GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Bachelor's Thesis

Modellierung von Meinungsdynamiken in sozialen Netzwerken mit unterschiedlichen Topologien

Modelling Opinion Dynamics in Social Networks with different Topologies

prepared by

Vincent Christoph Brockers

from Neuss

at the Max-Planck-Institute for Dynamics and Self-Organization

Thesis period: 25th September 2022 until 2nd January 2023

Supervisors: Dr. Joao Pinheiro Neto, Riccardo Carlucci

First Rreferee: Prof. Dr. Stephan Herminghaus

Second referee: Prof. Dr. Stefan Klumpp

Abstract

Opinion dynamics in social networks like social media are of big interest for science, economy and politics due to its complexity and power of influencing people on a global scale. Data driven research already gained a lot of insights about platform specific user behaviour and showed the capability of social media having an impact on elections, radicalization or mental health.

There is a variety of platforms with different features and mechanisms that control the way content is displayed for its users. I want to compare the opinion dynamics inside social media platforms with a vote and ranking system like Reddit with a chronologically ordered system like Telegram by using a modelling approach. This has the advantage of reducing the dynamics to its core forces and being able to produce observables which might lack of real data. Using the knowledge of classical opinion models I proposed the application of a Ricker-wavelet as the underlying weight function for the opinion update rule. Implementing this into a network adapted to each platform made it possible to analyze their characteristics. In a vote and ranking system content of a certain opinion can be favored to gain significantly more views than the rest which may only be seen by a few users. Those which browse and vote new content also play a big role in this process. In a chronologically ordered system content of all opinions has the same probability distribution for the amount of views. Furthermore I discuss the advantages and dangers of both systems under different circumstances regarding the initial conditions.

Contents

1. Introduction

Without a doubt, we live in a digital age where the Internet has become irreplaceable in terms of research, news, communication, entertainment, shopping and in nearly every other aspect of our lives. Especially social media play a big role in all of these [1]. Whether it is for receiving the latest news, following our favorite stars, interacting with friends, discussing a topic in a bigger community or just watching funny videos, there is a big variety of social media platforms with lots of different features [2]. Today, many platforms have reached a user base on a global level and therefore are a hub for spreading information quickly between millions of people all over the world. It is important to be aware of the enormous influence on users opinions and perception of reality. Recent events like the US elections or the pandemic have shown the dangers of misinformation and radicalization on the Internet. In order to prevent such things, it is necessary to analyze and understand the mechanisms which lead to these phenomena [3].

In this bachelor's thesis I want to focus on a small part of this research. The main goal is to model and compare opinion dynamics in social networks with different structures, as seen in social media, using mathematical and computational tools. A modelling approach has the advantage of reducing a very complicated system on a few key aspects while still having the ability to gain knowledge about certain processes that occur in reality. Furthermore it could be interesting to look at scenarios that might lack of data.

In the first chapter I will talk about the basics of network science and how to model opinion dynamics in social networks. Looking at three different classes of one-dimensional opinion models, I will explain the general mechanisms and build the foundation for the following chapters. Next I present the model of social differentiation and discuss if or under what adaptations I proceed to use it for my purposes. The last step is to put the resulting opinion model on different network topologies. Here I want to compare the opinion dynamics inside two network models motivated by the social media platforms Reddit and Telegram which differ in the way of displaying information. Precisely I want to find out how the vote and ranking system of Reddit impacts the opinion dynamics compared to a chronologically ordered system like Telegram.

All model implementations and other code used for this project will be available on `https://github.com/VincentBrockers/bachelorproject`.

2. Basics

2.1. Network Science

A network is an interconnected or interrelated chain, group or system. In general, its entities are called *nodes* who are connected by *links* [4, p. 1]. In network science many different types of complex networks are studied, for example computer networks, transportation networks, biological networks or social networks. This thesis will focus on the latter. A network can be *directed* or *undirected* meaning that links either represent a certain direction between nodes or just a bi-directional connection. On top of that links can be *weighted* or *unweighted* in a sense that an associated weight represents a quantity, e.g. the influence of one person on another [4, p. 15]. Another thing that should be mentioned is the case when two different types of nodes occur, for example in social media the users and the posts they interact with. This is called a *bipartite* network [4, p. 17]. After defining those network properties one can start to analyze it further. There are a lot of metrics that can be used like the *average degree*, *average shortest path length*, *clustering coefficient*, *connectedness* or *node centrality*. Another way to do it, is by e.g. plotting the time development of the system, looking at the outcome for different initial conditions and comparing those developments with other metrics for social networks like the distribution of initial vs. final opinion, the amount of peaks or the standard deviation of the final opinion distribution and so on [5, p. 3].

2.2. Opinion Dynamics in Social Networks

For my research I am particularly interested in opinion dynamics, meaning how opinions of people in a social network spread, influence each other and thus develop over time. Describing and analyzing this process is very complex since each individual behaves differently in the way of perceiving and dealing with other opinions plus the interaction between people is strongly dependent on the network topology [6, p. 2]. With a huge variety of methods, models and assumptions a lot of insights were gained in this field since 1970 and still there is much to discover [7, p. 23]. In order to do so, one has to chose what small part of a big process can be described by a mathematical model and for what cases the assumptions hold to make predictions for the real world.

Models of opinion dynamics represent an opinion as a number or a set of numbers. Usually those models are divided into two categories, whether they use discrete or continuous opinions, which can be separated into other subcategories. Typically binary positions like buy/sell or Android/iPhone are represented by discrete opinions. A continuous opinion spectrum is used for example in representing political views from very progressive (-1) to very conservative ($+1$) [4, p. 200]. In the modelling process, each node of the network randomly (with respect to a wanted probability distribution) gets a number assigned from this spectrum. Then the opinions change over and over again as they get updated by a certain rule that depends on whether the interacting nodes are connected and what their opinion distance is (see section 2.3). Under the right conditions, the system eventually will reach a stationary state. Typically these states are *consensus*, *polarization* and *fragmentation*, who are characterized by their uni-, bi-, or multimodal final opinion distribution [4, p. 203].

2.2.1. Concepts from Psychology and Sociology

All of the described mechanisms are based one some important principles of psychology and sociology. To make everything reasonable one should not forget what the underlying forces for the given opinion dynamics are. Mainly the interaction and behaviour between people in social networks can be described by the following concepts [7].

- *social influence*: individuals change their behavior to meet the demands of a social environment (neighbors becoming more similar to neighbors) [8] [4, p. 205].
- *homophily*: tendency of individuals to associate and bond with similar others (similar nodes becoming neighbors) [9] [4, p. 205].
- *confirmation bias*: tendency to search for, interpret, favor, and recall information in a way that confirms or supports one's prior beliefs or values [10].
- *emotional contagion*: tendency to automatically mimic and synchronize behaviour and beliefs of another person, and consequently, to converge emotionally. [11].
- *xenophobia*: the larger the dissimilarity between two interacting individuals, the more they evaluate each other negatively, triggering repulsive behaviour [12].

2.3. Models of Social Influence

Due to the huge amount of opinion modelling studies in literature, it is not easy to classify every contribution. One approach of dealing with this problem was done by Andreas Flache et. al. [7]. Here the models were grouped together if their assumptions about social influence were implemented in a similar way namely with assimilative, similarity biased or repulsive influence. From a mathematical point of view the opinion update rule of those models is generally structured like the following form (2.1). The new opinion $o_{i,t+1}$ of an agent (node) i at the time step $t+1$ is calculated by

$$o_{i,t+1} = o_{i,t} + f_w(o_{i,t}, o_{j,t}) \cdot (o_{j,t} - o_{i,t}) \tag{2.1}$$

with $o_{i,t}$ the opinion of agent i at time step t, $o_{j,t}$ the opinion of agent j at time step t and the so called weight function f_w that scales the opinion change depending on both agents current opinions. This formalization is used for randomly picking two agents i, j and updating the first picked agents opinion if they are connected. Another way to update an agents opinion is by assuming the whole state of the network in fact every other opinion has an influence. This would be implemented by taking the sum of the (normalized by weights) second term in (2.1) over all agents j [7, p. 7]. It should be mentioned that the definition of f_w may vary by some small adaptions, still the mechanisms that arise in each model class are similar in the core.

2.3.1. Assimilative Influence

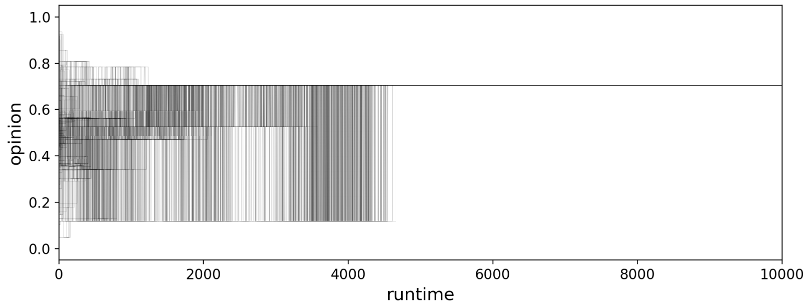
The first model category to look at is the one for models with assimilative influence. Its core assumptions are that individuals who are connected by a structural relationship always influence each other towards reducing opinion differences. This results in the weight function being defined as

$$f_w(o_{i,t}, o_{j,t}) = \mu \tag{2.2}$$

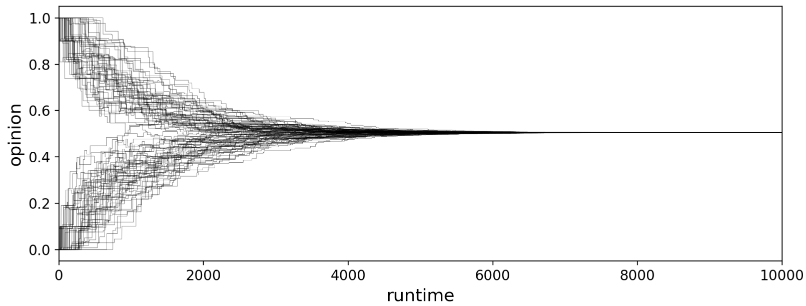
for all opinion distances with the parameter μ ($0 < \mu \leq 1$) that scales the strength of opinion convergence. Because of this property in the long run, if all agents in the network stay connected, the system inevitably will reach consensus [7, p. 7].

