Pure Agent Minority Game

Luezhen Yuan(袁略真)

Student ID: 3130103964. Major: Bioinformatics. Email: 3130103964@zju.edu.cn

1. Introduction

In many social and biological systems agents simultaneously and adaptively compete for limited resources[1]. The agents have heterogeneous strategies, expectations, and beliefs. Minority game is a simple model used to study this system.

This report studied the pure agent(one type of agent) minority game, and answered two questions. From a view of a boss of a bar, she likes the number of customers every day staying at some place(**Question 1**). And, she may also want to predict the number of customers tonight(**Question 2**).

This project implemented the minority game in C++ using the object-oriented programming method, and designed 82 cases to study this system. This study shows that a low changes of the number of customers everyday can be maintained at all the conditions, when the bar provides around 7 days history record. Everyday, the bar can expect the number of customers to be the number 8 days ago.

2. Minority Game Model Implementation

The minority game was implemented using the *object oriented programming* method. I designed three classes: *Game*, *history* and *agent*. When the program starts, a *Game* object is instantiated, and it initializes a given number of *agent* objects and a *history* object. When the *Game::BatchRun* or *Game::Run* method is called, the minority game model starts running. A *agent* object get history record from the *history* object, and makes decision. She sends the decision to the *Game* object, and the latter judges which side wins. The *Game* object send this message to the *history* object and all the *agent* objects, and they update their states. Till now, single game is finished, and the minority game steps to a next cycle. A minority game often runs 10000 cycles, and after that, the game stops, and the program comes to an end. These procedure is shown in figure 1.

Figure 1. Attendance changes when Δt=8.

This program supports *mixed model* in minority game. The users can design many types of agent. A type of agent is a common set of agents that have the same number of strategies and can handle the same size of history record. The following paragraphs show a minority game with two types of agents.

```
Minority Game
```

```
1th type of agents:
s strategies hold for one agent(in this type). [5]:5
m size of memory used by one agent(in this type). [3]:3
Number of agents(in this type). [101]:100

1 for entering next type of agent;-1 for finishing the entry.:1

2th type of agents:
s strategies hold for one agent(in this type). [5]:4
m size of memory used by one agent(in this type). [3]:6
Number of agents(in this type). [101]:101
1 for entering next type of agent;-1 for finishing the entry.:-1
How many time steps in one run? [10000]:10000
How many individual run? [32]:31

Finished

1 for a new game;-1 for exit. [-1]:-1

Finished Minority Game!
GoodBye!
```

The result is shown in table 1.

Table 1. Mixed model result. Type 1: $s_1 = 5, m_1 = 3, N_1 = 100$. Type 2: $s_2 = 4, m_2 = 6, N_2 = 101$.

Type 0th agents' Succeed Ratio	Type 1th agents' Succeed Ratio	Succeed Ratio
0.475248	0.45	0.462687
0.118812	0.07	0.0945274
0.50495	0.49	0.497512
0.50495	0.45	0.477612
0.485149	0.45	0.467662
0.49505	0.5	0.497512
0.445545	0.37	0.40796
0.554455	0.35	0.452736

3. Experimental design

There are three parameters that will affect the standard deviation of the minority game: s strategies hold for one agent, m size of memory used by one agent, and N agents. To study the effects of these parameters, I designed 82 test cases and each cases repeat 32 times.

```
three factor:
s: 3,6,9,12
m: 2,7,12,19
N: 101,201,301,401
```

The design of the first 18 cases was to let the s =3 and N=101, and to change m(ranging from 2 to 19).

The other 64 cases were constructed using the three-way factorial design. Each parameter had four levels(shown behind), so that there were 64 cases. Actually, the choosing levels of m was wrongly designed, so the test cases when m=19(16 cases) will drop in these sections: *Analysis of Quantitative factor on 48 cases*, and *linear regression on 48 cases*.

4. Results

4.1 Standard Deviation of the 18 cases

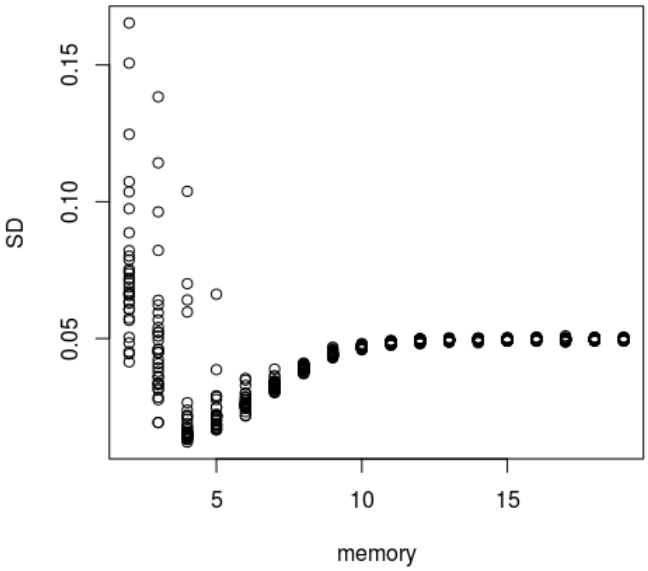

Figure 2. Standard deviation of the Minority game changes with m. Let s =3, N=101. Each parameter combination runs 32 times.

