Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB

Project Report

Opinion Dynamics in Social Networks

Ciro Salzmann, Thomas Leu, Théophile Messin-Roizard

Zürich, 18.12.2016

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Opinion Dynamics in Social Network

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
SALZMANN	CIRO
LEU	THOMAS
MESSIN-ROIZARD	THEOPHILE

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

ZÜRICH, the 18.12.2016

Signature(s)

T. Leu

Ciro Salzmann

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Agreement for free-download

We hereby agree to make our source code of this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Ciro Salzmann, Thomas Leu, Théophile Messin-Roizard

Table of Contents

Abstract

The purpose of this research was to study the dynamics of opinion formation on a theoretical level and to find a strategy to maximise the number of persons supporting a specific opinion. For this purpose a voter model based on a cellular automaton was designed and then implemented and simulated in MATLAB. In an experimental phase we observed visually how our model reacts to different conditions. Then we defined two strategies for “buying” voters or more specifically for targeting the bought voters. The first strategy has a targeted approach and the second an untargeted. We simulated both strategies on different network structure. The result was for all tested structures that if the cost of a voter is proportional to the number of other voters they can influence, the untargeted approach performs better in terms of the price-to-performance ratio.

Individual contribution

Everyone contributed in the development of the models, the implementation in MATLAB and the analysis of the results.

Introduction and Motivation

With the presidential elections in the USA upcoming, we asked ourselves which strategy for targeting voters would result in the most desired outcome. The underlying question is: How do opinions get formed and how do they change over time? If we had the answer to this question we could apply it to a much broader field than only politics.

We concentrated our research on the influence of individuals between each other in their social networks (i.e. friends, family, colleagues at work etc.). Our intuition was that the influence of the media on the voters is small compared to the social influence, especially after most people already formed an opinion. Nasser (2013), who reviewed studies on the influence of the media, concluded, “the role of the media in influencing election results is generally quite small”.

To model the dynamics of the social interactions, we used a form of the so-called voter model, which was first introduced by Clifford and Sudbury (1973) and has been extended and varied countless times. We used two of those extensions (described in “Description of the Model”) to create a setting, where we can simulate different strategies for targeting voters.

We came up with two main strategies: The first is to identify the most influential persons in the network and get them to support you. One can think of “getting them to support you” as paying them some amount of money. In practice, this would not necessarily look like a bribe but maybe more like making a campaign that specially targets those individuals, which of course also costs something. The second strategy is to just get random persons to support you. We will refer to those strategies as targeted approach and untargeted approach respectively.

Before we get to the final stage of comparing the two approaches, we will go through a few experiments, in order to see how the opinion spreads qualitatively.

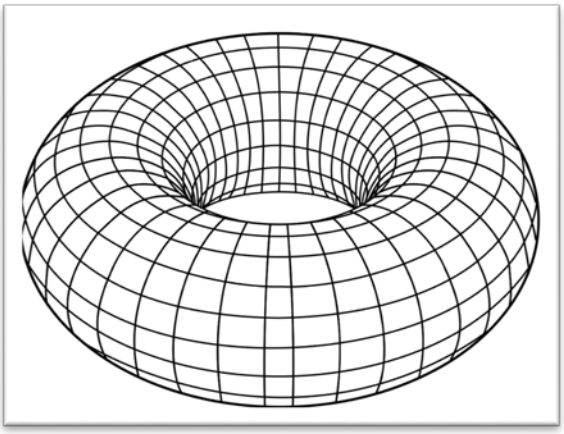
Description of the Model

Cellular Automaton

The cellular automaton (CA) is a discrete model that is used in broad range of science. It consists of a grid in a finite number of dimensions. Each cell of the grid holds a certain state, which can change over time according to an update rule. If a probabilistic rule is used, the model is referred as a probabilistic cellular automaton (PCA). For each cell, a set of neighbour cells is defined. At time t=0, an initial state is assigned to each cell in the grid. Every time step (iteration), the new state of the cells is assigned according to the current cell's state and the state of its neighbours using the update rule. This results in a new *generation*. The CA can be divided into synchronous and asynchronous. In the synchronous CA, the updates happen simultaneously while in the asynchronous CA, the updates come one after another.

In order to visually represent the dynamics of opinion spreading, we used asynchronous, two-dimensional CA where each cell represents a voter and the state is the voter's opinion.

Since the grid is finite, cells close to the borders would have a different behaviour from the others. In the example of a small town it can be a good approximation, since a town has boundaries and finite size. On the other hand, especially in a rather small grid, the distortion of the results due to the border could be substantial. To mitigate the problem we decided to construct the grid in a toroidal geometry in addition to the normal grid. Cells at the top edge are 'stretched' and connected to the bottom cells. Similarly, cells at the left edge are connected to the cells on the right edge. This way the grid has 'continuous boundaries' even if it is still finite, solving all boundary problems. The results of the grid with and without toroidal shape were then compared.



Representation of a torus, a toroidal shape

Voter model

The voter model describes a process over a network in which each node represents an entity holding one of two possible opinions. The word *entity* here is used because generally it must not necessarily be a single person, it can also be e.g. a group or an institution. However, in our model we think of them as single persons, which we will sometimes call *voter* or *agent*. The edges of the network refer to the interpersonal connections of the people. If there is a connection between a pair of voters, we call these neighbours - which does not mean the represented persons must be neighbours in the real world, but rather that they know each other. The voters adapt their opinions to that of their neighbours.

One can find many different update rules in literature. In this model a very simple probabilistic rule: at every time step randomly selected voter adopts the opinion of a random neighbour.

Note that the voter model is based on a network and the CA defines such a network.

Extensions of the voter model

Stubborn Agents

One can easily see that in the voter model described above, a total consensus will eventually be reached, i.e. after enough time, all agents will eventually hold the same opinion. (At any time exists a possible sequence of updates that would result in consensus and eventually one of those sequences will be chosen.) In reality though, we often don't see people reaching a consensus, particularly in the context of elections. This is why Yildiz et al. (2013) introduced the model of stubborn agents. In this model there are two types of voters:

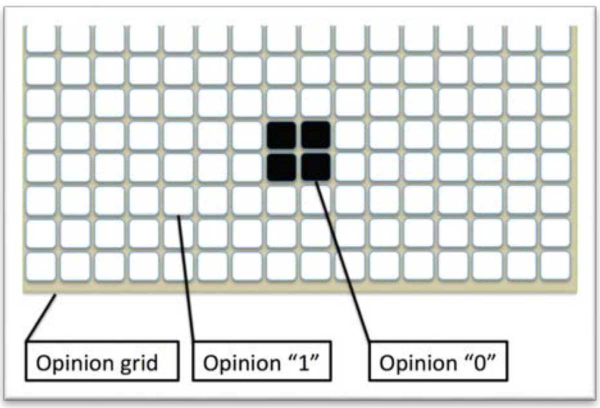
- Stubborn agents
- Non-stubborn agents

Stubborn agents are not influenced by their neighbours and will not change their initial opinion at any time, while still influencing others. Stubborn agents can be characterized as voters being part of the hard core electorate of a party.

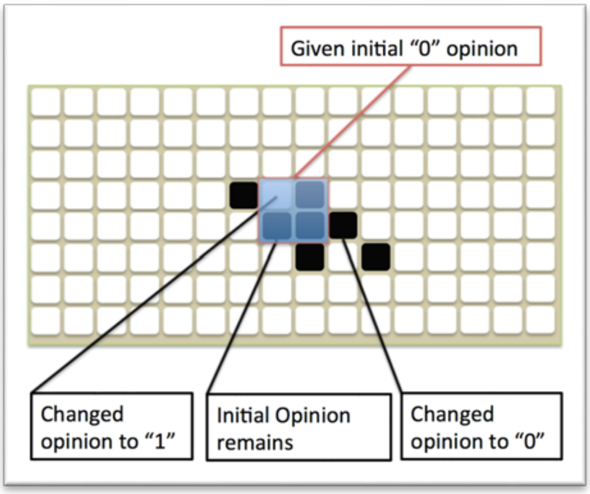
Non-stubborn agents on the other hand behave like the agents in the classic voter model. They may accommodate their opinions to the one of their neighbours, but also influence their neighbours. One may picture them as voters that don't feel committed to a specific party.

