Project 5: Monte Carlo simulation of financial interactions between homogeneous agents
Fys 4150

Ask J. Markestad, Thorbjørn V. Larsen

Abstract

In this project we look at some simple agent based models for financial transactions and wealth distribution. We base ourselves on the models presented by Marco Patriarca, Anirban Chakraborti, Kimmo Kaski [2], which has a savings component in the transactions but random transactions, and the models by Sanchari Goswami, Parongama Sen [3] which has interaction preferences. We spend some time analyzing the relations the models of [3] and the lack of information in this paper, make some changes and do in part our own analysis.

Introduction

The field of Financial physics can be said to start in 1897 when V. Pareto [1] found from an empirical study of wealth distribution that the higher end of the distribution followed a power law

$$w_m \propto m^{-1-\nu} \tag{1}$$

with $\nu \in [1, 2]$. This will be the criteria by which all our distributions will be measured, as they will then be considered possibly realistic. The goal of our analysis will be more along the line of seeing how simple changes/additions to the interactions change the wealth distribution. An interesting step further would be to take the distributions which we produce and compare with modern empirical distribution using standard statistical analysis tools of physics today such as hypothesis testing and goodness of fit etc. In this project we concern ourselves with producing the distributions. We start with the models of [2] and present the theory of them, produce the distributions ourselves, and compare with their results. Next analyze the models of [3] and critic them. With slightly expanded models we produce new distributions, compare with their results, and discuss.

Theory and Algorithms

The initial model of [2] is a simple one. It considers a system where money is conserved, and all transactions are between random pairs of agents with a random amount of the money an agent has to give. So, given a total of N agents you select two randomly, agents i and j with money m_i and m_j. There is some transaction such that $m_i \to m_i'$ and $m_i + m_j = m_i' + m_j'$. The transaction will be described by a randomly generated number ϵ between 0 and 1 using a uniform distribution. This leads to the relations:

$$m_i' = \epsilon(m_i + m_j) \tag{2}$$
$$m_j' = (1 - \epsilon)(m_i + m_j) \tag{3}$$

This type of transaction leads to a steady state given by a Gibbs distribution:

$$w_m = \beta e^{-\beta m} \tag{4}$$

with $\beta = \frac{1}{\langle m \rangle}$. We will assume here and for all further distributions an initial state where all agents start with an equal amount of money $m_0 = 1$. This means that the average money for each agent is the initial cash that they have. This also means that the variance in money is given by:

$$\sigma_m^2 = \langle m^2 \rangle - \langle m \rangle^2 = \frac{1}{N} \sum_i m_i^2 - 1 \tag{5}$$

We will use this variance as a measure of reaching the steady state as this quantity will start at zero for our initial distribution and grow until it fluctuates around the value for the steady state, if a steady state can be reached.

The next interaction type of reference [2] leads to non-Gibbs distributions. Here we introduce a savings component (If we were to consider taxation it would either always go to one specific agent to be the government, or simply just be redistributed equally among all the agents if we like Karl Marx). We introduce this savings component as a free parameter λ of the model and implement it as

$$m_i' = \lambda m_i + \epsilon(1-\lambda)(m_i + m_j) = m_i + \delta m \tag{6}$$
$$m_j' = \lambda m_j + (1-\epsilon)(1-\lambda)(m_i + m_j) = m_j - \delta m \tag{7}$$

with $\delta m = (1-\lambda)(\epsilon m_j - (1-\epsilon)m_i)$.

We move onto the models presented in [3]. Goswami and Sen propose that one should include a probability for a transaction to take place, so that not all agent pairs has the same likelihood for interacting. They present three such models with different probabilities, where all the transactions are as in eqs. 3. Interestingly, they discuss the λ model and point out a similarity between the distributions created from said model and the γ model, eq. 9, that we will talk about in a bit. Of the three models that they present we will look at two of these, the first one being