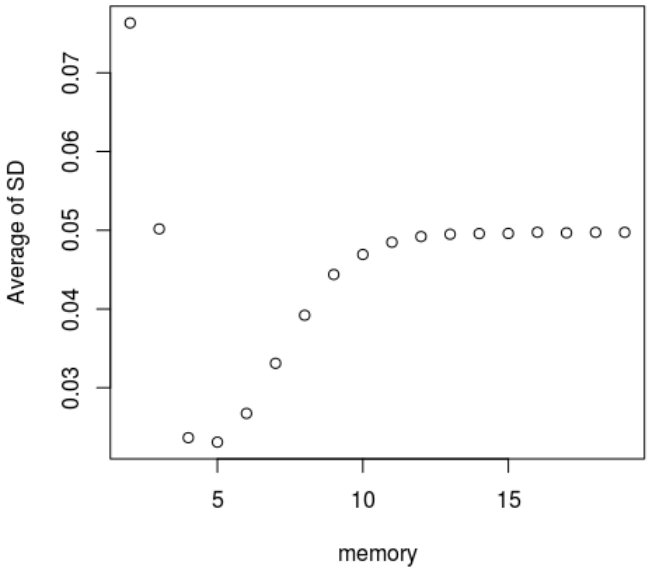

Figure 3. The average of the standard deviations of the Minority game, changes with m over 32 repeats. Let s =3, N=101.

4.2 Analysis of variance(ANOVA) on 64 cases

Anova model:

$$\begin{aligned} SD_{ijkt} &= \mu + s_i + m_j + N_k + sm_{ij} + sN_{ik} + mN_{jk} + smN_{ijk} + \varepsilon_{ijkt} \\ &\sim N(\mu + s_i + m_j + N_k + sm_{ij} + sN_{ik} + mN_{jk} + smN_{ijk}, \sigma_\varepsilon^2) \\ & i, j, k = 1, 2, 3, 4;\ t = 1, ..., 32 \end{aligned} \tag{1}$$

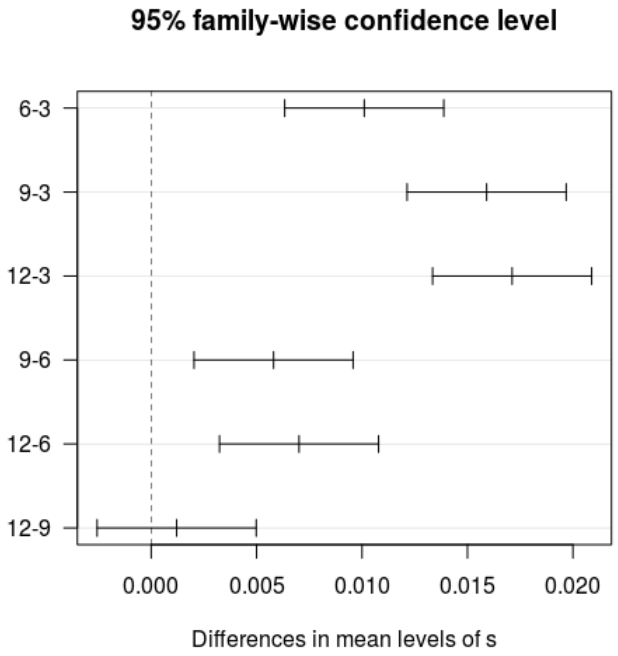

Figure 4. Multiple Comparisons of the mean *SD* of different *s*, using Tukey's procedure.