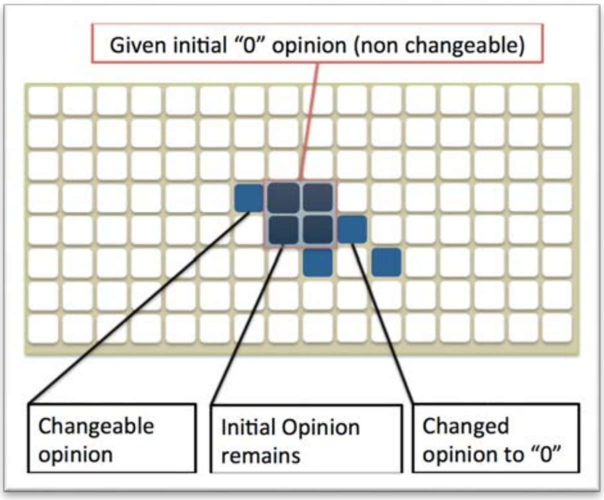
The introduction of stubborn agents into the model will prevent consensus within the network, if there is at least one of them for both opinions. This is trivial, because at any time each of the opinions is held by at least one of the agents (which is the stubborn one).

A general schematic of the model can be seen on the figure:



The grid structure used for the model with a given initial opinion distribution





Left: Possible evolution of the opinion distribution using non-stubborn agents
Right: Possible evolution of the opinion distribution using stubborn agents

Influence model

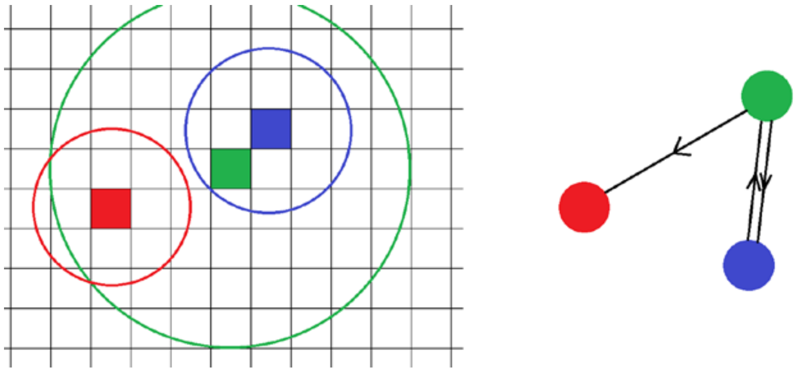
Naturally, the number of people on whose opinion one individual has influence on is not the same for all individuals. This may be due to a number of factors, e. g. the age of the individual, their profession, or simply how many times they tell others about their opinion.

Watts and Dodds (2007) proposed a model of interpersonal influence. This model embodies a network in which the degrees of the nodes follow a certain random distribution. They call this distribution *influence distribution* and argue that it should probably be a Poisson distribution.

Since we wanted our model to be a cellular automaton, we couldn’t attach the nodes at random positions in the network like Watts and Dodds did. Instead, we accredited each cell with an *influence radius*. The voter in that cell has influence on the voters in all the cells within the influence radius. Note that the influence is always directed, a node with a higher influence radius is not bound to get influenced by a high number of other nodes, but the opposite is neither the case.

This has an important consequence: the corresponding graph is directed. In terms of graph theory we distinguish between incoming neighbours and outgoing neighbours.

A basic example of the grid with two nodes and the respective directed graph are shown below:

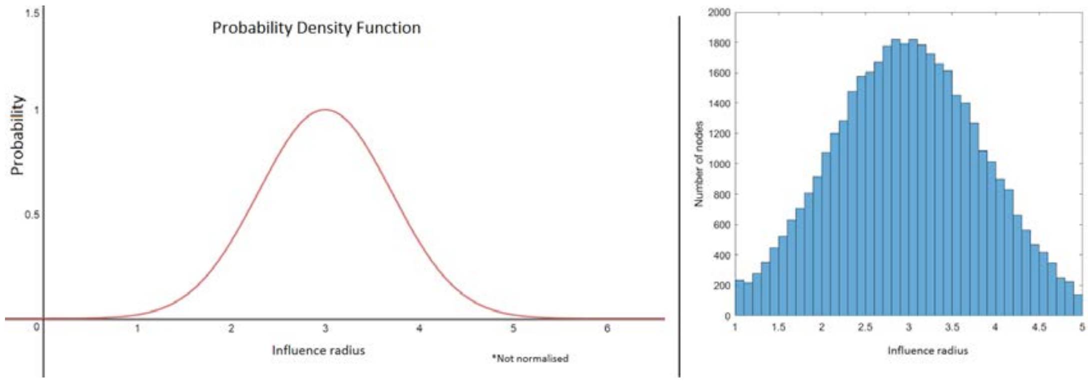


Left: Grid with influence radii of 3 nodes Right: directed graph only showing the coloured nodes

The arrow determines the direction of influence. In the figure above, the green node is an incoming neighbour to the red node, i.e. green influences red. The other way around is not the case.

The influence radii are randomly distributed, following a probability density function (PDF). For each valid value of the radius the PDF gives the probability that a voter is assigned to this radius. We excluded the possibility for nodes to have a radius smaller than 1, because they would have no influence at all.

An example for a distribution is shown below:



Left: probability density function Right: resulting histogram of radius distribution for 400'000 nodes

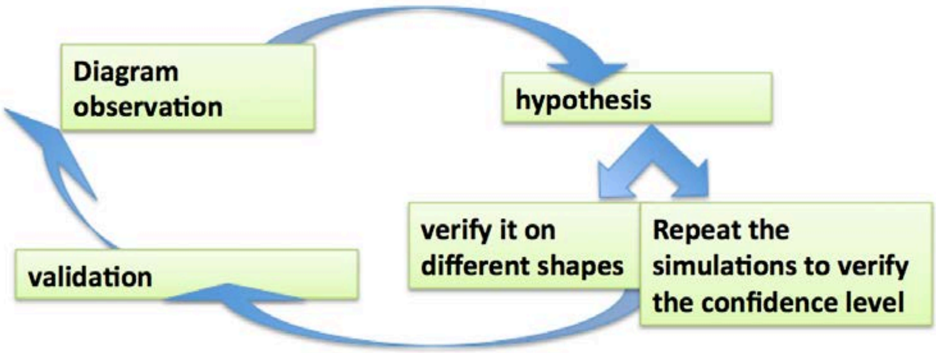
In the example, the chosen PDF is Gaussian (Normal). We also tried other PDFs such as Exponential in order to follow the idea of the Poisson influence distribution of Watts and Dodds.

Experimentation Phase

During an experimentation with a given initial opinion distribution, the development of the geometrical shapes of the distribution of the Opinions (colours) were compared.

Our goal was to create a big enough variety of shapes to identify better the rules followed by the model over time.

To be certain of our presumptions we implemented various shapes. For example we tested the reaction of the modelisation with a single point, a bigger point or a very big point as input in order to watch how the cluster evolves over time and maybe see it disappear or watch it get into a stable mode.



Method for defining the rules the Model follows

Strategy Simulation

As already mentioned in the introduction, our main goal was to compare the targeted approach to the untargeted approach. Both strategies rely on buying agents, to make them spread a specific opinion. We think it makes sense to set the bought agents as stubborn agents.

We created various initial configurations - a configuration consists of the initial opinions, but also of the network structure, i.e. the influence radii distribution - and then applied each of the approaches to that same initial configuration.

Implementation

Construction of the network and Grid:

The dimension of the grid was kept constant for all simulations: a 200 by 200 grid, resulting in 200^2=40'000 cells. It is big enough to make generalisations and small enough to let the simulation be run in an acceptable time. For each cell, we had multiple values to store (the current opinion of the agent, the influence radius, the stubbornness of the agent). We could have saved all the data in a multidimensional matrix, but we decided it was easier and more intuitive to use multiple matrices of size 200 by 200.

During the experimentation phase, the initial opinions were assigned by importing a black and white image, where each colour represents one opinion. We think this is a good choice to quickly test different shapes of initial distributions. Another advantage of this approach is that we could use the generated output image, i.e. a coloured map of the opinions after the simulation (that we wanted to see anyway), as input for another (next) simulation.

