards collected from them in the past, and all interactions are weighted equally throughout the agent's history. However, this will cause the agent to focus too much on the history of the neighbor and fail to respond promptly to the neighbor's latest action. Therefore, we introduce the HWR rule, which introduces a time discount factor that helps the agents to focus on the recent history.

3 Highest Weighted Reward (HWR) Neighbor Evaluation Rule

3.1 HWR Rule

The HWR rule is based on a very simple phenomenon we experience in our everyday lives: human beings tend to value recent events more than events which happened a long time ago. To capture this feature, we introduce a time discount factor, which is usually smaller than 1, in order to linearly devalue the rewards collected from past interactions. When time discount factor equals 1, the HWR rule will behave in the same way as the HRN rule does. Notice that both HWR and HRN are neighbor evaluation rules, which means the purpose of these rules is to help the agents value their connections with neighbors more wisely. In other words, these rules can only make decisions regarding connection choosing, but will not influence the agent's action choosing decisions.

According to the HWR rule, an agent will maintain a relationship if and only if the weighted average reward earned from that relationship is no less than a specified percentage of the weighted average reward earned from every relationship. Next we carefully go through one cycle of HWR evaluation step-by-step.

1. For each neighbor, we have a variable named *TotalReward* to store the weighted total reward from that neighbor. In each turn, we update the *TotalReward* through the following equation in which *c* represents the discount factor:

$$TotalReward = TotalReward \times C + RewardInThisTurn \tag{1}$$

2. In a similar way, we keep a variable named *GeneralTotalReward* to store the weighted total reward the agent got from all neighbors in the history. In each turn, we update the *GeneralTotalReward* through the following equation:

$$GeneralTotalReward = GeneralTotalReward \times C + TotalRewardInThisTurn / NumOfNeighbors \quad (2)$$

Here *TotalRewardInThisTurn* needs to be divided by the *NumOfNeighbors* in each turn in order to find the average reward per connection. Since we want to calculate the average reward of all interactions, the total reward in each round needs to be divided by the number of interactions in that turn.

3. When we start to choose non-rewarding neighbors, we first calculate the average reward for every agent. The *AvgReward* is calculated in the following way: suppose the agent has been playing with the neighbor for *n* turns, then:

$$AvgReward = \frac{TotalReward}{1 + c + c^2 + ... + c^{n-1}} = \frac{TotalReward}{\frac{1 - c^{n-1}}{1 - c}} \quad (3)$$

4. Calculate the average total reward. Similar to step 3:

$$AvgTotalReward = \frac{GeneralTotalReward}{1 + c + c^2 + ... + c^{n-1}} = \frac{GeneralTotalReward}{\frac{1 - c^{n-1}}{1 - c}} \quad (4)$$

5. Compare the ratio of *AvgReward/AvgTotalReward* with the *threshold*. If the former is greater than the later, then we keep the neighbor; if not, we regard that agent as a bad neighbor. In this way, the agent can evaluate its neighbor with an emphasis on recent history.

The above explanation has assumed an ideal environment in which agents have infinity amount of memory for ease of exposition. In reality and our experiments, agents have a limited amount of memory and can only remember the interactions of a certain number of turns.

3.2 Discount Factor Range

Generally speaking, the HWR rule can work with different games and in various networks. Since this rule just adds a dynamic factor into the network, it does not contradict other aspects of the network. However, in some special cases, a certain experiment setup may make some discount factor value meaningless. Assume we are playing a pure coordination game (Guillermo and Kuperman 2001) with payoff table (C: Cooperate; D: Defect):

Table 1. Two-Person Two-Action Pure Coordination Game

	C	D
C	1	-1
D	-1	1

Also assume the agents only want to keep the neighbors whose average reward is positive for them. Then in such a case, no matter how many rounds an agent has been playing with a neighbor, its most recent reward will dominate the whole reward sequence. To be more specific, if an agent receives a reward of 1 from a neighbor in the last interaction but has constantly received -1 in the previous n trials, the cumulative reward will be $1-c-c^2-\ldots-c^n$. If c is too low, the cumulative reward will still be bigger than 0. We can find the range of c through the following derivation:

$$1-c-c^2-\ldots-c^n > 0 \Rightarrow 1 > c+c^2+\ldots+c^n \Rightarrow 1 > c(1+c^n)/(1-c) \quad (5)$$

When $n\rightarrow\infty$, $c^n\rightarrow 0$, then

$$\Rightarrow 1 > c/(1-c) \Rightarrow c < 0.5 \quad (6)$$

Therefore, we usually limit c in range of 0.5 to 1.

In fact, for most cases c will take a value much higher than 0.5. Since the past rewards were devalued exponentially, a small c value will cause the past rewards to be practically ignored very soon. In our common experiment setup, we let the agent have a memory size of 30. In this case, if $c=0.5$, then we see the oldest action will be devalued by a factor of $0.5^{29}\approx 1.86E\text{-}9$, which makes the reward totally insignificant. In fact, even for a relatively high c value such as 0.9, the last factor will still be as small as $0.9^{29}\approx 0.047$. Therefore, just in case the past rewards will be devalued too heavily, we usually keep the c in the range of 0.8 to 1.

