

Model of the Stock Market

Fys3150

Fredrik Østrem
github.com/frxstrem

Joseph Knutson
github.com/mathhat

December 7, 2016

Abstract

Contents

1	Introduction	1
2	Theory of Models	2
2.1	2nd Model) The Saving Factor λ	2
2.2	3rd model) The Preferential Factor α	2
2.3	4th model) Memory Factor γ	3
3	Methods	3
3.1	Monte Carlo	3
3.2	A Metropolis Algorithm	3
A	stuff	4

1 Introduction

The aim of this project is to simulate transactions of money between financial agents, people, using Monte Carlo methods. The final goal is to extract a distribution of income as function of the income m . From Pareto's work (V. Pareto, 1897), and from empirical studies, it is known that the higher end of the distribution of money, rich end, follows a power distribution.

$$\omega_m = m^{-1-\alpha}$$

with $\alpha \in [1, 2]$. We will follow the analysis made by Patriarca and collaborators.

We assume we have N agents that exchange money in pairs (i, j) . We assume also that all agents start with the same amount of money $m_0 > 0$. At a given 'time step', we choose a pair of agents (i, j) and let a transaction take place. This means that agent i 's money m_i changes to m'_i and similarly we have $m_j \rightarrow m'_j$. Money is conserved during a transaction:

$$m_i + m_j = m'_i + m'_j. \quad (1)$$

In order to decide which agent gets what, we draw a random number ϵ . The change is done via a random reassignment (a random number) ϵ , meaning that

$$m'_i = \epsilon(m_i + m_j), \quad (2)$$

leading to

$$m'_j = (1 - \epsilon)(m_i + m_j). \quad (3)$$

The number ϵ is extracted from a uniform distribution. In this simple model, no agents are left with a debt, that is $m \geq 0$. Due to the conservation law above, one can show that the system relaxes toward an equilibrium state given by a Gibbs distribution

$$w_m = \beta \exp(-\beta m),$$

with

$$\beta = \frac{1}{\langle m \rangle},$$

and $\langle m \rangle = \sum_i m_i / N = m_0$, the average money. It means that after equilibrium has been reached that the majority of agents is left with a small number of money, while the number of richest agents, those with m larger than a specific value m' , exponentially decreases with m' .

In each simulation, we need a sufficiently large number of transactions, say 10^7 . Our aim is find the final equilibrium distribution w_m . In order to do that we would need several runs of the above simulations, at least $10^3 - 10^4$ runs (experiments).

2 Theory of Models

Now there are multiple models to explore. Our first model picks two entirely random agents, i and j , and makes them trade a random amount decided by the factor ϵ (see equation 2 and 3). The following three models each add another parameter to the previous resulting in more realistic distributions of money.

2.1 2nd Model) The Saving Factor λ

In our simplest model, nothing is stopping our agents from throwing away all their money. In order to make our agents a bit more rational, we will limit the amount of money they trade away.

The conservation law of Eq. (1) holds, but the money to be shared in a transaction between agent i and agent j is now $(1 - \lambda)(m_i + m_j)$. This means that the amount of money our previous model would transfer is reduced by $\lambda\%$

which can be written as

$$m'_i = m_i + \delta m \quad (4)$$

and

$$m'_j = m_j - \delta m, \quad (5)$$

with

$$\delta m = (1 - \lambda)(\epsilon m_j - (1 - \epsilon)m_i), \quad (6)$$

From Eq.(6), if $\lambda = 0.8$, our agents will never trade more than 20% of their total money. As λ goes to 1, our agents only engage in microtransactions. This reduces risk and you will soon see how helpful λ can be for the reduction of poverty.

2.2 3rd model) The Preferential Factor α

Our next model is the same as the last, except our agents will not only have risk reduction through λ , but also a preferred trading partner through α . How our agents will come to prefer each other will be based on how equal they are economically.

$$P(i, j; \alpha) \propto |m(i) - m(j)|^{-\alpha} \quad (7)$$

We use the expression in Eq.(7) to produce a probability for agent i and j 's transaction. We can't know what the proportionality constant is without calculating the probability of every pair of agents (which is costly). Instead we generate a random number r between 0 and S , such that S is larger than the probability expression some large fraction of the time. We discard a transaction between pair of agents if r is less than the probability expression in (7). If S is too small, any pair where the probability expression in (7) is larger than S

will always be accepted, causing a less accurate distribution, while if S is too large, too many pairs of agents will be discarded, so computation time will take longer.

2.3 4th model) Memory Factor γ

This time, we're adding another parameter to the way the agent's prefer trading with each other. In real life, people build trust with those they've traded with before. We're going to make the agent's remember each other. The goal is to make them prefer those they've traded with before. The more trades that a pair has committed in past, the more likely it is for that pair to trade again in the future.

The new probability expression will contain a new factor:

$$P(i, j; \alpha, \gamma) \propto |m(i) - m(j)|^{-\alpha} (C_{ij} + 1)^\gamma \quad (8)$$

C is a matrix which stores the amount of times that agent i and j have traded. This means that each time agent i and j trades, C_{ij} grows with 1.

We see that for $\gamma = 0$, the probability of a transaction is the same as before, but for $\gamma > 0$ the probability of trade increases.

3 Methods

To model the trades between our agents, we use random sampling, otherwise known as Monte Carlo methods.

3.1 Monte Carlo

We have both used different random number generators. Fredrik used a c++11 RNG and I used `rand()`, a function from C. With these we were able to draw pairs of random agents, produce probabilities and compare our agents' money and history to determine if a pair should trade.

3.2 A Metropolis Algorithm

The last two models are the only models where the agents gain preferences of who to trade with. For this, we added two probability expressions with relatively low acceptance ratios. If a random number between 0 and S was lower than the probability expression produced by the random pair, then the transaction was allowed.

References

- [1] navn, *Hvor det er fra*, University of Oslo, 2016.

A stuff

Explainexplain