In the following figure 2.1 a few recreated exemplary simulations are shown to qualitatively understand the impact of the initial opinion normal distribution (mean 0.5, varying standard deviation σ), the network size N and the convergence parameter μ.

(a) $N = 100$, $\mu = 1$, initial opinion normal distribution with $\sigma \approx 0.2$.



(b) $N = 100$, $\mu = 0.1$, initial maximum polarization ($\sigma = 0.5$).



(c) $N = 1000$, $\mu = 0.1$, initial opinion normal distribution with $\sigma \approx 0.2$.

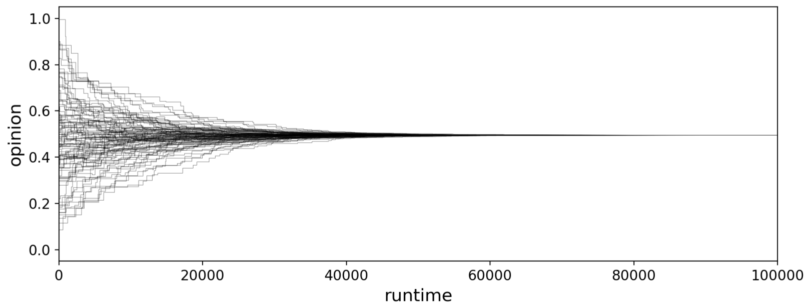


Figure 2.1.: Model with assimilative influence. Opinions of all agents dependent on runtime for a different network size N, convergence parameter μ and initial opinion distribution with standard deviation σ.

In the first plot 2.1a one can see that the opinion dynamics for $\mu = 1$ result in pretty drastic opinion changes with big jumps. If we look back at equation (2.1) with our given weight function this makes sense since the updated opinion $o_{i,t+1}$ from agent i is just set equal with the old opinion $o_{j,t}$ from agent j. After some iteration steps the formation of subgroups occur, meaning that groups of multiple agents adapted the same opinion. In the end there are only two subgroups left which always converge to one group of the opinion from one of the subgroups before, most likely the one with more agents.

Another side effect of the big opinion changes is that the final consensus opinion differs significantly from the mean value 0.5.

In contrast to that the second plot 2.1b shows that $\mu = 0.1$ instead leads to much smoother but a bit slower opinion dynamics due to smaller opinion changes. Furthermore it gets clear that for this model even maximum initial opinion polarization results in consensus since the weight function always causes attraction between agents.

The last plot 2.1c illustrates that on one hand the system behaves pretty identical for a different initial distribution, the smaller standard deviation just results in faster convergence for the same μ. On the other hand the runtime seems to scale linearly with the network size N as the behaviour for $N = 1000$ is the same as for $N = 100$ but with a runtime ten times longer. In both this and the former case the final opinion barely distinguishes from the mean value. Additional example can be found in figure A.1 in the appendix.

2.3.2. Similarity Biased Influence

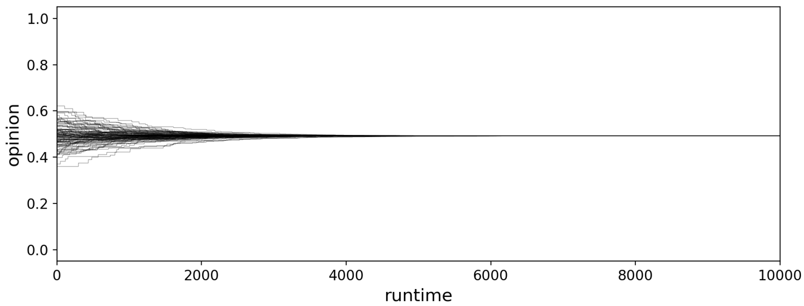
Models with similarity biased influence assume that only sufficiently similar individuals (depending on additional psychological mechanisms) can influence each other towards reducing opinion differences. Basically this leads to the former weight function but with limiting the opinion convergence as proposed in the new definition

$$f_w(o_{i,t}, o_{j,t}) = \begin{cases} \mu\ , & \text{if } |o_{i,t} - o_{j,t}| \leq \varepsilon \\ 0\ , & \text{otherwise} \end{cases} \tag{2.3}$$

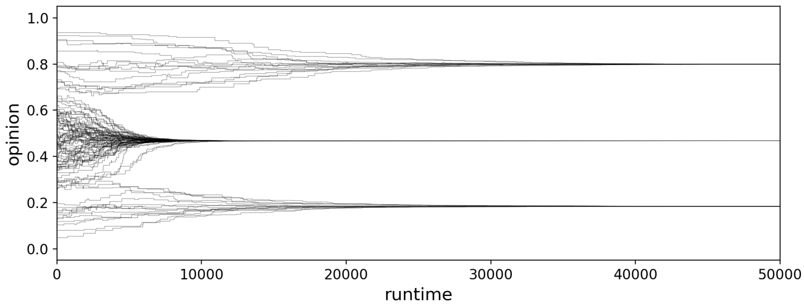
with the tolerance parameter ε that sets the threshold where the influence of another agent is cut off. Depending on the initial opinion distribution and the choice of ε the system can reach various final states. It is possible to reach either consensus, fragmentation or bi-polarization [7, p. 10]. Consensus can be achieved by setting the tolerance parameter big enough so the opinions of all agents will converge. Setting a smaller parameter will cause the appearance of opinion clusters because after some initial convergence time the agents form clusters in which they are stuck due to being outside the possible interaction range with agents from other clusters. Bi-polarization though is only reached via placing agents that have a fixed opinion at the borders of the opinion space so they will drag other agents towards their extremes, if the tolerance parameter is not set too small. Furthermore similarity biased influence models are so called bounded confidence models meaning the opinions of all agents never leave the initial opinion range [13].

As done for the former model class I will take a briefly look into the time development of the system for different initial conditions. The following figure 2.2 displays three exemplary simulation for $N = 100$ agents, $\mu = 0.1$ but varying ε and σ (for mean 0.5).

(a) $N = 100$, $\varepsilon = 0.15$, $\mu = 0.1$ and initial $\sigma \approx 0.05$.



(b) $N = 100$, $\varepsilon = 0.15$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.



(c) $N = 100$, $\varepsilon = 0.1$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.

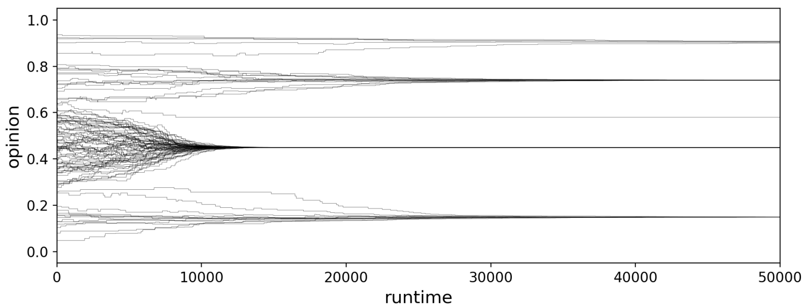


Figure 2.2.: Model with similarity biased influence. Opinions of all agents dependent on runtime for a different network size N, tolerance parameter ϵ, convergence parameter μ and initial opinion distribution with standard deviation σ.

The first plot 2.2a shows that consensus can be achieved by setting the tolerance parameter ε at a value where in combination with the initial opinion distribution the majority of absolute opinion distances between agents are below ϵ so the weight function basically is identical as in the former model class.

In the next plot 2.2b one can see what happens if the standard deviation is big enough to have sufficient agents with absolute opinion distances above ε resulting in the formation of opinion clusters, a final state of fragmentation. This happens as distant agents do not interact at all ($f_w = 0$) but rather converge with close agents to one group of one opinion that is too distant from other opinions to change as well.

Lastly, plot 2.2c emphasizes the impact of ε regarding the amount of opinion clusters. For smaller ε it is, with the same argumentation as before, possible for the system to form more groups of agents the same opinion in comparison to a system with equal σ. As seen in the timeline it is even possible for one agent to form a group alone since neither the agents above or below can attract him as every absolute opinion distance is greater than ε. Another mechanism which can be observed for systems resulting in fragmented opinion states is that the opinion dynamics are slower due to the fact that many of the interactions between agents are not opinion changes, in fact only those who have sufficiently close opinions.

In the appendix figure A.2 again shows that the speed of the opinion dynamics for same initial conditions is dependent on μ, as a value twice as big reduces the runtime for the same outcome by the half. Also the systems behaviour seems to scales linearly with the amount of agents N like in the previous model class.

2.3.3. Repulsive Influence

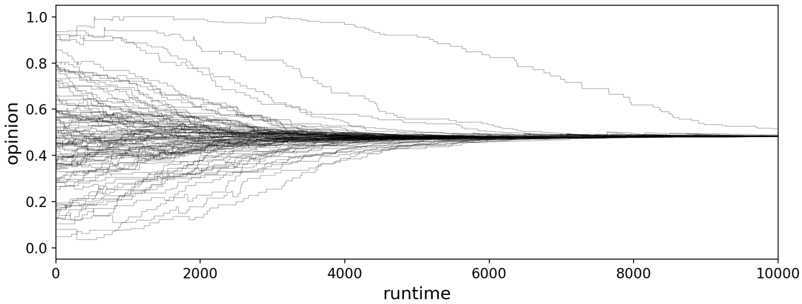
Lastly, models with repulsive influence assume that if individuals are too dissimilar (depending on additional psychological mechanisms) they can also influence each other towards increasing opinion differences. The weight function for models of this type can be defined as

$$f_w(o_{i,t}, o_{j,t}) = \mu \left(1 - \frac{1}{\varepsilon} \left|o_{i,t} - o_{j,t}\right|\right) \ . \tag{2.4}$$

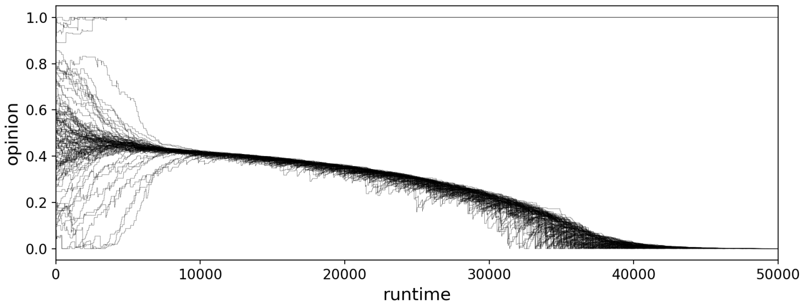
In this form the tolerance parameter ε sets the threshold whether the opinion of the agent gets reinforced for absolute opinion distances below or rejected for a distance above it. Furthermore not only the sign of μ will be decided, the further the absolute opinion distance varies from the threshold, the bigger the influence in the resulting direction. Due to this characteristic models with repulsive influence behave in a very flexible way. It is possible to realize consensus, fragmentation and bi-polarization even without placing stubborn extremists and for many initially moderate agents just as a result of the repulsive behaviour. Without truncating the opinions at some value or cutting agents connections it would be possible for the opinions to leave the initial range. Thus such models are called unbound confidence models as well [13].