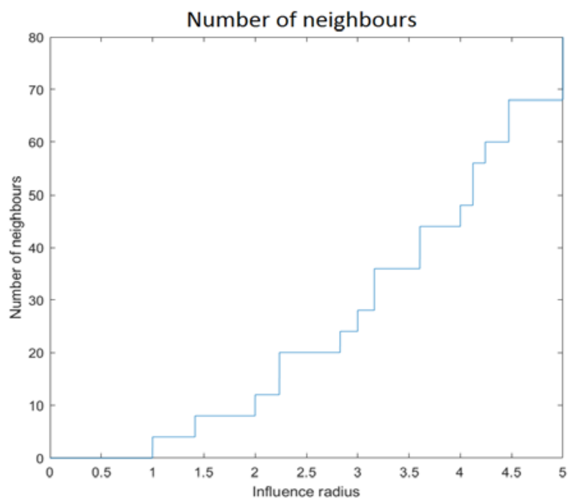
In the strategy-testing phase we also used random initial opinions, for which we could easily set the desired percentages for each opinion.

Other parameters that we varied are the number of iterations, i.e. the number of updates in one simulation, the maximum influence radius, and the probability density function.

Before running the actual simulation, the network is created (creation of the neighbourhood sets). For each cell in the grid a list of neighbours is created by comparing the strength of the possible neighbour with its distance to the specified node. The distance between them is easily computed by using their coordinates within the grid, as seen below:

$$Distance = (x_1 - x_2)2 + (y_1 - y_1)2$$

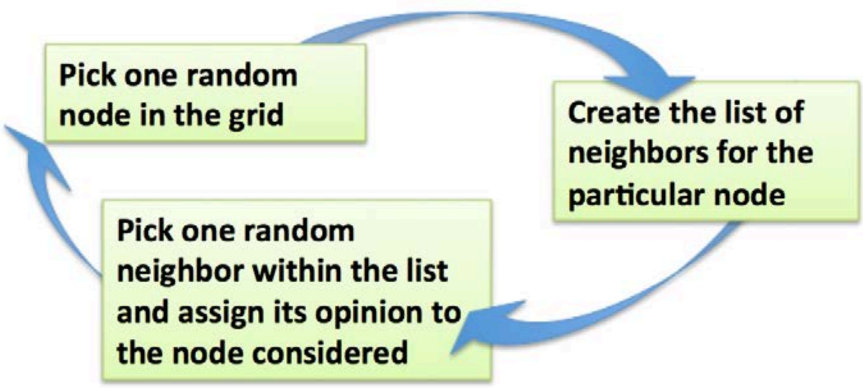
Where (xi, yi) is the coordinate of node i.



Dependency of the maximum number of outgoing neighbours and the influence radius

The simulation algorithm:

The simulation runs a very simple algorithm, briefly described below:



Algorithm implemented in MATLAB

At each iteration, one random node from the grid communicates with its neighbours and adopts the opinion of one of them. It is important to note that the probability of taking the opinion of a neighbour is the same for all neighbours. Stubborn agents do not change opinion and therefore, if randomly chosen, skip one iteration without any updates in the grid.

Strategy simulation

The testing of both approaches can be best described with pseudo code. One run for one initial condition looks conceptually as the following:

```
create initial configuration init
create list targetedList of all voters and influence radii, ordered by descending influence
radius
create list untargetedList of all voters in a random order

for percentage = 0:10 in 100 steps
        bought = the first percentage voters from top of targetedList
        targetedResult = runSimulation(init, bought)
        bought = the first percentage voters from top of untargetedList
        untargetedResult = runSimulation(init, bought)
        save both results in a table/matrix
end
```

Where the function *runSimulation(init, bought)* first overwrites the opinion of the *bought* agent, sets them as stubborn agents and then runs the already described simulation loop.

Because our model contains a lot of randomness, we had to run everything multiple times and look at the mean numbers. Since our simulation algorithm still took some time to complete (note that the function *runSimulation* was called 200 * 24 = 4800 times for one initial configuration) and since the variance in the data appeared to be sufficiently small, we chose an ensemble size of only 24.

Regarding the runtime of the simulation, we also had to make the computation of the list of incoming neighbours for each node before running the simulations, within which the neighbours don't change. This made the simulation a lot faster.

The different initial configurations we used in the strategy testing are described in the next section.

Simulation Results and Discussion

Spreading of the opinion

Our experimentations always start with the choice of a specific geometric shape. This choice results in a singular evolution of the opinion map over time. In order to analyse the evolution of the opinions' diffusion, a simple grid with boundaries (no toroidal shape) was used. Here the used influence radius distribution (probability density function) is:

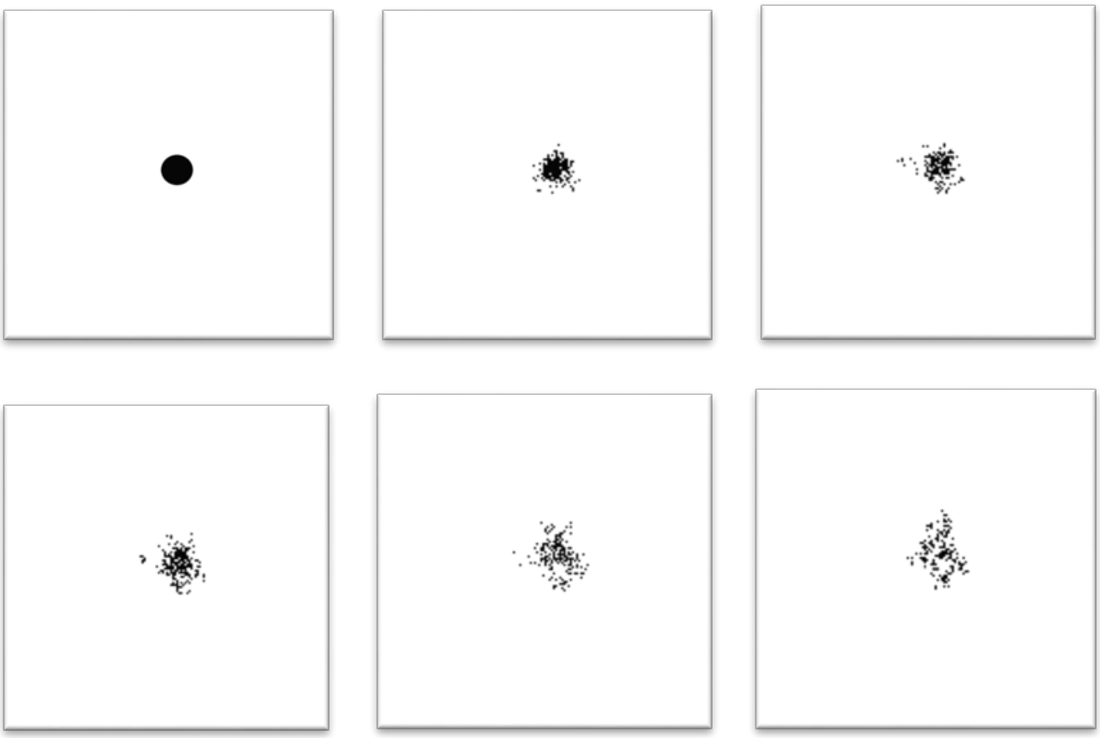
$$f(x) = e^{(-32(x-2)2)} \qquad x\ [1,5]$$

Note that the influence radius for this case will only range from 1 to 5. Furthermore the function is not the probability density function itself, since it is not normalised. It is a reference function that will be normalised at initialisation.

Note: each output comes after 40'000 iterations.

Example 1, small disc

First, with the observation of a small circle disappearing pretty quickly we can state that small opinions cluster use to disappear over time.



A small disc implemented for the simulation (Radius = 1/20 of the height = 10 Agents) and the first 5 image outputs.