$$p_{ij} \propto |m_i - m_j|^{-\alpha} \tag{8}$$

with $\alpha > 0$. This is intended to simulate that people of equal means are more likely to interact than people of different net worths. The problem here is that Goswani and Sen does not provide, or comment, on the normalization for this probability. The problem with this is that the importance of the normalization is connected to the amount of money in the system, i.e. the initial money. In the paper, they use randomly distributed money as an initial state, but they do not provide the initial amount. Why does this matter? Well, say you are in the situation that there is a low amount of money in the system. Then the transactions will be small, and while the relative wealth distribution should be the same the actual differences between the money the agents have will generally be small, which for this probability gate would mean most transactions would be accepted and the distribution should tend towards the one of eqs. 3. If however, there is a large amount of money in the system then the differences will generally be larger and most transactions will be rejected. Both of these situations are in and off themselves interesting and relevant but when you don't know which situation you are in, or if you normalize to some in between situation, then the distributions do not tell you that much by themselves. The comparison plots between different α value do include relevant data as it tells us something about the important of this probability gate to a degree, however if most moves where accepted to begin with increasing from 1 to 2 does not change the distribution much while the other case gives larger changes to the distribution. And since this information is not provided, recreating the result to confirm is not possible. We have therefore chosen to have the normalization for this probability as a free parameter that we can tweak such that most transactions get probabilities between 0 and 1, though a larger analysis of its importance to the distribution would be a natural next step.

The same problem persists in the next model that is presented where we include a component that prioritizes transactions between agents that have interacted before.

$$p_{ij} \propto |m_i - m_j|^{-\alpha}(c_{ij} + 1)^{\gamma} \tag{9}$$

Where c_{ij} is a matrix that counts the amount of interactions. This of course, introduces a new factor that needs to be normalized. Again, we have a similar problem as above, that given several distributions with different γ value we can attain some relevant insight into the model but with problems in knowing

exactly what the distribution represents and with reproducibility. How much the c_{ij} should increase with for an interaction is an important factor that needs to be considered in congruence with the normalization. The point is that with running up to 10^{10} possible transactions this factor can blow up and produce an always accept situation, and just setting a static value as normalization will not scale with the number of possible interactions like it will in the previous case. This means that how much the c_{ij} increases with says something about how fast you are moved back to model 3. Our solution is then to dynamically normalize with the largest element of c_{ij}

$$\frac{(c_{ij}+1)^{\gamma}}{(c_{ij}(max)+1)^{\gamma}} \leq 1 \tag{10}$$

This does ensure that the importance of previous interactions is still included in the model, but we will never be reduced to a previous model simply by the passage of "time". However, how much c_{ij} in increased with for each transaction between a pair still matters. If the value is increased very slightly, then the chances of leaving an interacting pair "behind" is small, while if the increase is large then it only requires a pair being selected a couple of times in quick succession for "unlucky" pairs to be left behind. And both these cases are interesting, and it says something about how much your model should respect "customer loyalty", and should be included to be able to have reproducibility. Given that [3] does not comment on the normalization it is hard to say if our model with our dynamic normalization is equivalent to theirs, but we proceed with our model as we believe it to be more interesting.

Algorithms

Our Github page for the calculations, program files and benchmarks is found on
https://github.com/ajmarkestad/Fys4150/tree/master/Project5

To simulate these transactions we use the Monte Carlo method. We set up the program such that the function transaction takes in all our free parameter from the models, i.e. number of agents N, λ, γ, α, the normalization, the max element of c_{ij}, and the matrix c_{ij} itself. It also takes in the parameter totaltransactions, which is the number of possible transactions that should occur for each run. This way, we can consider each time the function transaction is called a measurement and do mean values with respect to the relevant number of runs. We further split up the output part of our main loop in two parts, one for what we call initial cycles and then for after. So for the initial cycles we output the variance, eq. 5 of the money per agent. We do this as a way to check for steady state solution and give us the "time" it requires to reach this steady state. A second benefit of this approach is that we cut out all the runs where we don't have a steady state from the final distribution, leaving us with only relevant data. The parameter initial cycles is a command line variable and has to be chosen appropriately for each run, as the "time" required to reach equilibrium changes for the different models. The Histogram function is, self written and puts the number of agents within a cash range into an appropriate bin, outputting the distribution that we want for each model. When writing to file we normalize with (total runs - initial cycles). The variable moneyBins is a pre-made array which contains the money each bin should represent and does not change in the program, while the numberofBins gives us the desired resolution in money.

```
for ( int run= 0; run<= total_runs; run++){
 transaction(previous_interaction_counter,agentlist, numberofAgents, idum,
 total_transactions, lambda, gamma, alpha,normalization, max);
 if(run>= initial_cycles){
  Histogram(Hist, moneyBins, agentlist, numberofAgents, numberofBins);
 }
 else{
  Output_M(numberofAgents,agentlist);
 }
}
```