3.3 Pattern Predictions

Based on the HWR rule, we make some predictions about the outcome of the network pattern.

> **Argument:** *Given a pure coordination game, placing no constraints on the initial choices of action by all agents, and assuming that all agents employ the HWR rule, then the following holds:*
>
> - *For every $\varepsilon > 0$ there exists a bounded number M such that if the system runs for M iterations then the probability that a stable clustering state will be reached is greater than $1-\varepsilon$.*
> - *If a stable clustering state is reached then the all agents are guaranteed to receive optimal payoff from all connections.*

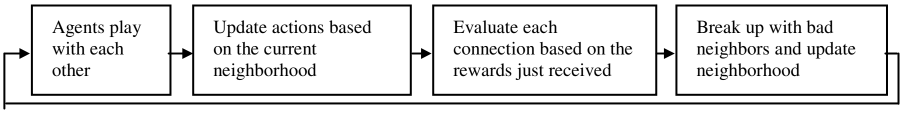
In order to further clarify the argument, the following definitions are given:

> **Stable Clustering State:** A network reaches a stable clustering state when all the agents in the network belong to one and only one closed cluster.
>
> **Closed Cluster:** A set of agents forms a closed cluster when there does not exist any connection between any agent in the set and another agent outside of the set, and also all the agents in the set are playing the same action.

Basically, what the argument claims is that the whole network will eventually converge into one single cluster of same action or two clusters with different actions, and once in such a state, the rewards collected from the network will be maximized.

4 Experiment

The operating procedure can be shown in the flow chart.



All experiments are conducted in an environment that has the following properties:

1. All trials are run in a random network having 1000 to 10000 agents.
2. Agents play with each other the pure coordination game defined in Table 1.
3. The number of connections in the network remains the same throughout the trial.
4. Every time a connection is broken, both agents have a 50% chance to gain the right to connect to a new neighbor. Exactly one of the two agents will find a new neighbor. This restriction guarantees the number of connections remains the same.
5. All agents have a limited memory size.
6. All agents adopt obey the simple majority rule, which means the agent will always choose the action that the majority of its neighbors used in the last turn. If there is the same number of neighbors adopting different actions, the agent will not change its current action.

4.1 Two-Cluster Stable Clustering State

First, we show the emergence of the stable clustering state. Since the final outcome can have two different patterns with either one or two final stable clusters, we present the patterns in two parts. First, let's check the situation where the network eventually converges into one single cluster. The following experiments are conducted with the parameters: *Number of Runs: 50, Time Steps: 30, Network Size: 5000, Neighborhood Size: 10, Reward Discount: 0.9, Threshold: 0.9.*

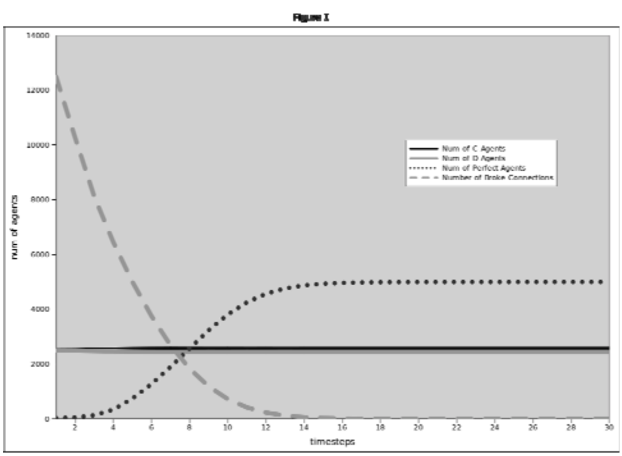


Fig. 1. Network Converges into Two Clusters

Fig. 1 shows the average results collected from 50 trials. We see that at the end of each trial, all agents have split into two camps and entered into a unique stable clustering state. First, we notice that the lines that represent the numbers of C and D agents almost merge together in the middle of the plot. Since the pay-off matrix for C and D actions is symmetric and there is no difference between these two actions except different symbolic names, the average numbers of C and D agents should be both around 2500, just as the plot shows. Here in order to capture the dynamics of the network, we introduce a new term named perfect agent. An agent will be counted as a perfect agent only when all of his neighbors plays the same action as he does. In Figure 1, while the numbers of C and D agents remain almost constant, the number of perfect agents rises steadily from 0 to 5000 throughout the trial. The increasing number of perfect agents shows the network is evolving from a chaotic state towards a stable clustering state. At the same time, we see the number of broken connections is steadily dropping towards 0. As there are more and more agents becoming perfect agents, connections between neighbors are broken less and less frequently.