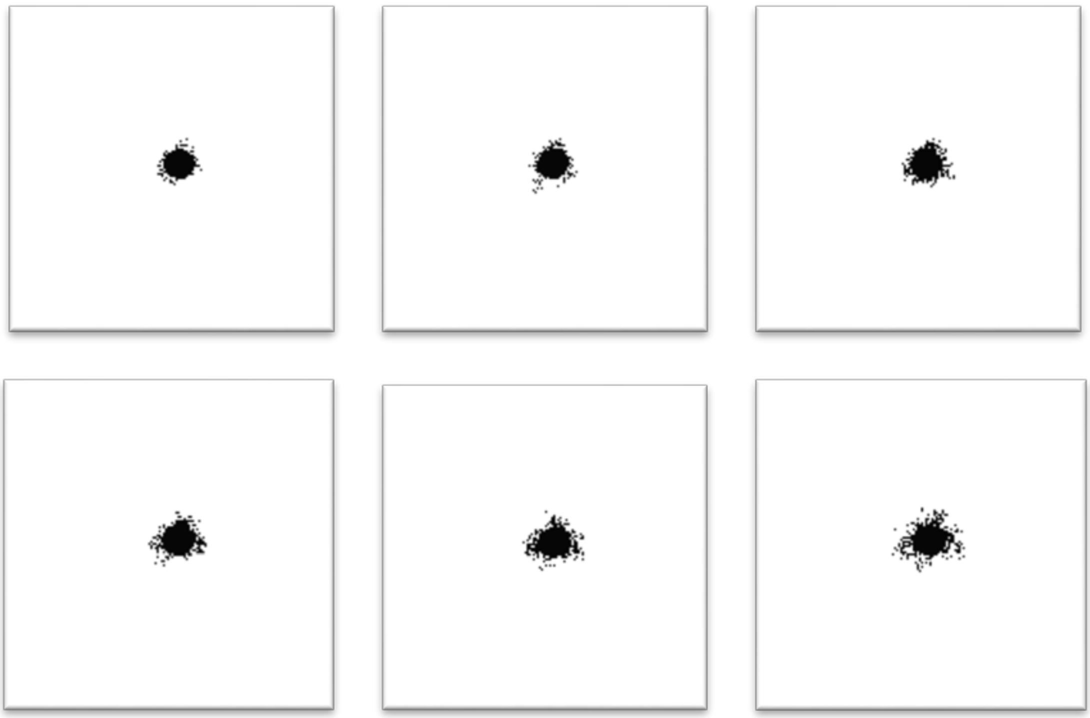
Small disc after 400’000, 1’000’000 and 3’000’000 iterations

It is interesting to note that after 3’600’000 iterations, the opinion map remains white. (there is no more agent having a “0” opinion)

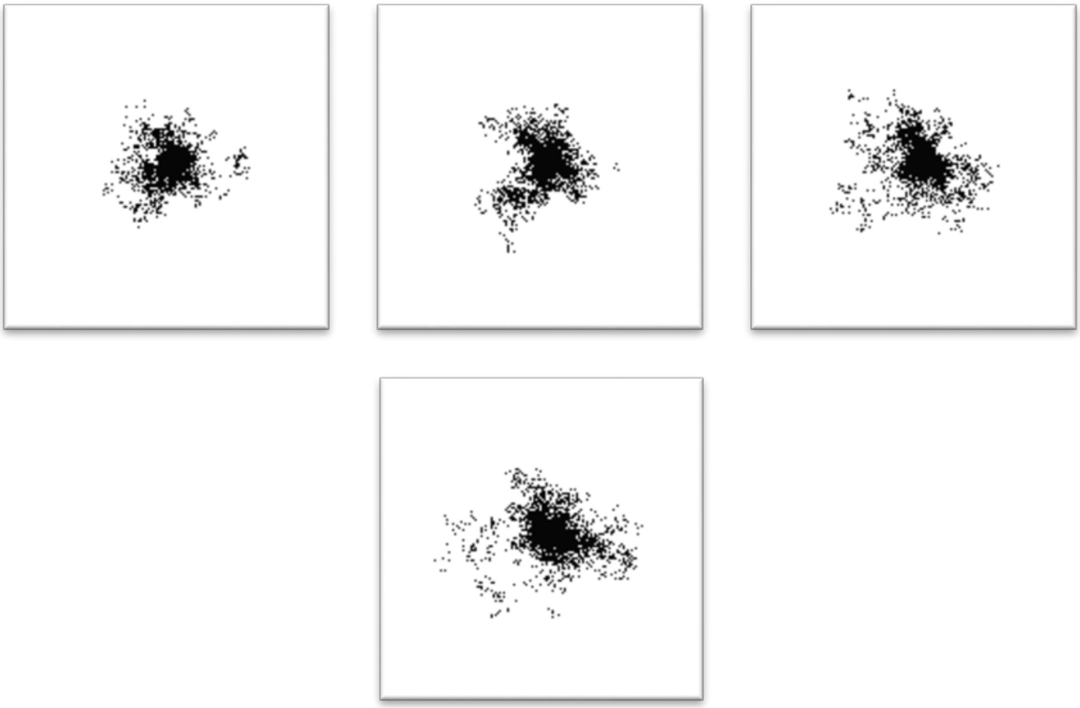
Out of this observation we can state that small enough opinion cluster use to disappear. But we must state that first, the big cluster transforms itself into many smaller clusters. Those ones become less compact and form some very small clusters which will accumulate one with an other. This can be observed over a long time (approximately 3 million iterations) before the density of the “zero” opinion decades radically fast. This last step corresponds to the elimination of the last agents having a different opinion to more than 90% of their neighbours (see definition of neighbour).

Example 2, small disc with stubborn agents

Implementing the same disc but deciding the initial “0” opinion to remain “0”, we get:

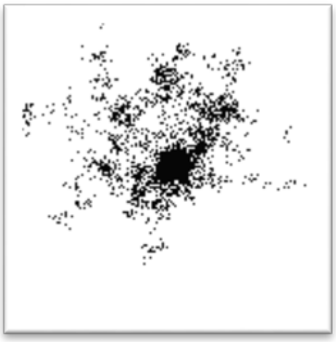


The six first iterations, using a small disc with stubborn agents.

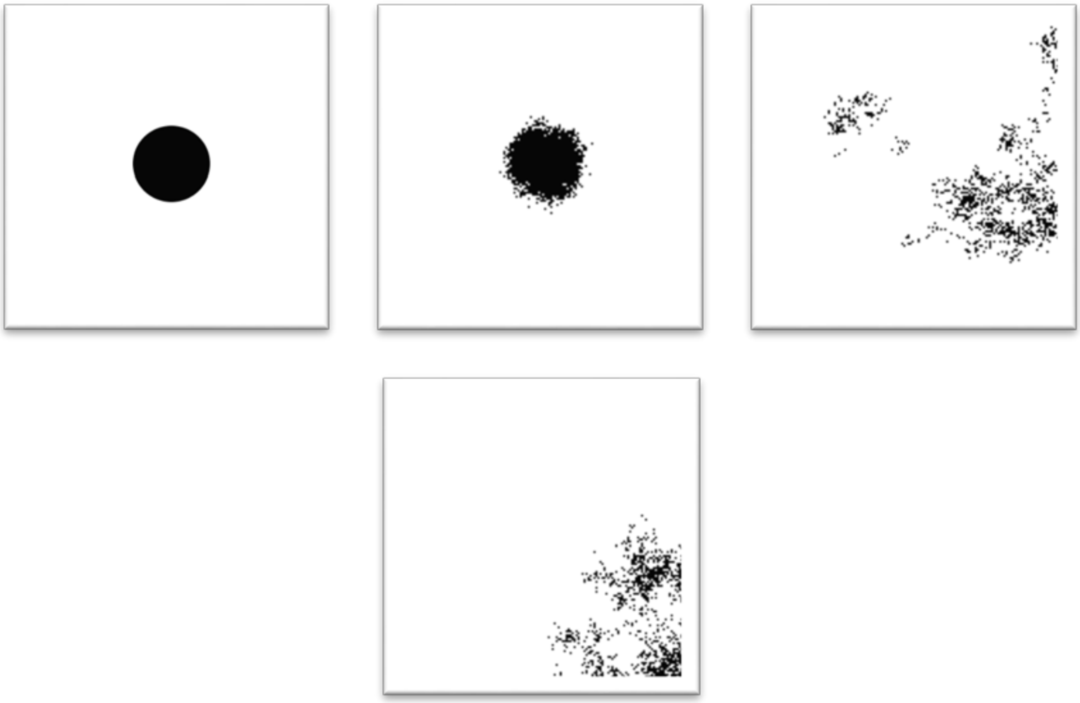


Small disc with stubborn agents after 2, 2.4, 2.8 and 3.2 million iterations

This other simulation enable us to observe the same mechanism. The only difference here results in the presence of a source in the opinion map. The source is the reason of the creation of small cluster which get further away from the source. As stated in the first example, those small clusters will disappear (because it globally veers from the source) unless it comes together with another cluster. Note that the diffusion from the source is not fast enough to maintain every single cluster alive. Therefore we can observe some opinion waves coming from the source. The waves veer always farther and farther away from the source. After 10’000’000 iterations, the majority of the new convinced agents remains within 5 times the radius of the initial cluster.