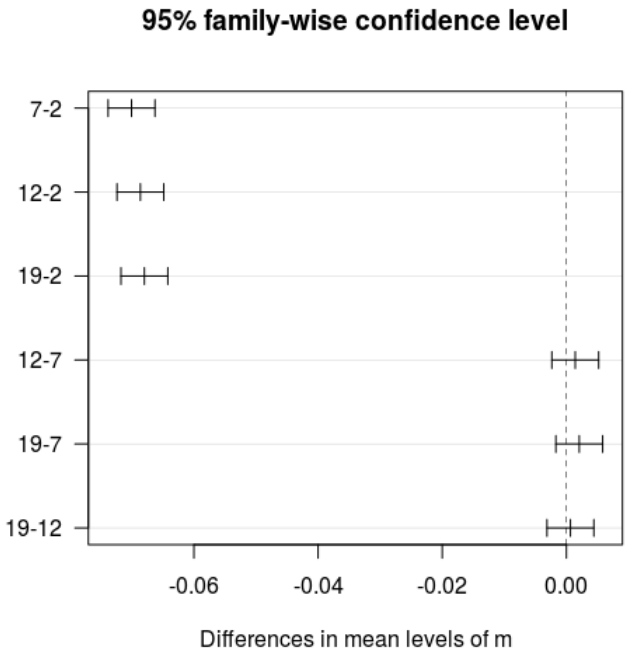

Figure 5. TMultiple Comparisons of the mean *SD* of different *m*, using Tukey's procedure.

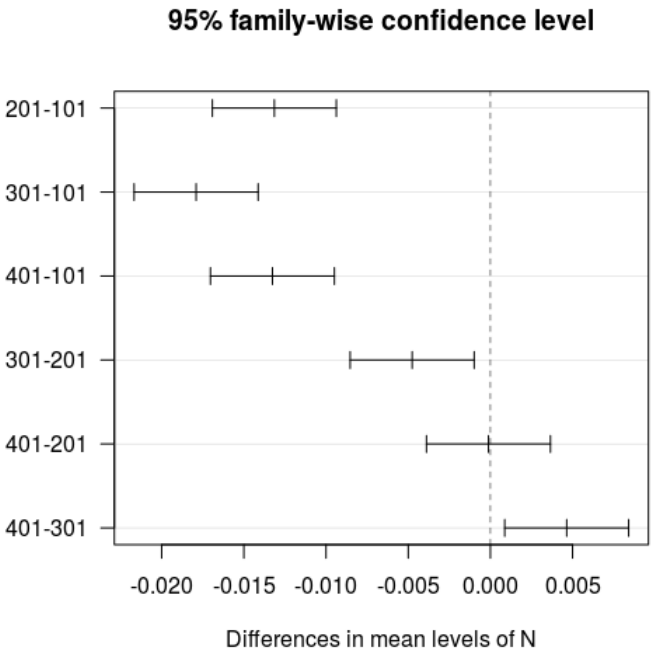

Figure 6. Multiple Comparisons of the mean *SD* of different *N*, using Tukey's procedure.

4.3 Analysis of quantitative factors on 48 cases

ANOVA of quantitative factors can help to separate the variance of SD into linear effects, quadratic effects and so on. To achieve these, I first changed the data:

```
  s m   N         SD
1 3 2 101 0.16536375
2 3 2 101 0.15068368
3 3 2 101 0.12465193
4 3 2 101 0.10735185
```

```
5 3 2 101 0.10362809
6 3 2 101 0.09747083
```

to:

```
      s_l s_q     s_c       m_l      m_q     N_l N_q     N_c         SD
1 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.16536375
2 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.15068368
3 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.12465193
4 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.10735185
5 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.10362809
6 -0.6708 0.5 -0.2236 -0.707107 0.408248 -0.6708 0.5 -0.2236 0.09747083
```

Here, $*\_l$ means linear, $*\_q$ means quadratic, $*\_c$ means cubic.

After that, I found this model is significant:

```
SD~s_l+s_q+m_l+m_q+N_l+N_q+s_l:m_l+s_q:m_l+s_l:N_l+s_q:N_l+s_c:N_l
+m_l:N_l+s_l:N_q+s_c:N_q+m_l:N_q+m_q:N_q+s_l:m_l:N_l+s_q:m_l:N_l
+s_c:m_l:N_l+s_c:m_q:N_l+s_l:m_l:N_q+s_c:m_l:N_q+s_l:m_q:N_q
```

Here the model is written in a R like format for convenience.(R is a language used in statistics)

4.4 Linear regression on 48 cases

Linear model was used for prediction. In order to get a good ability of generalization, I used a simple model, and stepwise regression method shows that there is no simpler model better than this:

$$SD_{ijkt} = \beta_0 + \beta_1 s\_l_i + \beta_2 s\_q_i + \beta_3 m\_l_j + \beta_4 m\_q_j + \beta_5 N\_l_k + \beta_6 N\_q_k + \varepsilon_{ijkt}$$

10-fold cross validation was used to validate the generalization ability. The Cross-Validated R-square was 0.5705647, and the original R-square was 0.5738707.