The following figure 2.3 again shows some exemplary simulations for $N = 100$ agents, $\mu = 0.1$ but varying tolerance parameter ε and standard derivation σ. To keep the initial opinion range I just used a simple truncating rule. Whenever an agent would cross the upper or lower border according to the opinion updating rule, the opinion is just set to the borders value 0 or 1.

(a) $N = 100$, $\varepsilon = 0.55$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.



(b) $N = 100$, $\varepsilon = 0.48$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.



(c) $N = 100$, $\varepsilon = 0.15$, $\mu = 0.1$ and initial $\sigma \approx 0.05$.

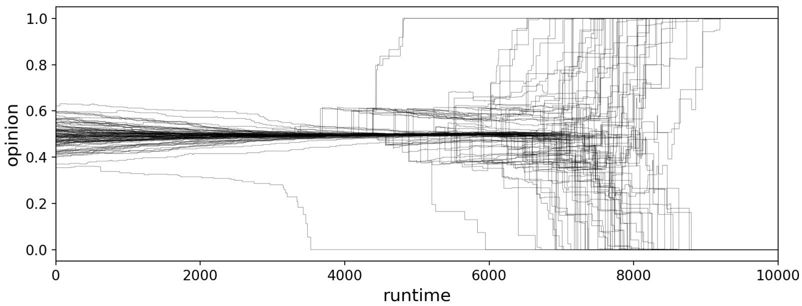


Figure 2.3.: Model with repulsive influence. Opinions of all agents dependent on runtime for a different network size N, tolerance parameter ϵ, convergence parameter μ and initial opinion distribution with standard deviation σ.

As for the two previous model classes it is possible to reach consensus, as seen in plot 2.3a, by setting the tolerance parameter ε big enough to ensure as many attractions between agents as possible for the given initial opinion distribution. Even though repulsive opinion changes can happen, if no agent has a greater absolute opinion distance than ε regarding the convergence point, consensus is unavoidable.

In the next plot 2.3b one can see what happens, if this would be the case. A slightly smaller choice of ε causes that even a few agents who were pushed to the upper border in the beginning are always being rejected by the majority which as a consequence of that slowly move to the other border until a state of bi-polarization is reached.

In plot 2.3c one can see another phenomenon if both the initial standard deviation σ and ε is pretty small. At first it seems like consensus will be achieved but after a while the one agent that went to one border pushes a few other agents to the opposite border such that in the end all agents are rejected until a final polarized state. In the appendix figure A.3 additionally shows how a final polarized state can be reached immediately and how extending the opinion range leads to faster dynamics due to greater weights.

The following figure 2.4 shows a measurement of the convergence time for a final state of consensus to check the observed scaling with N and μ.

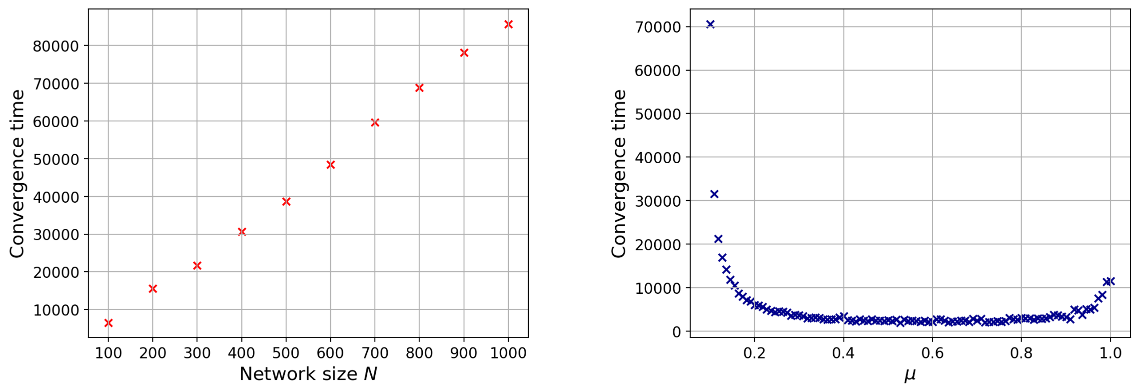


Figure 2.4.: Convergence time for a final state of consensus depending on the network size N and μ. A state is considered stable, if all agents have an absolute opinion distance below 10^{-3} to the mean of the current opinion distribution.

At least for this range of N, there definitely is a linear relation. As only one pair of agents interacts per iteration, larger networks need more time to converge. The convergence time for μ decays exponentially before increasing again. For small values there are smaller opinion changes that slow down the dynamics. Increasing μ therefore speeds things up again. But if μ gets too big, the opinion changes are big jumps which prevent a quicker convergence.

3. The Model of Social Differentiation

Since only models with repulsive influence can reach bi-polarization from various initial conditions, which is especially seen in social media [14, p. 29], I also chose one to work with, namely the model of social differentiation [15, p. 76]. The core assumptions are motivated again by the concepts explained in section 2.2.1.

3.1. Model Properties

As seen before, each member of a population is represented by an agent i that holds the opinion $o_{i,t} \in [0, 1]$. During every time step t of the simulation the program randomly picks one of the N agents and updates its current opinion as

$$o_{i,t+1} = o_{i,t} + \frac{\sum_{j=1}^{N} f_w(o_{i,t}, o_{j,t}) \cdot (o_{j,t} - o_{i,t})}{\sum_{j=1}^{N} f_w(o_{i,t}, o_{j,t})} + \xi_{i,t} \tag{3.1}$$

with the weight function

$$f_w(o_{i,t}, o_{j,t}) = \begin{cases} \left(1 - \frac{1}{\varepsilon} \cdot |o_{i,t} - o_{j,t}|\right)^a, & \text{if } |o_{i,t} - o_{j,t}| \leq \varepsilon \\ -\left(\frac{1}{\varepsilon} \cdot |o_{i,t} - o_{j,t}| - 1\right)^a, & \text{if } |o_{i,t} - o_{j,t}| > \varepsilon \end{cases}. \tag{3.2}$$

and the parameter

$$\xi_{i,t} = \mathcal{N}\left(0,\ \frac{s}{N} \cdot \sum_{j=1}^{N} e^{-|o_{i,t} - o_{j,t}|}\right) \tag{3.3}$$

which implemented to add a bit of noise and drive the agent away from his current opinion as a mechanism of “striving for uniqueness” [15, p. 77]. The factor is determined by drawing a value from a normal distribution with mean 0 and the standard deviation $s/N \cdot \sum_{j=1}^{N} e^{-|o_{i,t} - o_{j,t}|}$ that gets bigger the more agents have a low opinion distance regarding the currently viewed agent. The factor s/N is for scaling and normalizing purposes. In the weight function the tolerance parameter ε occurs again to set the critical opinion distance at which the sign switches plus the exponent $a > 0$ appears to vary the shape of the weight function [15, p. 76]. The following figure 3.1 highlights the impact of this exponent a.

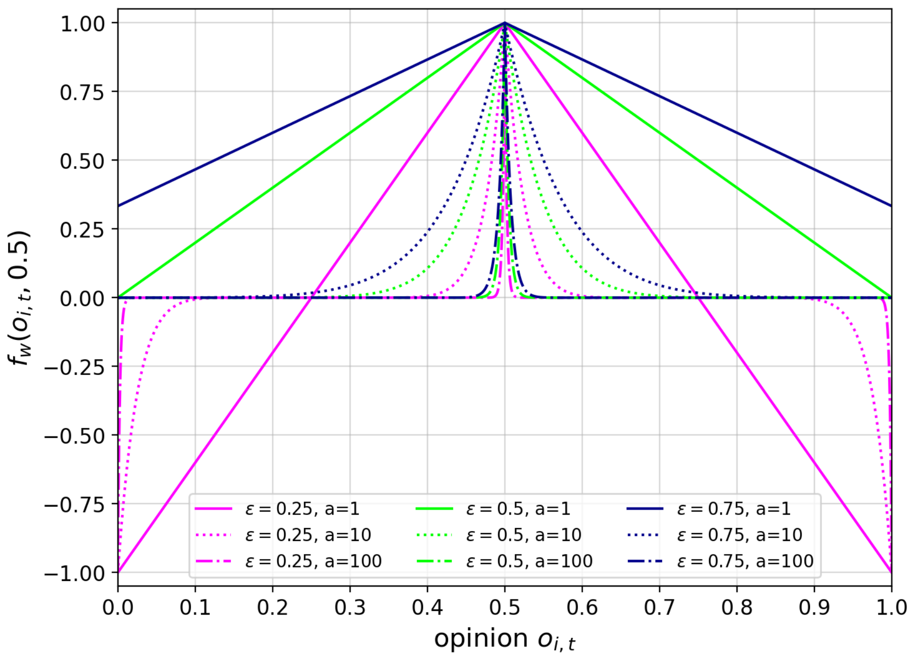


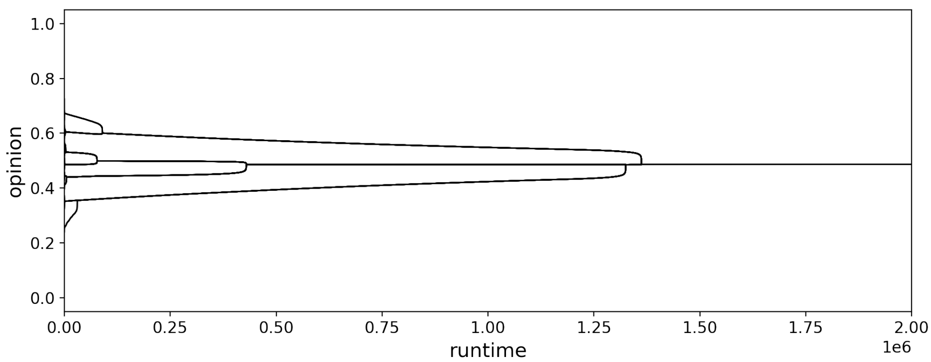
Figure 3.1.: Shape of the weight function f_w depending on the agents opinion $o_{i,t}$ for different values of the tolerance parameter ε and exponent a regarding another agent with fixed opinion $o_{j,t} = 0.5$.