The majority of the new convinced agents remains within 5 times the radius of the initial cluster.

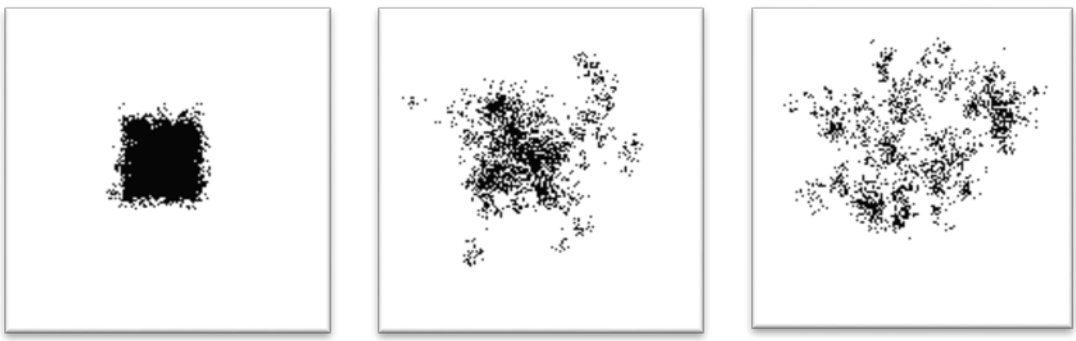


A medium-size disc implemented for the simulation (Radius=1/10 of the height of the map)
The initial disc, the disc after one, 10'000'000 and 20'000'000 iterations

Here again, we observe the same general evolution. Furthermore we see how two much smaller clusters than the initial cluster are forming on the middle right and on the bottom right of the grid.

Without letting the simulation run any longer, we know that the cluster will disappear within a high number of iterations.

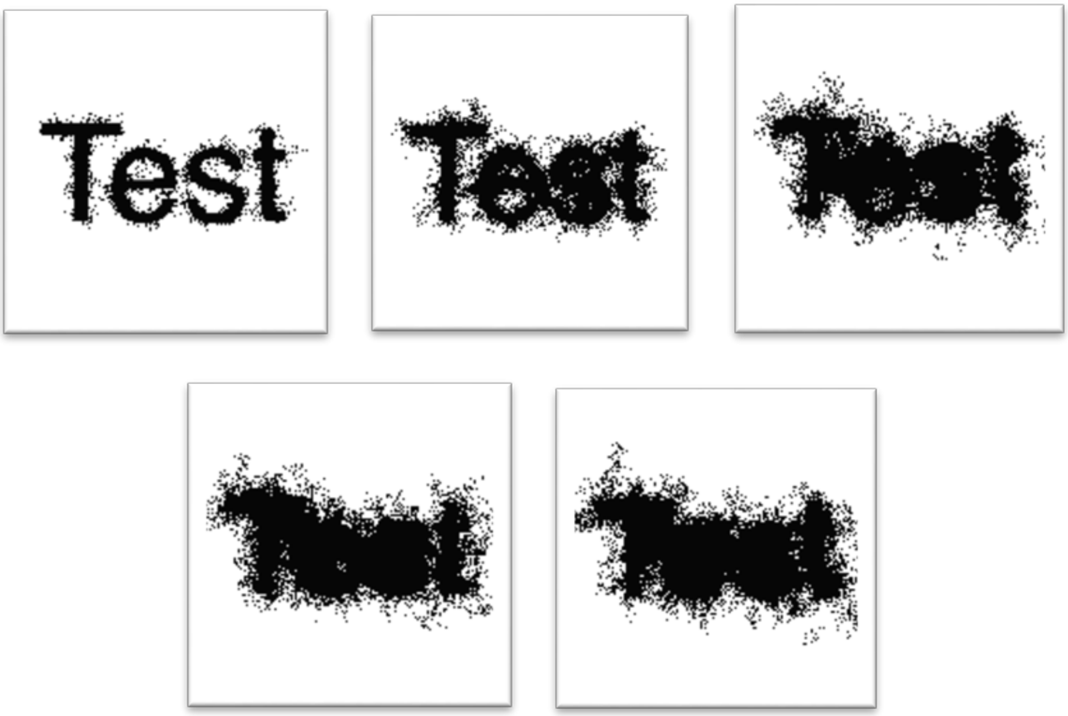
The same results can be observed with a square of approximately the same size.



A square in the simulation after 1, 1.6 million and 3.2 million iterations

Example 3, enclosed areas

We now want to observe the effect of closed or semi-closed spaces.



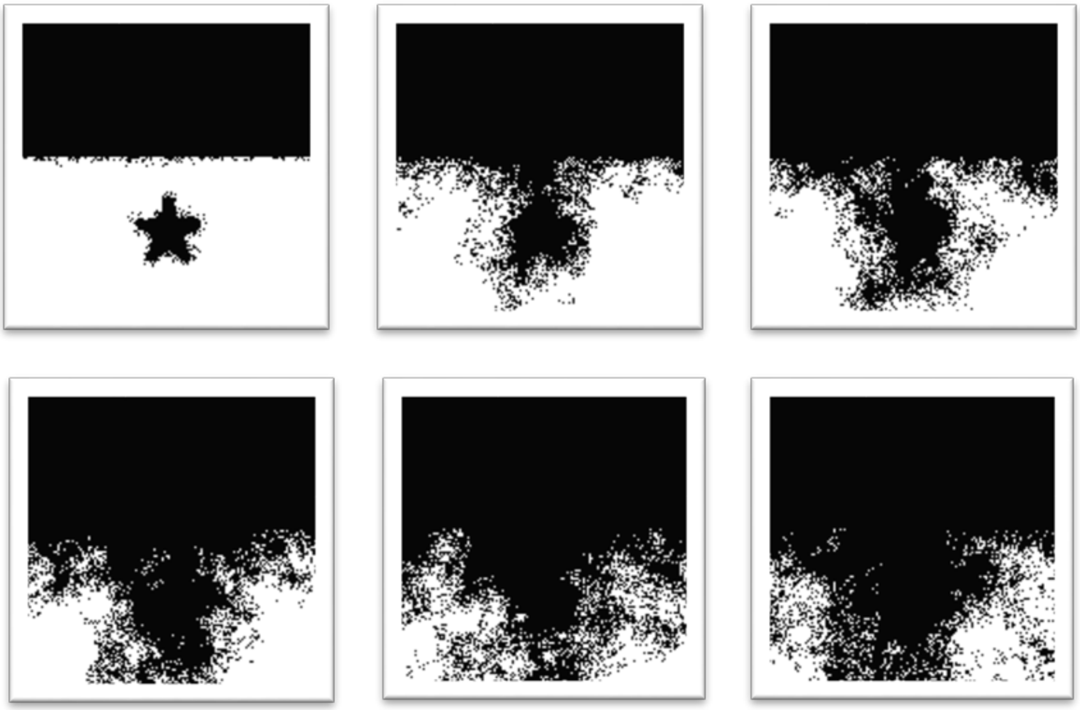
Implementing the word "test" in our simulation to observe how closed and semi-closed spaces become quickly entirely black (200'000 iterations between each picture)

As we can observe, the upper part of the "e" becomes pretty quickly fully black. The upper and bottom parts of the "s" and the bottom part of the "e" follow. Finally, the initially white space between the "t" and the "e" becomes black. Depending on the enclosed space, the closed spaces become more or less quickly black.