The main component of the code is the function transaction. It contains all the information about all four of the models that we consider. For each possible transaction, i.e. total transaction, we select two agents

randomly, agent1 and agent2. We test that these are two different agents and if they are, we proceed with calculating the intermediate quantity normalization intercounter which we use in the transaction probability and to find the max value later on. We then create a random number between 0 and 1, and use the transaction probability as a gate, so if the random number is less than the probability then we proceed with the transaction, if it is greater than we reject the transaction and find two new random agents. If it passes the test then cash is exchanged, we update c_{ij} with 0.01 to ensure that we most likely, do not have any runaway pairs, update normalization intercounter and test if it is larger than the previous max value. Note that with the way the probability gate is set up if the transaction probability is larger then 1 all moves are accepted, hence the importance of the normalization parameter to ensure that probabilities are between 0 and 1.

```
void transaction(double **previous_interaction_counter,double *agentlist,
int agents, long& idum, int total_transactions, double lambda, double gamma,
double alpha, double normalization, double &max)
{
 double cash_exchange;
 double normalization_intercounter;
 double transaction_probability;
 for (int i=0; i<total_transactions; i++)
 {
  int agent1 = (int) (ran2(&idum)*(double)agents);
  int agent2 = (int) (ran2(&idum)*(double)agents);
  double transaction_rate = (double) (ran2(&idum));
  if(agent1!=agent2)
  {
   normalization_intercounter =
   pow(previous_interaction_counter[agent1][agent2] +1,gamma);
   transaction_probability =  pow(abs(agentlist[agent1]-agentlist[agent2]),-alpha)
   *normalization_intercounter*(normalization/max);
   if (transaction_probability>1){
    transaction_probability=1.0;
   }
   double transaction_random = (double) (ran2(&idum));
   if(transaction_random<transaction_probability){
    cash_exchange = (1-lambda)*(transaction_rate*agentlist[agent1]-
    (1-transaction_rate)*agentlist[agent2]);
    agentlist[agent1]-=cash_exchange;
    agentlist[agent2]+=cash_exchange;
    previous_interaction_counter[agent1][agent2] += 0.01;
    previous_interaction_counter[agent2][agent1] += 0.01;
    normalization_intercounter =
    pow(previous_interaction_counter[agent1][agent2] +1,gamma);
    if(normalization_intercounter> max){
     max = normalization_intercounter;
    }
   }
  }
 }
}
```

Results

Conclusion

References

[1] V.Pareto *Cours d'economie politique* 1897 http://www.institutcoppet.org/2012/05/08/cours-deconomie-politique-1896-de-vilfredo-pareto

[2] Marco Patriarca, Anirban Chakraborti, Kimmo Kaski *Gibbs versus non-Gibbs distributions in money dynamics* Physica A: Statistical Mechanics and its Applications Volume 340, Issues 1–3, 1 September 2004, Pages 334-339, ISSN 0378-4371 http://dx.doi.org/10.1016/j.physa.2004.04.024. http://www.sciencedirect.com/science/article/pii/S0378437104004327

[3] Sanchari Goswami, Parongama Sen *Agent based models for wealth distribution with preference in interaction* Physica A: Statistical Mechanics and its Applications Volume 415, 1 December 2014, Pages 514-524, ISSN 0378-4371 http://dx.doi.org/10.1016/j.physa.2014.08.018. http://www.sciencedirect.com/science/article/pii/S0378437114006967