First of all, the function is always point symmetrical around its roots $o_{j,t} \pm \varepsilon$. For $a = 1$ the weight scales linearly to the opinion distance. For bigger a only opinions very close (or far, if negative values are allowed regarding ε) to the other agent result in weights that are not approximately zero. In fact there is an exponential decay to the left and right in the area of close opinions. Next there is a region with values close to zero which gets bigger for greater a. Due to the point symmetry after the root is reached, the function follows the same curve but mirrored to the x-axis. Note that this symmetry only holds for opinions which are at max 2ε away from $o_{j,t}$. For bigger distances the negative weight exceeds the absolute value of the positive maximum. This is important to keep in mind when choosing the parameters and truncating values as otherwise there is the possibility of very strong repulsion between agents.

3.2. Reproduction of Past Results

In this section I want to present my reproduction of the most important past results for the described model. At first I will take a look at two cases in figure 3.2 without "striving for uniqueness", so $s = \xi_{i,t} = 0$ for all i, t, for a fully connected network with $N = 100$, $\varepsilon = 0.5$ but varying initial standard deviation σ for a normal distribution with mean value 0.5. The exponent of the weight function was set to $a = 100$. After that figure 3.3 illustrates three cases where the previous conditions are only changed in $s = 0.025$ and $\varepsilon = 1$ so with "striving for uniqueness" but only for attractive opinion changes. As done before, the opinions are simply truncated at the according border.

(a) Initial standard deviation $\sigma \approx 0.1$.



(b) Initial standard deviation $\sigma \approx 0.3$.

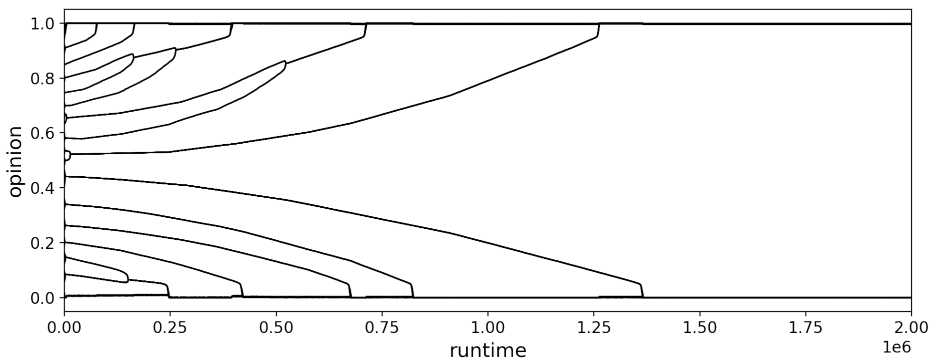
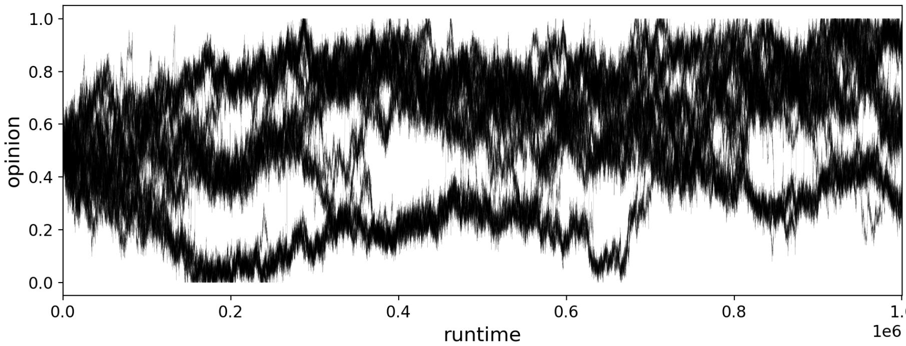
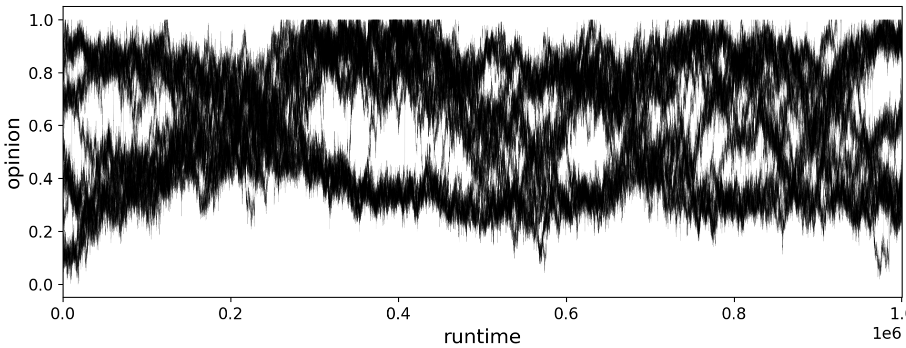


Figure 3.2.: Model of social differentiation. Opinion timeline for different initial distributions with $N = 100$ agents and parameters $\varepsilon = 0.5$, $s = 0$, $a = 100$.

(a) Initial standard deviation $\sigma = 0$.



(b) Initial standard deviation $\sigma \approx 0.3$.



(c) Initial standard deviation $\sigma = 0.5$.

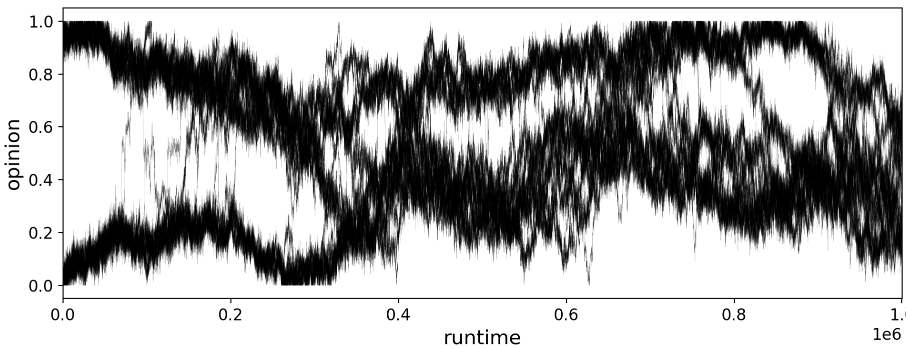


Figure 3.3.: Model of social differentiation. Opinion timeline for different initial distributions with $N = 100$ agents and parameters $\varepsilon = 1$, $s = 0.025$, $a = 100$.

The first two plots 3.2a, 3.2b for $s = 0$, $\varepsilon = 0.5$ show a known behaviour at first glance. For a small initial standard deviation even with allowed repulsive opinion changes consensus can be reached in the end. If in contrast to that the standard deviation is set higher, the majority of interactions are rejections so a bi-polarized state will be reached. The new phenomenon observed is the formation of branches consisting of very distinct subgroups which merge to new or existing branches that converge to the final state. The reason for that is the value $a = 100$ as discussed later. Another interesting observation is that in the case of bi-polarization the final branches are not exactly converging towards the maximal/minimal opinion. This makes sense since even opinions initially sitting at the border will be dragged towards the center sometimes. In the end the two final branches have an absolute opinion distance above $\varepsilon = 0.5$ which makes the system stable.

Adding the mechanism of "striving for uniqueness" results in a completely new behaviour as seen in figure 3.3. During all times visible fluctuations within a certain range $\xi > 0$ occur on top of the initial opinion changes. Besides that, it is also possible to declare the formation of subgroups within branches that eventually wander into others but are never stable as a consequence of fluctuations. It seems like there are time intervals in which the current state of the system is especially chaotic or ordered in a sense of less or more distinguishable branches with bigger spaces between them.

3.3. Model Conclusions

Before I continue applying an opinion model on more complicated network structures, I want to add a few comments about the recent insights and discuss how to proceed the modelling process. The first thing I want to mention is that I limit myself to one dimensional models as they are very intuitive to understand, still produce various outcomes with complex dynamics and have a faster computational time.

In the comparison of the three model classes I showed how the network size N and the convergence parameter μ scale to the runtime, while the tolerance ε is the parameter mainly determining the outcome besides the initial distribution. There may be cases where μ is also relevant for the outcome due to the scaling of opinion changes which has to be considered in further analysis. Lastly it gets clear that models with repulsive influence should be used in context of social media since we especially observe very polarized communities and platforms there and such models deliver the most flexible dynamics and possibilities for adaptions as seen in the model of social differentiation.

This model shows how the formation of very distinct subgroups can arise. A look at the exponent $a = 100$ explains this as the weight function has a really steep curve such that only very close and very distant agents will change their opinion significantly. When the standard deviation is small enough only the first effect is relevant and the formed subgroups converge to consensus. A bigger standard deviation makes the second effect relevant as well and the subgroups converge to either one border until a bi-polarized state is reached. One thing that should be mentioned here is that choosing the exponent this big really limits the interaction range between agents. This might be helpful to explain the formation of subgroups in small world networks which are fully connected but is probably not the best choice when looking at bigger networks like social media as the network structure also has an impact on the formation of those groups.

Adding the mechanism “striving for uniqueness” was done as an approach to model the fact that when humans in a bigger group perceive that they are very similar to others, they tend to change their opinion towards becoming more unique and individual. If this assumption holds as well for users in social media has to be determined later on. Nonetheless varying noise on every opinion change shows very interesting characteristics. It turns out that fluctuations reduce the sensitivity of the system regarding the initial opinion distribution. For both extreme cases, initial consensus and polarization, the dynamics transition into a similar behaviour of forming and dissolving subgroups within a certain “tube” of opinions that fluctuates around itself repeatedly as the system can never reach a stable state.