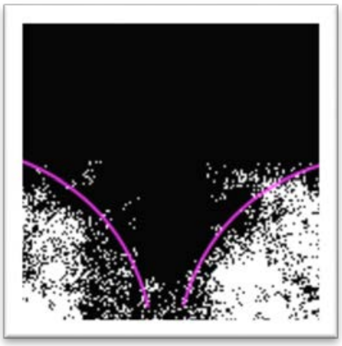
Example 4, semi-closed areas

The following figures enable us to observe a very similar evolution of the semi-closed spaces, which becomes and remains black.

Since we observe a very big source (more than half of the grid), the hole map will be fully black at one point.



Simulation using stubborn agents. (1’2 million iteration separate each figure)



Opinion map after 6’2 million iterations with marked resulting geometrical shape

Results of the strategy comparison

We tested the targeted vs. the untargeted approach on different sets of parameters. To encompass the every possible combination of parameters would have been too much. We set one configuration as “point zero”, from which we then varied one parameter at a time.

Our chosen point zero consisted of:

- a simulation with 400’000 iterations (every voter is updated on average 10 times)
- the probability density function $e^{(-5x\text{^}2)}$
- a maximum influence radius of 4

Other parameters, which were fixed in all simulations, were:

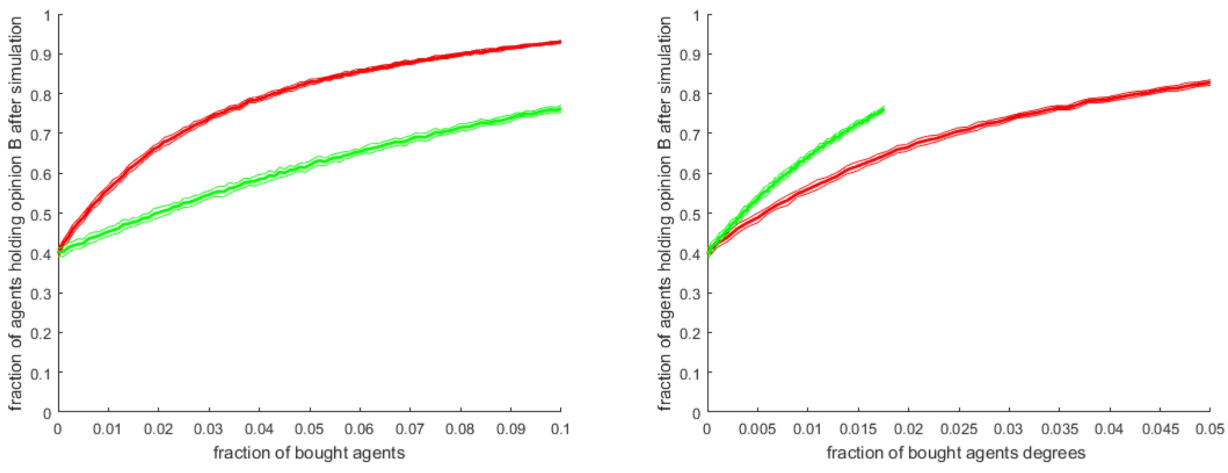
- Initially, 60% of the voters are for candidate A, 40% for candidate B (the buying was then done for candidate B), also the initial distribution is random and does not follow a pattern as before (but clusters of voters with the same opinion will quickly emerge).
- The grid size, 200x200

For each parameter set, we created two diagrams: the first showing the relation between the number of bought agents and the fraction of voters at the end of the simulation (one line for each approach) and the second one showing the relation between the sum of the outgoing degrees of the bought voters and the fraction of voters at the end of the simulation. This is because it is clear that the voters with higher influence radius are "worthier" in the sense that if you buy the same number of persons with the targeted approach as with the untargeted, the targeted approach will have a "better" outcome for the candidate. So we wanted to compare rather the two outcomes, where the sum of the bought agents outgoing degree was the same. The underlying assumption is that the more influence a voter has, the more it costs to buy him. We can only speculate if the cost-to-influence ratio is linear, exponential or anything else.

In all diagrams of this type, the red line is for the **targeted approach** and the green for the **untargeted approach**. Further, for every colour there are actually three lines. The thicker one in the middle is the mean over the ensemble, while the two smaller ones are the mean with the added and subtracted standard deviation respectively.

The green line in the right diagram stops in the air, because the simulation was run for a specified amount of bought voters, but the randomly chosen voters have the lesser sum of outgoing degrees. We could have cut the x-axis so that this would not happen, but we chose not to, in order to see how the red line proceeds.

Here are the results of the point zero simulation:



Simulation results for 400'000 iterations with a maximum influence radius of 4 and exponential PDF

As expected, when comparing the approaches for the same amount of bought voters, the targeted approach clearly wins. The steeper slope at the beginning (when buying only few voters) can be explained with the fact that the influence of each bought voter decreases, as the most influential voters are bought first.

Varying maximum influence

The maximum influence radius was varied from 2 to 5. We also computed typical network properties for the resulting network:

Maximum influence radius	1	2	3	4	5
Average outgoing degree	4	4.7243	6.7964	9.6818	13.221
Clustering coefficient	0	0.2108	0.41953	0.48361	0.5084

The clustering coefficient here refers to the definition of Watts and Strogatz (1998). It is the average of the local clustering coefficients and is sometimes also called the *transitivity* of the network. The local clustering coefficient of a node is defined as the number of edges between the neighbours of the node divided by the number of possible edges between the neighbours. In a directed graph the latter is n(n-1), where n is the number of neighbours. It is therefore a measure of how close the neighbours are to being a clique.

Also the maximum influence radius of 1 is included in the table only for reference. Recall that the minimum radius is also 1, which means that every agent has the same level of influence and the two approaches look exactly the same. (As a matter of fact, we did run the simulation for that and the untargeted approach performed better, which we found to be because the bought agents were better distributed over the network compared to the targeted approach, where they were all together.)

We implemented a function to compute the (average) path length too, but in the end it was nowhere near fast enough to finish in a day or two (speaking about one network configuration only). This is because there are roughly 1.6x10^9 pairs of nodes for which you need to find the shortest path.

The different diagrams for these simulations can be found in the appendix.

They look generally very similar to each other, except that with an increasing maximum influence radius, both approaches generated a steeper line. This could maybe be accredited to the idea, that in networks with higher transitivity, the opinions spread faster.

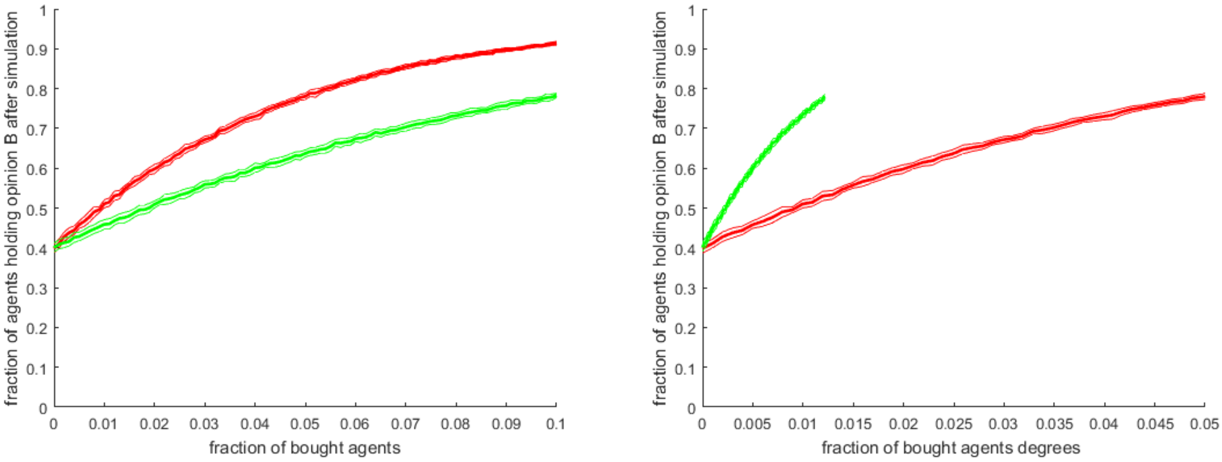
Varying number of iterations

The number of iteration steps was chosen from the set {100’000, 200’000, 400’000, 800’000}, where 400’000 is equivalent to our point zero.