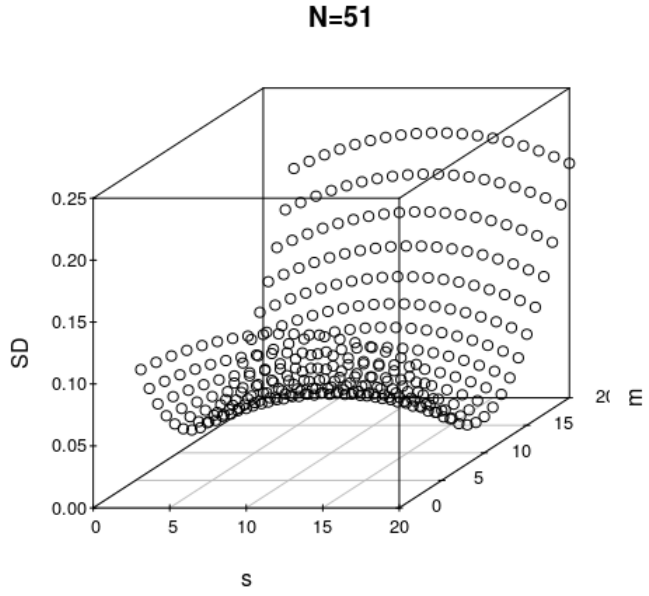

Figure 7. The predicted standard deviation using simple linear regression model. Let N=51, s and m changes from 2 to 20.

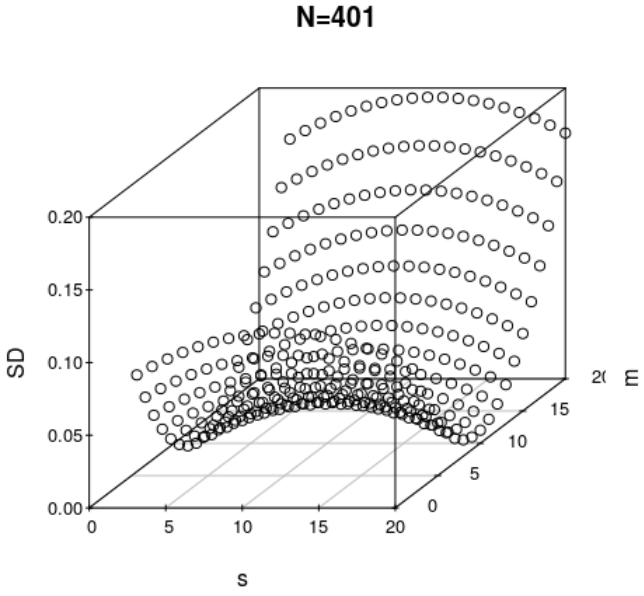

Figure 8. The predicted standard deviation using simple linear regression model. Let N=401, s and m changes from 2 to 20.

4.5 Attendance changes on 18 cases

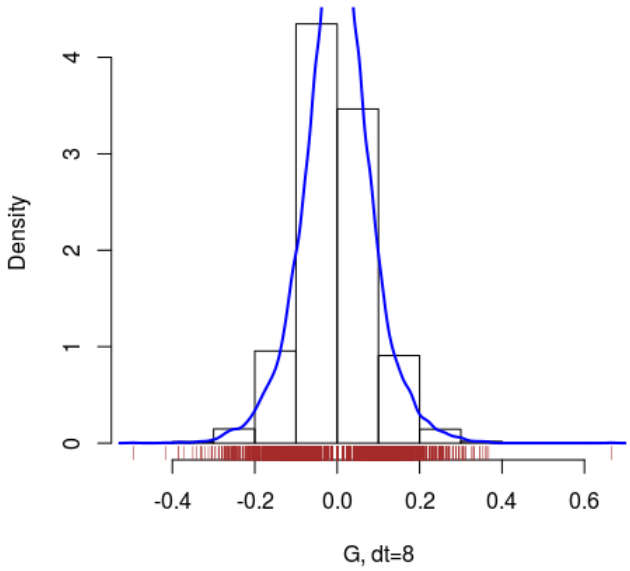

Figure 9. Attendance changes when Δt=8.