Furthermore it occurred to myself that the model initially was implemented with setting values of the weight function to -10^{-5} or 10^{-5} if it would return values closer to zero otherwise to ensure weak interaction is not falsely seen as no interaction by the program due to floating point inaccuracies. The problem with that is the creation of a new invisible hyperparameter that could change the dynamics which in fact can be observed as seen in figure A.4a as I left out this limitation and ran the simulation for the same initial conditions. The system shows a behaviour that can not be explained by statistical fluctuations only as there are only two branches left that do not have a flux of agents between them. Another little modification I noticed is that the quotient in the opinion update rule needs to be limited for the case of initial polarization as it can get really big for opinions only beeing set to 0 or 1. Without such limitations one group will quickly move to the other border as figure A.4b shows. So for recreating 3.3c I only allowed $|f_w| \leq 0.075$. In general this showed me to be more careful of artificially changing the dynamics with small code adaptions and invisible hyperparameters and also to differentiate between statistical fluctuations or those hidden mechanisms.

4. Opinion Dynamics in Networks with a Social Media like Structure

Until this point I only applied the opinion model on a fully connected network with one type of nodes. From now on as I want to compare opinion dynamics in networks with a social media like structure, this will change. My main idea for the whole modelling process was to assume that there are two type of nodes, users and posts. An user represents a human browsing the social media, creating content (posts) and getting influenced by the content from other users. So the posts are the digital representation of ones opinion. The advantage of doing this is that I can still apply the former opinion model and also create mechanisms to control the way users behave on the network, how they create posts and how the see posts. I will further explain this in sections 4.2 and 4.3 for each network archetype. The important thing to keep in mind is that I only look at user to post interaction for the following parts.

4.1. Adaptations to the Opinion Model

For the applications of the model on networks with social media like structures I thought of two ways adding a few modifications to the weight function f_w. This was motivated by the fact that the current model allows infinite repulsion if not being truncated. Instead I thought of rather disallowing any interaction if the absolute opinion distance is too big. With this idea the new definition is given by

$$f_w(o_{i,t}, \tilde{o}_j) = \begin{cases} \mu \left(1 - \frac{1}{\varepsilon} |o_{i,t} - \tilde{o}_j|\right)^a, & \text{if } |o_{i,t} - \tilde{o}_j| \leq \varepsilon \\ -\frac{\mu}{r} \left(\frac{1}{\varepsilon} |o_{i,t} - \tilde{o}_j| - 1\right)^a, & \text{if } \varepsilon < |o_{i,t} - \tilde{o}_j| \leq \Omega \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

including a new parameter $r > 0$ to scale the repulsive term if wanted and the interaction width $\Omega > \varepsilon$ to ensure that there is a certain point where an extreme opinion stays and can not be interacted with. I also introduced the convergence parameter μ again to ensure smoother opinion changes. Note that $\tilde{o}_j$ now represents the opinion of the post j which is constant over all times as once a post got created the opinion stays fixed and can not be changed at some point. The shape of this function is basically like the one in figure 3.1 but with a hard cut to zero at the interaction width Ω.

The second idea was motivated by wanting the weight function to be fully continuous and differentiable in order to not have a hard cut off at the interaction width or a hard turn at zero but rather a smooth slope. One function which fulfills this and has the wanted shape is the Ricker-wavelet

$$\psi(t) = \frac{2}{\sqrt{3\sigma}\pi^{1/4}} \left(1 - \left(\frac{t}{\sigma}\right)^2\right) \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

which is basically the normalized second derivative of a Gaussian function and known for its use in numerical analysis [16]. To apply it as a weight function, the argument t has to be exchanged with the absolute opinion distance, σ with the tolerance parameter ϵ, as $\pm\sigma$ are the roots of the function where the sign has to switch, and the normalization factor with the convergence parameter μ, the maximal amplitude. This results in the new definition

$$f_w(o_{i,t}, \tilde{o}_j) = \mu \left(1 - \left(\frac{|o_{i,t} - \tilde{o}_j|}{\varepsilon}\right)^2\right) \exp\left(-\frac{|o_{i,t} - \tilde{o}_j|^2}{2\varepsilon^2}\right) . \tag{4.2}$$

The shape of this implementation is seen in the following figure 4.1.

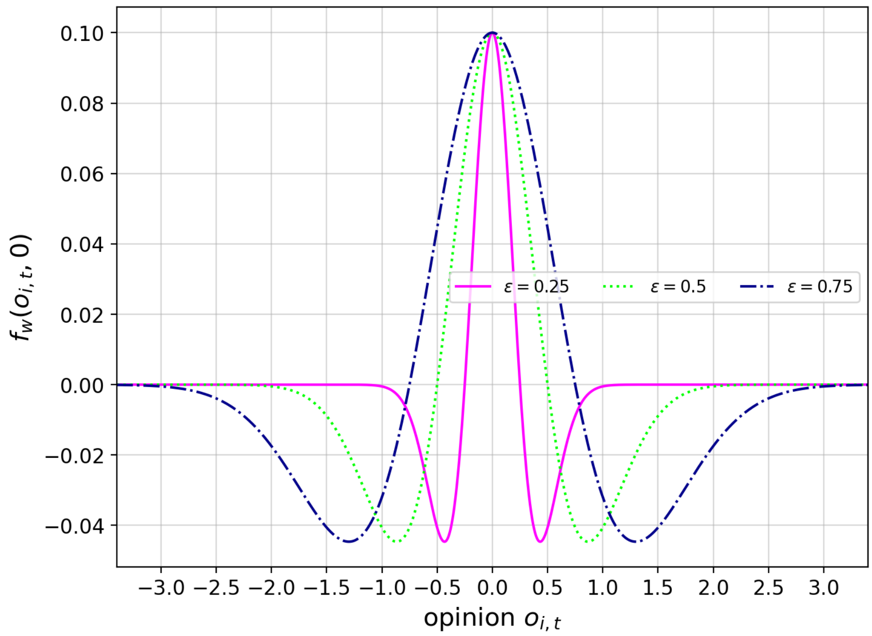


Figure 4.1.: Shape of the Ricker-wavelet weight function dependent on the user opinion $o_{i,t}$ for different values of ε with fixed $\mu = 0.1$ regarding a post with opinion zero. Bigger ε stretch the function along the x-axis.

One important thing to mention is that with this weight function it is not necessary anymore to limit the opinion space as it tends to zero for larger opinion distances. The width of range with positive values is 2ϵ with its maximum at $|o_{i,t} - \tilde{o}_j| = 0$ and a value of μ. On both the left and right of this range the function only has negative values with the minima at $\pm\sqrt{3}\varepsilon$ and a value of $-2\mu e^{-3/2} \approx -0.45\mu$. For opinion distances above/below $\pm 4\varepsilon$ the function is approximately zero. Thus the width of range with negative values (relevant for the dynamics) on both sides is ca. 3ε.

To sum it up, the Ricker-wavelet weight function returns repulsive values in a range three times as big as for attractive values but the maximum absolute value for the attractive ones is about as half as big. If I proceed to use this opinion model, only under the assumption that repulsive behaviour of users is weaker than attractive one, but happens in a wider range.