We found nothing noteworthy about the differences in the results. The more time passes, the higher the fraction of voters for candidate B, regardless of the approach. Again, all the diagrams can be found in the appendix.

Varying PDF

As a second probability density function, we used a Gaussian (Normal) distribution. The network had an average outgoing degree of 20.1381 and a clustering coefficient of 0.5350.



Simulation results for 400'000 iterations with a maximum influence radius of 4 and normal PDF

On the right side, the angle between the two lines is significantly larger than it was with the exponential probability density function. This is probably not due to the more than double average outgoing degree, because we also had a double average outgoing degree in the max. influence radius 5 network compared to the max. influence radius 3 network where we didn't see such a difference between the angles. One interpretation could be that the ideal influence radius to target is neither extremely low nor extremely high, but somewhere in the middle.

The final implementation of the model to define the best strategy was implemented carefully using a toroidal shape to prevent dealing with any grid borders. It also shows and uses the variance and therefore stays statistically correct.

We see that very influential persons have a bigger impact on the simulation than most of the usual people. Since they reach out farther and potentially influence more people, the spreading of their opinion is faster than usually. This also shows the limits of this simulation since it doesn't show how much faster it can get.

Furthermore the model cannot take into account all real-life parameters, since we have to limit ourselves to the most obvious ones. For example, it doesn't take into account how difficult it can be to change an opinion. This is a very complicated parameter to implement. This model also counts the number of iterations but does not give any idea of the time it would need to get to the same state in the real world, since this is dependent on each country and each community.

Summary and Outlook

In short, we defined a model, observed what the model did, and used the model to simulate our two strategies. We found that for the same sum of degrees of the bought voters, the untargeted approach yields better results than the targeted. This means, if the cost-to-influence ratio was linear, a presidential candidate would be served better by not targeting the most influential voters. Furthermore, we proposed that voters somewhere in the middle of the influence distribution might have the greatest influence on the outcome compared to their degree. It would be interesting to examine this further.

A more fundamental research should consider adding stubborn agents to both sides of the opinion, since only one party having them is a very “unfair” or unrealistic state.

References

N. N. Alotaibi, *Media Effects on Voting Behavior*, European Scientific Journal Vol.9, No.20 (2013)

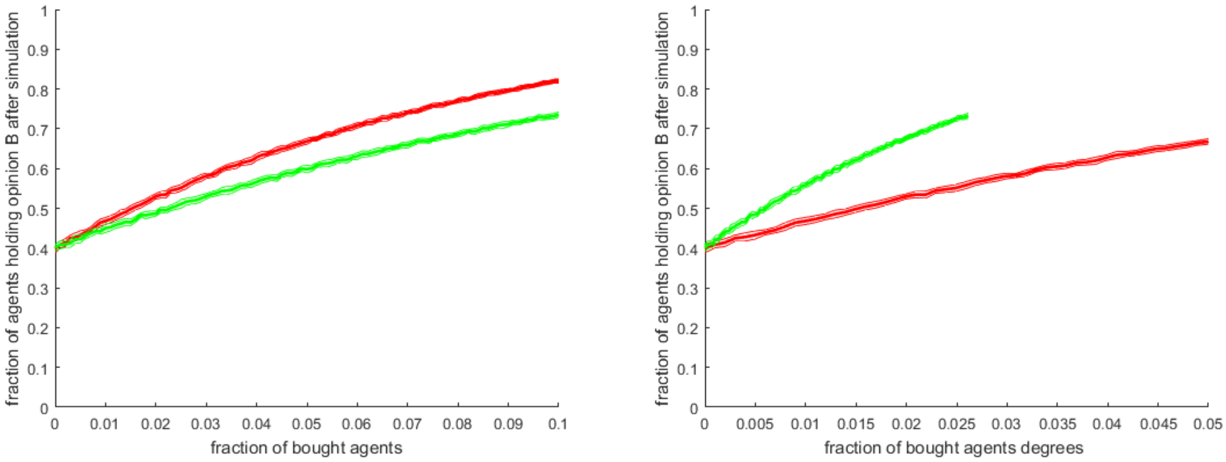
P. Clifford, A. Sudbury, *A model for spatial conflict*, Biometrika, Vol.60(3), pp. 581-588 (1973)

E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, A. Scaglione, *Binary Opinion Dynamics with Stubborn Agents*, ACM Trans. Econ. Comp. 1, 4, Article 19 (2013)

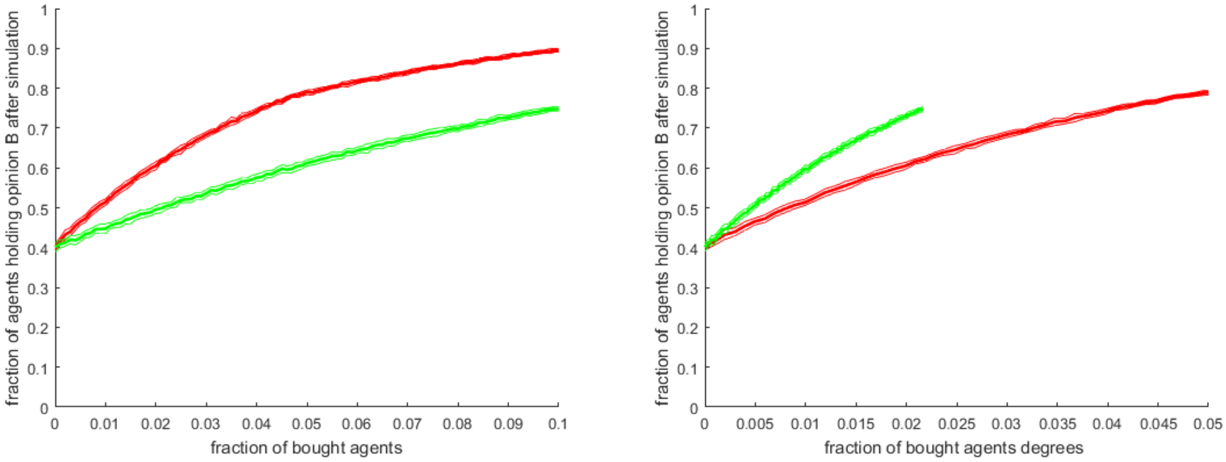
D. J. Watts, P. S. Dodds, Influentials, Networks, and Public Opinion Formation, Journal of Consumer Research Vol. 34, pp. 441-458 (2007)