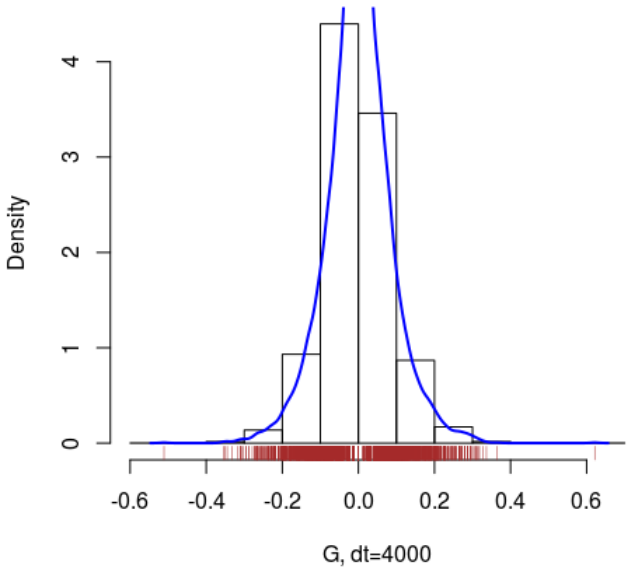

Figure 10. Attendance changes when Δt=4000.

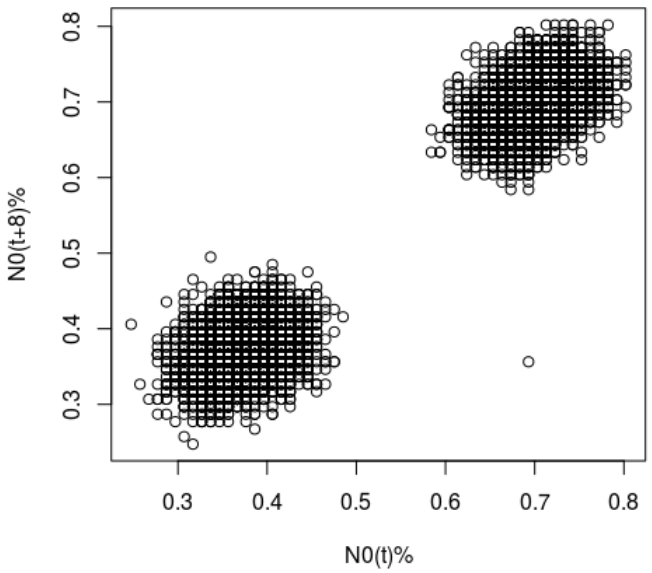

Figure 11. Attendance changes when Δt=8.

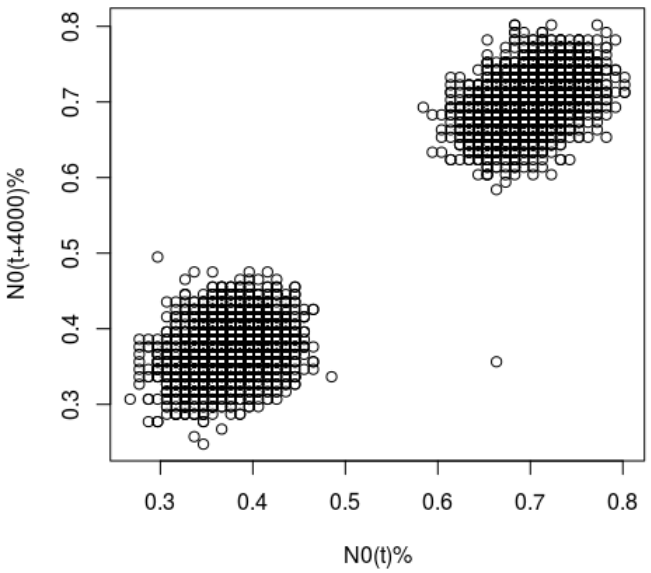

Figure 12. Attendance changes when Δt=4000.

5. Discussions and Conclusions

5.1 How many records provided is good for low standard deviation

Figure 2 and 3 shows that provide 5 days history record will get a low standard deviation and average of standard deviation, when s=3, N=101.

When s and N change, figure 7 and 8 shows that provide around 7 days history record is good for nearly all the cases.

5.2 How to predict the number of customers tonight?

An interesting pattern occurs when the Δt jump from 1,2,..,7 to 8. Figure 9 and 11 shows a 8 days cycle pattern. Figure 10 and reffig:11 provide another evidence that the 8 days cycle pattern exist, even when Δt is a multiple of 8 days.

In conclusion, the boss of the bar should provide 7 days history record, and prepares beer for the number of customers 8 days ago.

Supplement

Source code and analysis procedure for the data analysis can be found in data/README.md . All these analysis is performed in R.

References

[1] R. Savit, R. Manuca, and R. Riolo, "Adaptive Competition, Market Efficiency, and Phase Transitions," Physical Review Letters, vol. 82, no. 10, pp. 2203–2206, Mar. 1999.