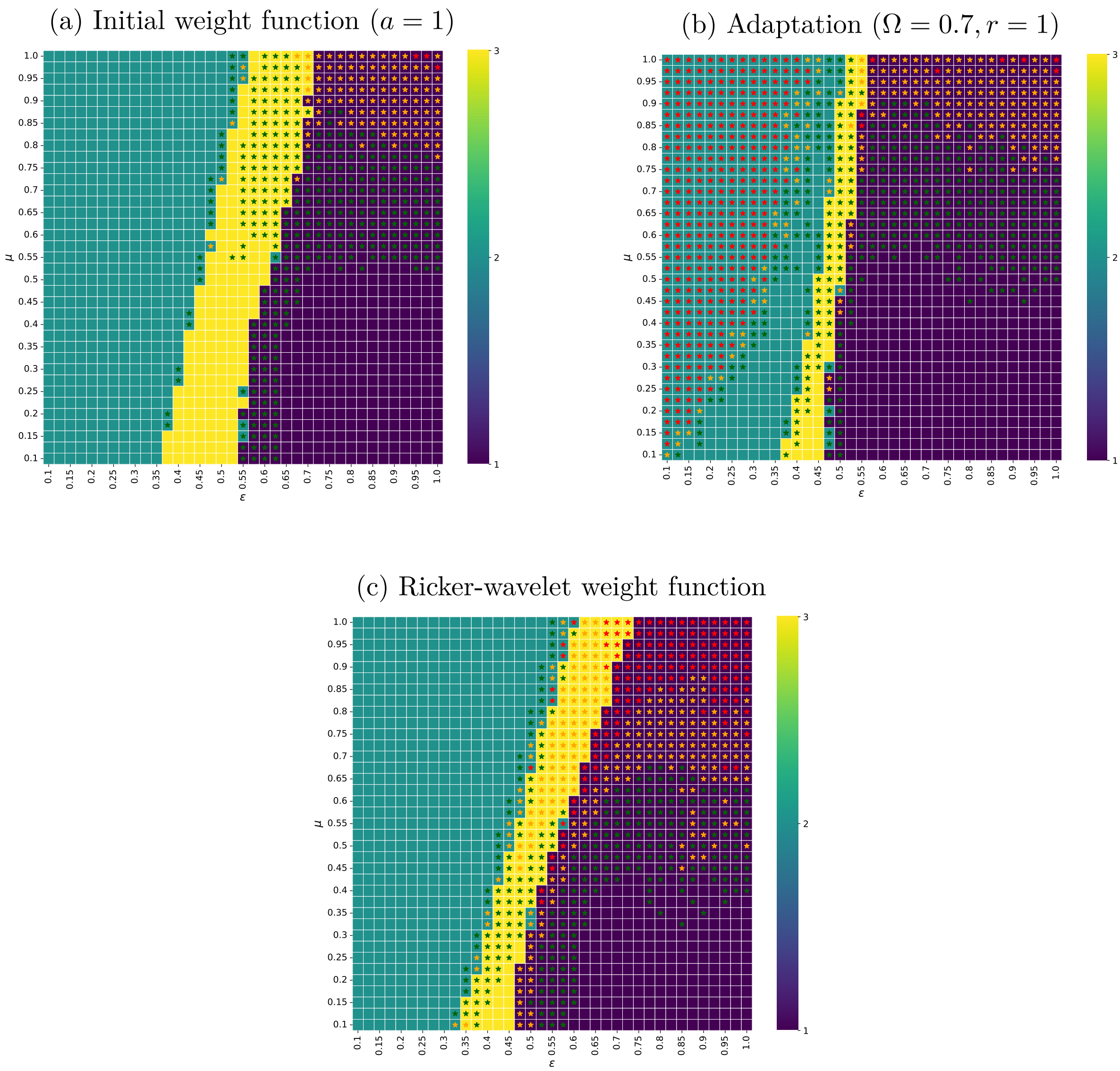
4.1.1. Comparing Final States

To get a quick overview if both manipulations of the weight function are still able to produce all wanted final distributions (consensus, fragmentation, bi-polarization) I developed a small program which lets each opinion model run for 10^7 iterations for the same initial distributions but different parameter combinations (ε, μ). In fact both the initial user and the initial post opinion distribution were set to a beta distribution[1] with $\alpha = \beta = 10$. The opinion updating works exactly like explained before in section 2.3 with the only change that I look at user to post interaction and the post distribution stays fixed all the time. Afterwards the algorithm counts the amount of peaks for each final opinion distribution and calculates how many users are not within a certain range of either one of these peaks. This mechanism is a quick measurement for the stability of the current system because if they are many users between the peaks, it is clear that there are still wandering around and either form a peak on their own or also get repelled from the posts opinions. When most or all users are nearby the peaks, the system is more stable as any interaction with a post does not lead to a big opinion change. They reached an opinion distance regarding most posts in which the weight function becomes close or equal to zero. The other option is that they are so close to the posts opinions which according to the general opinion update rule (2.1) also leads to almost no opinion change. The implementation and further explanations of the peak finding and stability

[1] Initially the beta distribution returns a random variable $B \in (0, 1)$. Since I need a random variable $\tilde{B} \in (-1, 1)$, I calculate $\tilde{B} = 2B - 1$. The beta distribution is used to be more flexible in the shape of an initial distribution as it can be a bi-modal, uni-modal or uniform distribution.

check algorithm can be found in the appendix B.

In the following part I will present the results of this procedure for each newly defined weight function and for the initial one in form of three heatmaps as seen in figure 4.2 and some corresponding final distributions as seen in figure 4.3.



(a) Initial weight function ($a = 1$)

(b) Adaptation ($\Omega = 0.7, r = 1$)

(c) Ricker-wavelet weight function

Figure 4.2.: User to post interaction, same initial beta distribution ($\alpha = \beta = 10$). Amount of peaks for different parameter combinations (ε, μ) after 10^7 iterations for three different implementations of f_w. Red/Yellow/Green star indicates that $(\geq 1000)/(999-500)/(99-0)$ users are not within a range of ± 0.25 around either one peak in the final state. All heatmaps show a similar pattern for the amount of peaks but differ in their regions of stability.

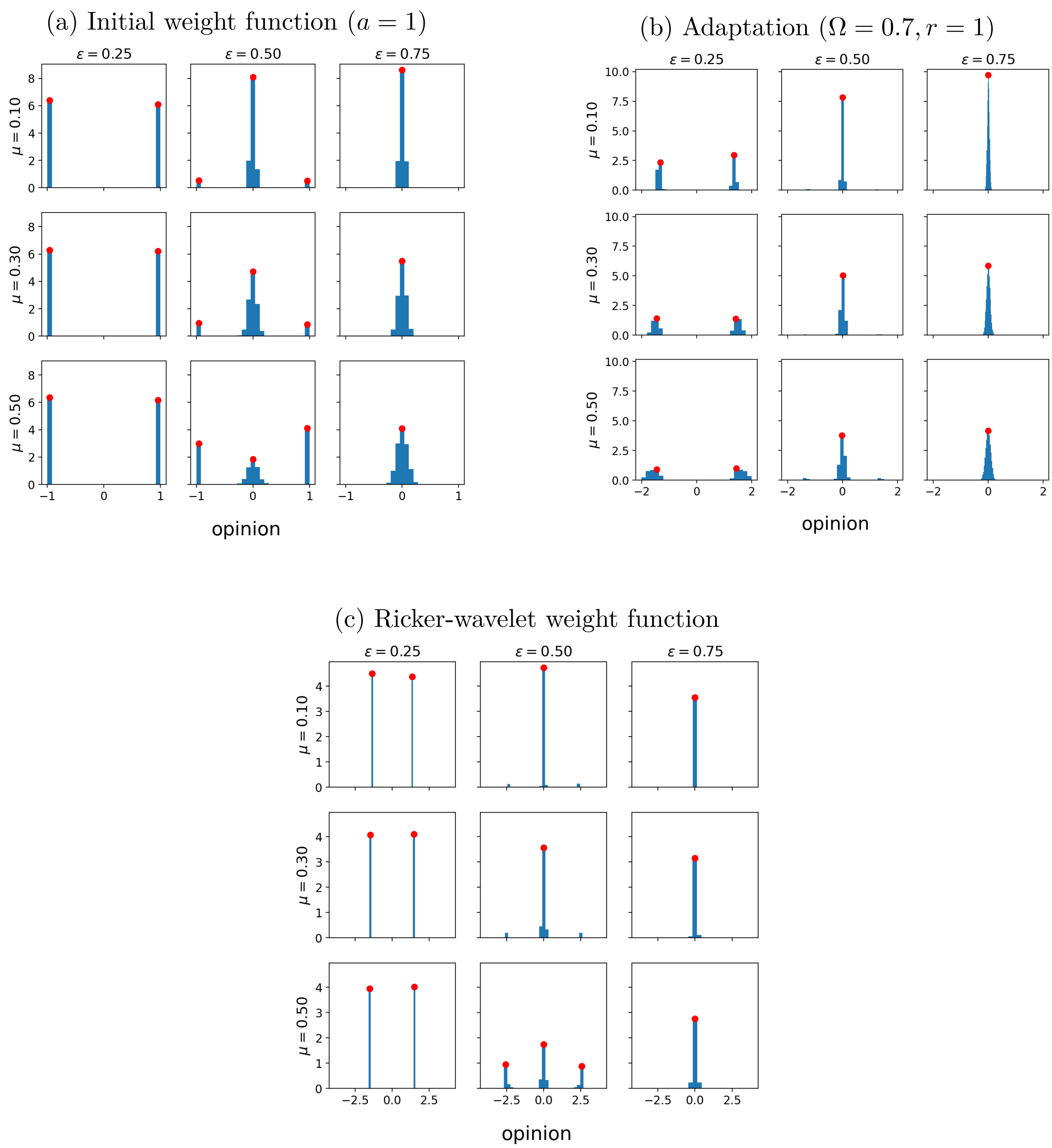


Figure 4.3.: User to post interaction, same initial beta distribution ($\alpha = \beta = 10$). Final distributions for nine different parameter combinations (ε, μ) after 10^7 iterations for three different implementations of f_w. Each row shows the transition of states with different amount of peaks. On top of that the peak width already is a hint for the different stability of each same state for each function.

The first observation when looking at all three heatmaps is that they all approximately share the same regions for the final amount of peaks. For smaller values of ε, there are two peaks for all values of μ. At $\varepsilon \approx 0.35$ a third peak emerges, but only for small values of μ. The bigger μ the higher ε has to be in order for three peaks to occur. This transition could also be described with a linear function.

After the region of three peaks for even higher $\varepsilon \approx 0.5$ there is only one peak left. The transition here follows the same linear slope as the one before. Looking at the final distributions makes sense of this phenomenon. For low ε two peaks emerge as a consequence of predominantly repulsive opinion changes until the two groups are sufficiently distant from the posts opinion (which distributions stays fixed). Increasing ε allows more attractive opinion changes so more users stay in the middle but the majority still wanders to the outside. After some point the middle peak reaches the height to be classified as a third peak. The bigger ε gets from now on, the more the two outlying peaks shrink and the more the middle one grows up until the point where the outside peaks are not classified as two peaks anymore.

At this transition it could happen that the peak classifier still identifies one outlying peak but not the other. This is the case when their height is so close to the classifiers threshold and one is above it while the other is below. One can see this in the heatmaps as well as there are a few point with two peaks between the region of normally three or one peak. One difference between all weight function though is that the width of the region with three peaks is the biggest for the initial implementation before the slightly smaller of the Ricker-wavelet and the smallest one for the weight function with the interaction width.

Other differences can be spotted when looking at the stability check in the heatmaps. For both the initial weight function and the Ricker-wavelet, the region for two peaks got no stars indicating any users far away from either one of the peaks. For the weight function with the interaction width this region got many red stars indicating that there are a lot of users away from the peaks. Again looking at the final distributions explains this. In the end, the two peaks for the adapted initial weight function are not very dense in contrast to the other weight functions where all users are nearby the peaks and therefore are put into one bin in the resulting histogram.

In the region of three peaks the initial weight function only has any indicator stars for values of μ above 0.5. The other functions also have stars for lower values of μ but not within a small range of $0.4 \leq \varepsilon \leq 0.45$. This is because the initial weight function has faster opinion dynamics. The repulsive values are not limited so users that will get dragged towards the outside anyway will do that faster than an user with one of the other weight functions being used. Another factor that speeds up the dynamics is the opinion truncation at -1 and 1 while the opinions for other functions are only forming peaks at opinions of $|o| \approx 2$ and above. This process of self-stabilization of course takes longer than just setting the opinions to fixed values. In the region of one peak, so for bigger values of ε, the initial weight function and its adaptation got a similar pattern of indicator stars. For approximately $0.5 \leq \mu \leq 0.8$ there are green stars, for even bigger values of μ yellow and a few red stars.