D. J. Watts, S. H. Strogatz, *Collective dynamics of ‘small-world’ networks,* Nature 393, pp. 440-442 (1998)

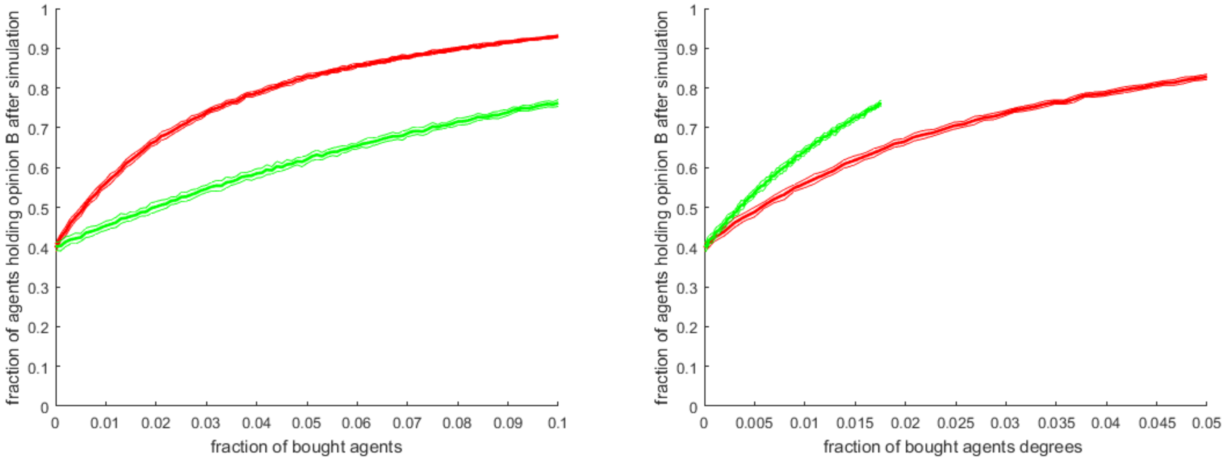
APPENDIX



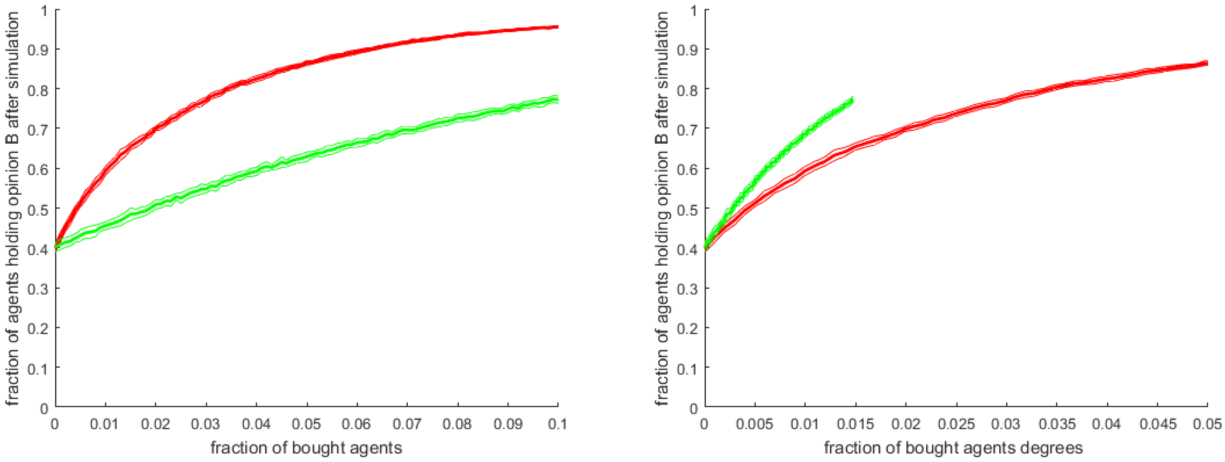
Simulation results for 400’000 iterations with a maximum influence radius of 2 and exponential PDF



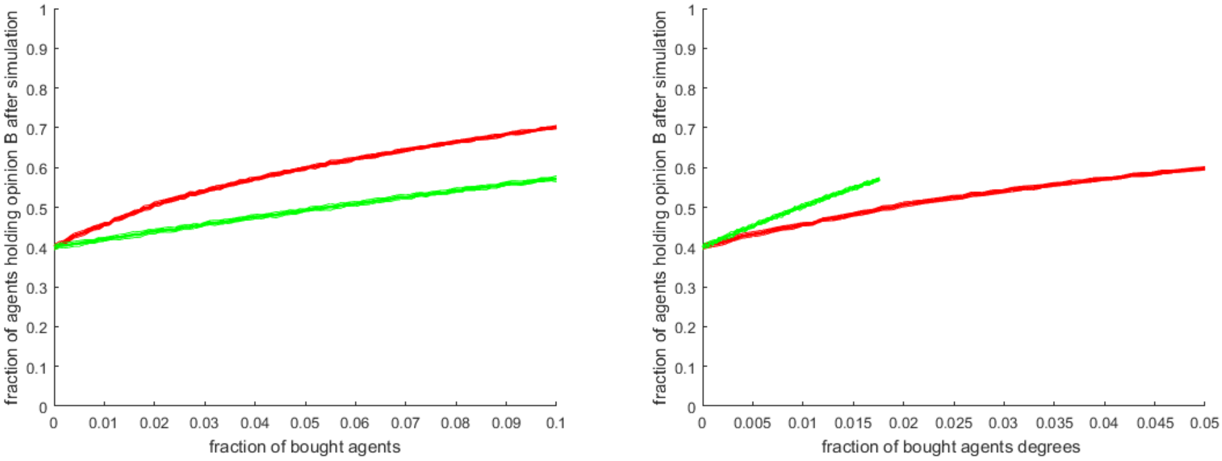
Simulation results for 400’000 iterations with a maximum influence radius of 3 and exponential PDF



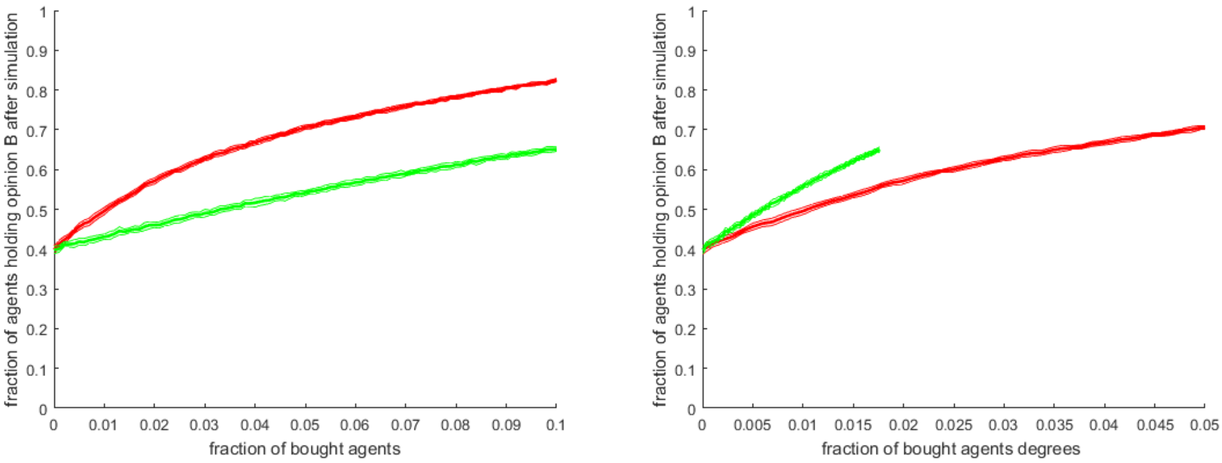
Simulation results for 400'000 iterations with a maximum influence radius of 4 and exponential PDF



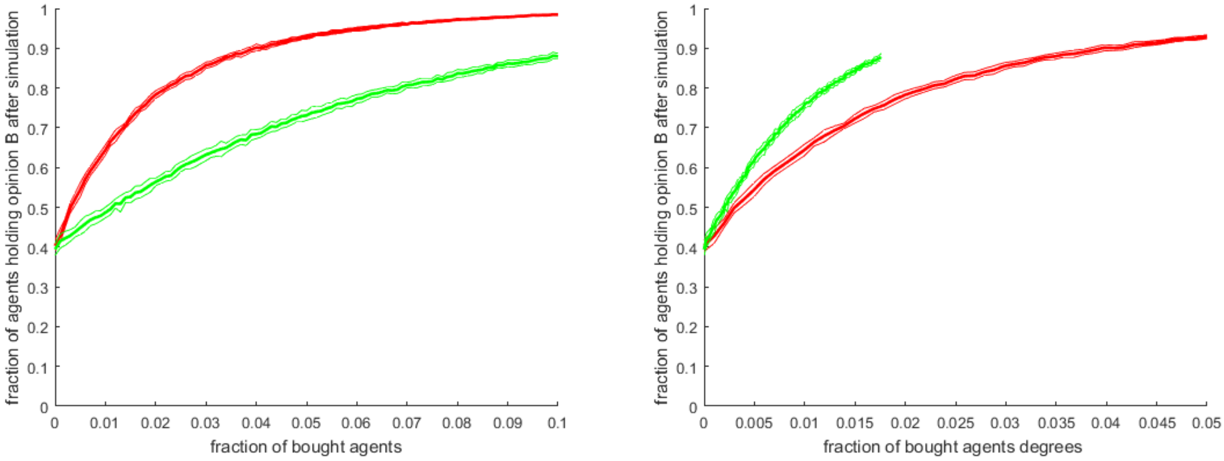
Simulation results for 400'000 iterations with a maximum influence radius of 5 and exponential PDF



Simulation results for 100’000 iterations with a maximum influence radius of 4 and exponential PDF



Simulation results for 200’000 iterations with a maximum influence radius of 4 and exponential PDF



Simulation results for 800'000 iterations with a maximum influence radius of 4 and exponential PDF