4.2 One-Cluster Stable Clustering State

Now we present the experiment results where the agents all adopt the same action at last and merge into one cluster. In order to show the results more clearly, we have run 50 trials and selected 23 trials where C eventually becomes the dominant action. The experiments are carried out with the following parameters: *Number of Runs: 50, Time Steps: 30, Network Size: 5000, Neighborhood Size: 100, Reward Discount: 0.9, Threshold: 0.9.* In Fig. 2, we see number of C agents is steadily rising while number of D agents dropping until the whole network converges into a single cluster. The number of perfect agents and broken connections behaves in a manner similar to that in Fig. 1.

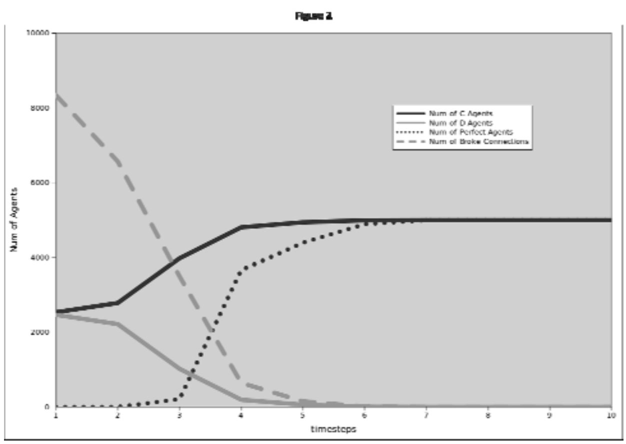


Fig. 2. Network Converges into One Single Cluster

4.3 Pattern Possibilities with Various Network Sizes

The above experiments demonstrate that the network sometimes converges into one single cluster and sometimes into two clusters. After a series of simulations, we found out that the size of neighborhood has a significant influence over the probability that a network will converge into one cluster or not. In order to show this, we created 16 networks with different sizes and ran 50 trials with each network. Here are the parameters: *Number of Runs: 50, Time Steps: 30, Network Size: 1000, Reward Discount: 0.9,* Threshold: 0.9. Fig. 3 shows the number of one-cluster trials and two-cluster trials with each network.

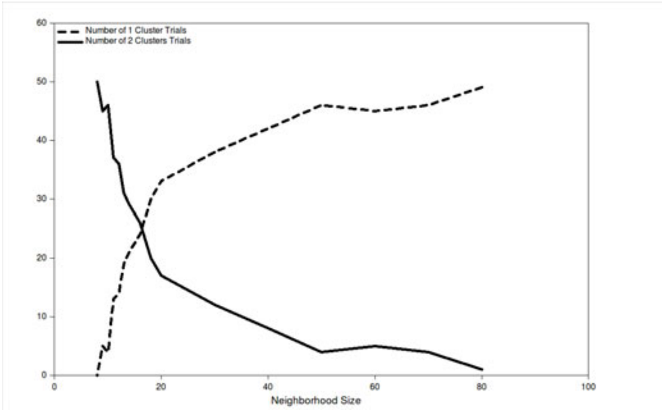


Fig. 3. Network Converges into One Single Cluster

5 Conclusion

The HWR rule enables agents to increase their rewards not only through switching actions but also through updating neighborhoods. As the experiment results show, agents adopting the HWR rule are able to change the network structure to maximize their own interests. Even though the agents behave selfishly, the network still reaches a Pareto-optimum social convention at last.

Acknowledgement

This work was supported in part by the U.S. National Science Foundation under Grants IIS 0755405 and CNS 0821585.

References

1. Abramson, G., Kuperman, M.: Social Games in a Social Network. Physical Review (2001)
2. Albert, R., Barbási, A.L.: Statistical Mechanics of Complex Networks. Modern Physics, 47–97 (2002)
3. Shoham, Y., Tennenholtz, M.: On the Emergence of Social Conventions: Modeling, Analysis and Simulations. Artificial Intelligence, 139–166 (1997)
4. Delgado, J.: Emergence of Social Conventions in Complex Networks. Artificial Intelligence, 171–175 (2002)

5. Borenstein, E., Ruppin, E.: Enhancing Autonomous Agents Evolution with Learning by Imitation. Journal of Artificial Intelligence and Simulation of Behavior 1(4), 335–348 (2003)
6. Zimmermann, M., Eguiluz, V.: Cooperation, Social Networks and the Emergence of Leadership in a Prisoners Dilemma with Adaptive Local Interactions. Physical Review (2005)
7. Watts, D.J.: Small Worlds. Princeton University Press, Princeton (1999)
8. Zhang, Y., Leezer, J.: Emergence of Social Norms in Complex Networks. In: Symposium on Social Computing Applications (SCA 2009), The 2009 IEEE International Conference on Social Computing (SocialCom 2009), Vancouver, Canada, August 29-31, pp. 549–555 (2009)