For the Ricker-wavelet the green stars already begin at $\varepsilon \approx 0.4$ and there are a lot more red stars for the biggest values of μ. This observation can be explained by looking back at figure 2.1a where dynamics for $\mu = 1$ are shown. There are a lot of big opinion changes which is counterproductive for the formation of one peak as the users fluctuate within a wider range around it than for smaller μ. So this is just another factor to slow down the convergence time. One last thing that should be mentioned is that at the transition from three to one peak more users are not around the given range of either one peak as even when only one peak is classified there are still some users left from the outside peaks. This can be seen in the final distributions where the small outside peaks are not classified anymore.

After this analysis I decided to continue using the Ricker-wavelet weight function as the behaviour is pretty similar to the initial weight function. The convergence speed may be slower but the advantages outweigh this. It is only dependent on two parameters μ and ϵ which is a good thing when putting it into another network model that also depends on many parameters. On top of that it has a non-linear shape which was also seen in the model of social differentiation.

Lastly, the biggest advantage, it automatically tends to zero for big opinion distances so there is no need to involve an external truncating function or an interaction width that implies a hard cut off for the function. In fact this is another reduction of parameters as well. Now I can use the Ricker-wavelet weight function and do not have to worry about artificially limiting the opinion space but rather know that even for a small tolerance parameter ε the opinion dynamics will stabilize at some point as the repulsive values do not tend towards plus or minus infinity.

4.2. Reddit Archetype

The first network topology I want to analyze, while using the current opinion model, is inspired by the structure of the social media platform Reddit. With over 50 million daily active users it surely is a globally influential social media and therefore interesting to look at [17]. The platform is organized in big communities, the so called “subreddits”, in which topic-specific content in form of text, images or videos is being posted. Inside the “feed”, all posts are displayed below each other. Each post has its own comment section where all users can leave a comment or reply on other users comments or replies as well. Figure 4.4 illustrates these different levels of the platform.

Figure 4.4.: Visualization of the structure and different levels of Reddit [18].

Simulating every level of such network would require an enormous computational power and probably just overcomplicates things due to the many different parameters and assumptions that have to be analyzed. My approach was to rather take one part of the whole network and focus on a few interesting aspects that can be derived by a simulation of this process. Doing this also makes the insights more comparable with simulations from another network archetype which only contains a slight variation of the observed process. This is the reason why I choose to only focus on the middle level of Reddit and model the opinion dynamics inside one subreddit.

The first important assumption is that the dynamics inside the comment sections are completely neglected. Instead I assume that every post represents the opinion of the user that has posted it and this opinion is significant for the influence on other users as they either do not even look inside the comment section or any interaction inside the comment section would result in a similar influence like only looking at the post. Thus my modelling approach only takes user to post interaction into account like explained in the section above.

Reddit features a voting system which allows an user to leave an upvote (like) or downvote (dislike) on a post. It is not necessary to vote, one can also just browse through the platform without interacting at all. Dependent on the total amount of up-and downvotes and the age of the post, each post gets a score which then determines the default order of shown posts inside the subreddit. Given t, the time difference between the submission date and a reference date in seconds, and $x = U - D$ the difference between the amount of up- and downvotes U, D where

$$y = \begin{cases} 1\,, & \text{if } x > 0 \\ 0\,, & \text{if } x = 0 \\ -1\,, & \text{if } x < 0 \end{cases}$$

and

$$z = \begin{cases} |x|\,, & \text{if } |x| \geq 1 \\ 1\,, & \text{if } |x| < 1 \end{cases}$$

the exact function for the score of a post[2] is given by [19]

$$f_{\text{score}}(t, y, z) = y \cdot \log_{10}(z) + \frac{t}{45000}\,. \tag{4.3}$$

This implementation has a few interesting characteristics. The first term determines how the difference x between up- and downvotes influences the score. If $x = 0$ or $|x| < 1$, the vote influence becomes zero. For $x < 1$ the logarithm gets a negative, for $x > 1$ a positive sign. In consequence the argument of the logarithm, the absolute value of the difference $|x|$, scales the either positive or negative influence of the votes. Due to the logarithmic scaling, a post with $x = 10$ has to reach $x = 100$ in order to double its score. To double it again even $x = 10000$ and so on.

The second term determines how the time of submission influences the score. In contrast to the first one, it scales linearly and is always positive. In my simulation one iteration reflects the dynamics of one hour. Setting the reference point to zero, t can just be defined as the submission time until the start of a simulation in seconds, which is the current number of iterations multiplied by 3600. First of all, the term is always the biggest for recently posted content. Dividing t by 45000 ensures that e.g. a 24 hours old post with $x = 100$ has the same score as a new post with no votes. Without this feature, very popular posts could stay on top of the subreddit forever and new content would have to beat this score in order to be shown.

[2] The Reddit code only was open-source until 2017 so this formula refers to the archived code who might not have the same implementation of the score function as today.

As already mentioned, the score determines the default order of the shown posts inside the feed. Inside Reddit this is also called sorting by "hot". On top of that there are a few other sorting criteria each users can set manually. Sorting by "new" orders the post chronologically with the newest submissions on top. Users browsing this way have a lot of influence whether a post gets popular as according to (4.3) the first ten upvotes count as much as the first one hundred. Another possibility is sorting by "top" which orders the posts descending by the greatest difference of up- and downvotes during a specified range (last day, week, month or all times). Furthermore there are also the criteria "controversial" and "rising" which show posts with both great amount of up- and downvotes or posts with a lot of recent votes and comments. For my simulation I make the assumption that the opinion dynamics inside one subreddit can be explained by only looking at users browsing hot and new post.

Since each user can manually switch to another sorting criteria instead of the default sorting by hot, I have to distinguish this user behaviour in the simulation. Besides this there are other differences, for example the time being online or if they submit posts or only lurk inside the subreddit. Therefore I distribute the user types into three different categories.

The first category are so called lurkers, which only browse hot posts and never submit anything. The second category also only browses hot posts but is able to submit. The last category is similar to the second one but in addition browses new posts as well. It is known that in social media most users never really contribute anything but rather lurk [20]. This should be remembered when picking the exact category distribution later on. To separate users in their time being online and being able to interact on the platform, I assign each user a certain activity which should be distributed by a power law [21] and reflect the probability of interacting during one iteration step. On top of that, I decided to scale the amount of viewed posts every iteration with the activity as well.

At this point, the remaining design choices for the model are to specify the triggers for an user to vote on a post and when he decides to leave the subreddit as I want to keep the possibility of a dynamical network size. Here I make the assumption that an user only decides to up- or downvote a post, if he significantly likes or dislikes the content. If the absolute opinion difference $|o_{i,t} - \tilde{o}_j|$ between user i looking at post j is smaller than $\frac{1}{2}\varepsilon$, he gives an upvote. If the difference is greater than $\frac{3}{2}\varepsilon$, a downvote. As a leaving mechanism I simply decided that if an user rejects more than 90% of the posts he sees during one iteration, he leaves. This makes sense as one user can only draw conclusions about the communities average opinion from the recently submitted posts.

4.2.1. Model Overview

In the following part I want to visualize the structure of my Reddit model, especially the order of different mechanisms. In addition to that I again list all assumptions and every parameter to have a compact overview and discuss their impact later on. To begin with, the following diagram illustrates how I run one simulation of the Reddit model. The first step is to initialize all distributions and parameters. After that the main iteration cycle starts and repeats for the given simulation time. This part reflects the opinion dynamics inside the subreddit using the ricker-wavelet weight function for user to post interaction. At last all relevant distributions and statistics are shown and saved for further analysis.

Initializing the model

- Simulation time, amount of users and initial opinion distribution.
- User attribute distribution (activity and type/category).
- Parameters for the opinion model (Ricker-wavelet weight function).
- Network specific parameters (posting probability, scrolling duration, joining rate).

$$\Downarrow$$

Iteration cycle

1. New posts are submitted by chance from users that are online and active.
2. Users browse by new posts, might vote and get influenced by them.
3. Score calculation for new posts and score update for older ones. (*loop*)
4. Users browse by hot posts, might vote and get influenced by them.
5. New users join, some users might leave.

$$\Downarrow$$

Statistics and data visualization

- Final user/post opinion distribution.
- Score and vote distribution.
- Opinion timeline plus weight function values.
- Additional observables (amount of posts and leaving users, views per post, etc.).

The following table 4.1 contains all parameters of both the opinion and network model that need to be set when running a simulation. I will qualitatively discuss the impact of every parameter but fix most to focus on the comparison of the network structures. Underneath I listed all the assumptions made for the modelling process.

Parameter	**Meaning**
t_s	simulation time (amount of total iterations)
N	initial network size (amount of users)
α, β	shape of initial user opinion beta distribution
c_0, c_1	thresholds for assigning users to category $0, 1, 2$
γ	exponent of the activity power law distribution
ε, μ	tolerance and convergence parameter for the weight function
p	probability per iteration that an active user posts
d	proportionality constant for the duration of scrolling
ν	joining rate of users per iteration

Table 4.1.: Parameters for the opinion and Reddit network model.

Assumptions

1. Opinion dynamics can be reduced to only user/post interaction.
2. One iteration step is equivalent to one real time hour $\Rightarrow$ discrete dynamics.
3. Users only browse by "hot" and "new" posts.
4. There are three types/categories of users
   (0) Only browse hot posts, never submit.
   (1) Only browse hot posts, submit.
   (2) Browse hot and new posts, submit.
5. The probability of being online/active follows a power law distribution.
6. More active users see more posts per iteration (scroll longer in feed).
7. User $\begin{cases} \text{upvotes}\ , & \text{if } |o_{i,t} - \tilde{o}_j| \leq \frac{1}{2}\varepsilon \\ \text{downvotes}\ , & \text{if } |o_{i,t} - \tilde{o}_j| \geq \frac{3}{2}\varepsilon \\ \text{doesn't vote}\ , & \text{else} \end{cases}$.
8. User leave the subreddit if they reject more than 90% the posts seen per iteration.

4.3. Telegram Archetype

The next network archetype is inspired by social media in which the content is shown chronologically and does not have any voting system like before. Since I want to compare it with Reddit, due to the relatively similar amount of total active users per day, I picked Telegram as an example. Besides its use as a messenger service for private chats, it also features public groups with up to 200.000 members [22].

The structure of a group is pretty similar like the structure of a subreddit. Users of a group can submit content which then builds the “feed” other users can browse trough or answer on something. Telegram does not have a separate comment section. One important assumption to be made then is that the answers on posts, which are shown together with the posts on top, will be neglected. As already mentioned there is no voting system, the newest posts are always on top of the digital interface. Besides groups there are also so called channels where only the administrators of those can submit content. Other users can follow the channels but only see the content inside. Channels do not have a restriction in the amount of members.

In order to model this network structure, I thought of using the same model like before but reducing it to the features of Telegram. Therefore I can nicely compare the two models which are similar in the core and feature the same parameters. In other terms I only have to remove the score calculation, the voting feature and redefine the user categories. Looking back at section 4.2.1 this means that in the iteration cycle steps 2. – 4. are removed and replaced with only one step. In this step the users simply browse all posts and get influenced by them.

Since I do not have to distinguish users browsing by “hot” or “new” anymore, the last user category will be removed. There are only two categories left with users that only lurk and users that also post. One can see the latter category in the Telegram model as a unification of user category 1 and 2 from the Reddit model as they are the ones responsible for the creation of new content.

4.3.1. Model Overview

For the sake of completeness I again visualize the structure of the Telegram model and list all assumptions and parameters. Note how this model is basically a simplification of the former one.

Initializing the model

- Simulation time, amount of users and initial opinion distribution.
- User attribute distribution (activity and type/category).
- Parameters for the opinion model (Ricker-wavelet weight function).
- Network specific parameters (posting probability, scrolling duration, joining rate).

$$\Downarrow$$

Iteration cycle

1. New posts are submitted by chance from users that are online and active.
2. Users browse all posts and get influenced by them. (*loop*)
3. New users join, some users might leave.

$$\Downarrow$$

Statistics and data visualization

- Final user/post opinion distribution.
- Score and vote distribution.
- Opinion timeline plus weight function values.
- Additional observables (amount of posts and leaving users, views per post, etc.).

Parameter	**Meaning**
t_s	simulation time (amount of total iterations)
N	initial network size (amount of users)
α, β	shape of initial user opinion beta distribution
c_0	threshold for assigning users to category $0, 1$
γ	exponent of the activity power law distribution
ε, μ	tolerance and convergence parameter for the weight function
p	probability per iteration that an active user posts
d	proportionality constant for the duration of scrolling
ν	joining rate of users per iteration

Table 4.2.: Parameters for the opinion and Telegram network model.

Assumptions

1. Opinion dynamics can be reduced to only user/post interaction.
2. One iteration step is equivalent to one real time hour $\Rightarrow$ discrete dynamics.
3. There are two types/categories of users
   (0) Browse all posts, never submit.
   (1) Browse all posts, submit.
4. The probability of being online/active follows a power law distribution.
5. More active users see more posts per iteration (scroll longer in feed).
6. User leave the group if they reject more than 90% the posts seen per iteration.

4.4. Fixed Network Size

In the following part both models will be compared for exactly the same initial conditions with a network size of $N = 10000$ and a simulation time of $t_s = 336$ (two weeks). The initial user attribute distributions are displayed in figure 4.5. The parameters of the opinion and network model are picked as

$$\varepsilon = 0.4, \quad \mu = 0.1, \quad p = 0.01, \quad d = 50, \quad \nu = 0 \; .$$

Since $\nu = 0$ the network size will be fixed at first. Picking $(\varepsilon, \mu) = (0.4, 0.1)$ ensures smooth dynamics with some repulsive behaviour, $(p, d) = (0.01, 50)$ results in a moderate flow of new posts and that users with maximum activity see 50 posts per iterarion.

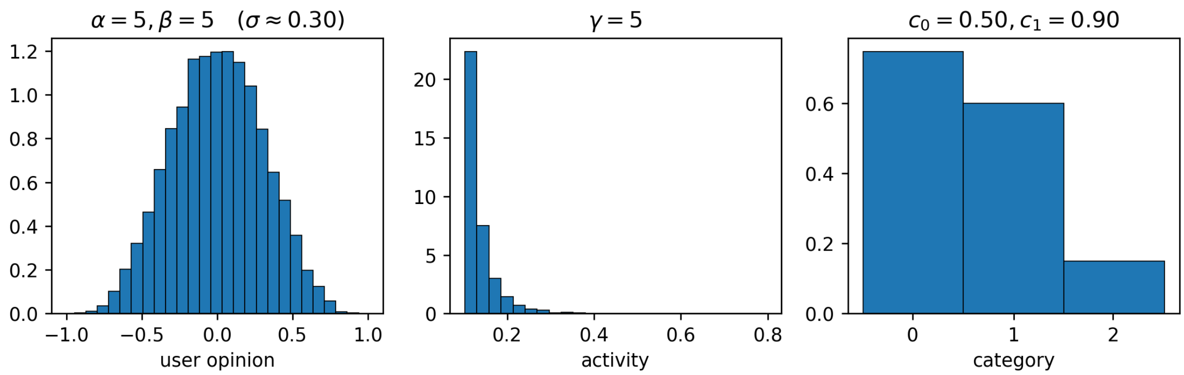


Figure 4.5.: Initial user opinion, activity and category distribution. This state reflects a community with a moderate mean opinion but some opinion variance. Most users have a low activity and 50% are "lurkers" that do not submit content.

Choosing $\alpha = \beta = 5$ leads to an initial state in which the mean opinion is set to zero. Since the opinion distribution is symmetrical the standard deviation $\sigma \approx 0.30$ represents a case in which a few users are present towards both extremes. The exponent of the activity power law distribution $\gamma = 5$ and the category thresholds $(c_0, c_1) = (0.5, 0.9)$ capture the observation of reality that many users have a low activity and do not participate in the community besides browsing the content.

For the Telegram model the category threshold c_1 is omitted as it does not appear in the model. The parameter c_0 and every other stays the same.

Figures 4.6 and 4.7 show the observed quantities for both models in the final state for the given initial conditions.

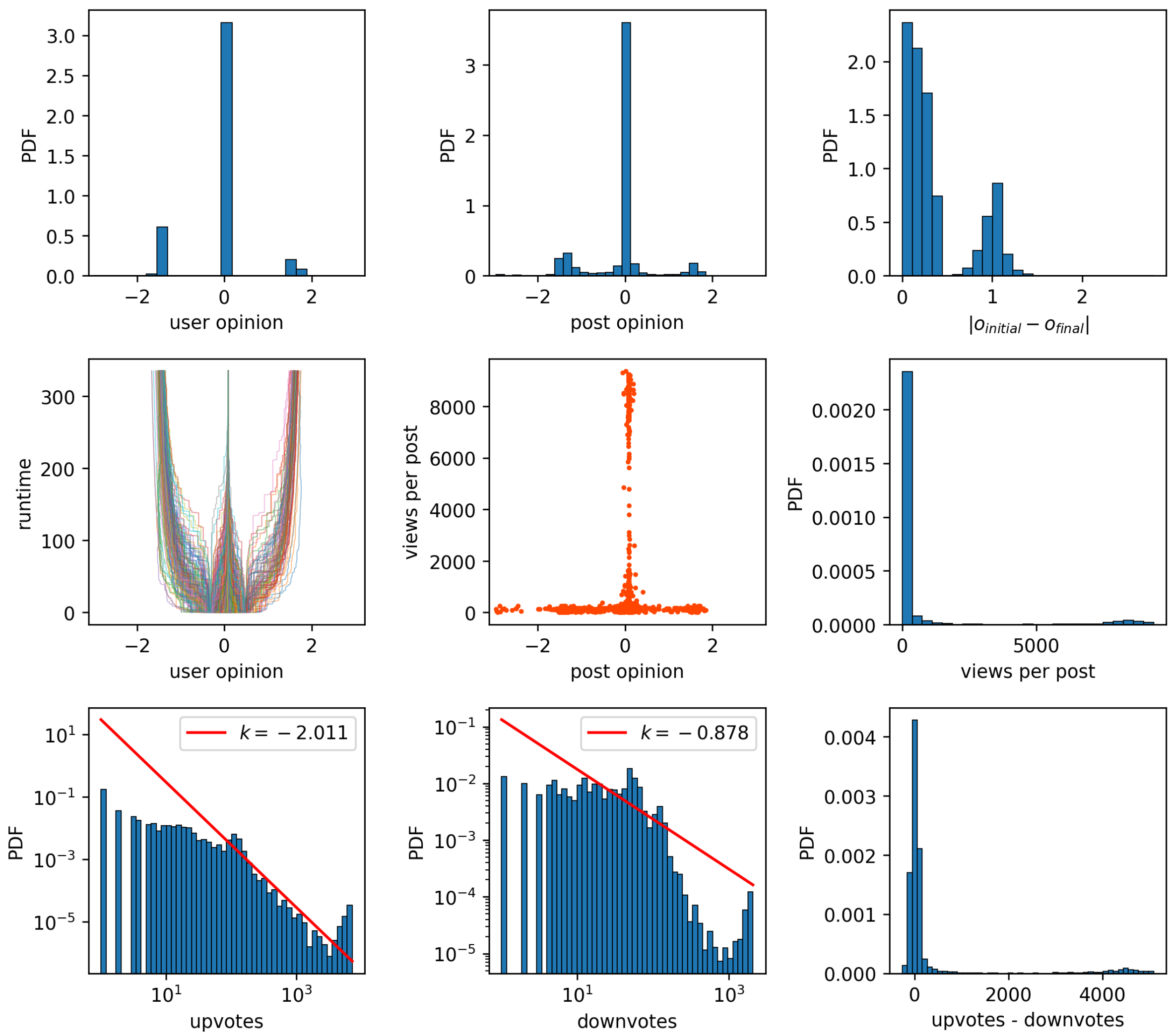


Figure 4.6.: Final state of the Reddit model after $t_s = 336$ iterations for a fixed network size $N = 10000$. User, post and initial vs. final user opinion distribution in the top row. Opinion timeline, views per post dependent on post opinion and views per post distribution in the middle row. Up- and downvote with their difference distribution in the bottom row. Additional power law check with resulting exponent k. Final state of fragmentation with three peaks in the opinion distribution. Many views per post for moderate opinions.

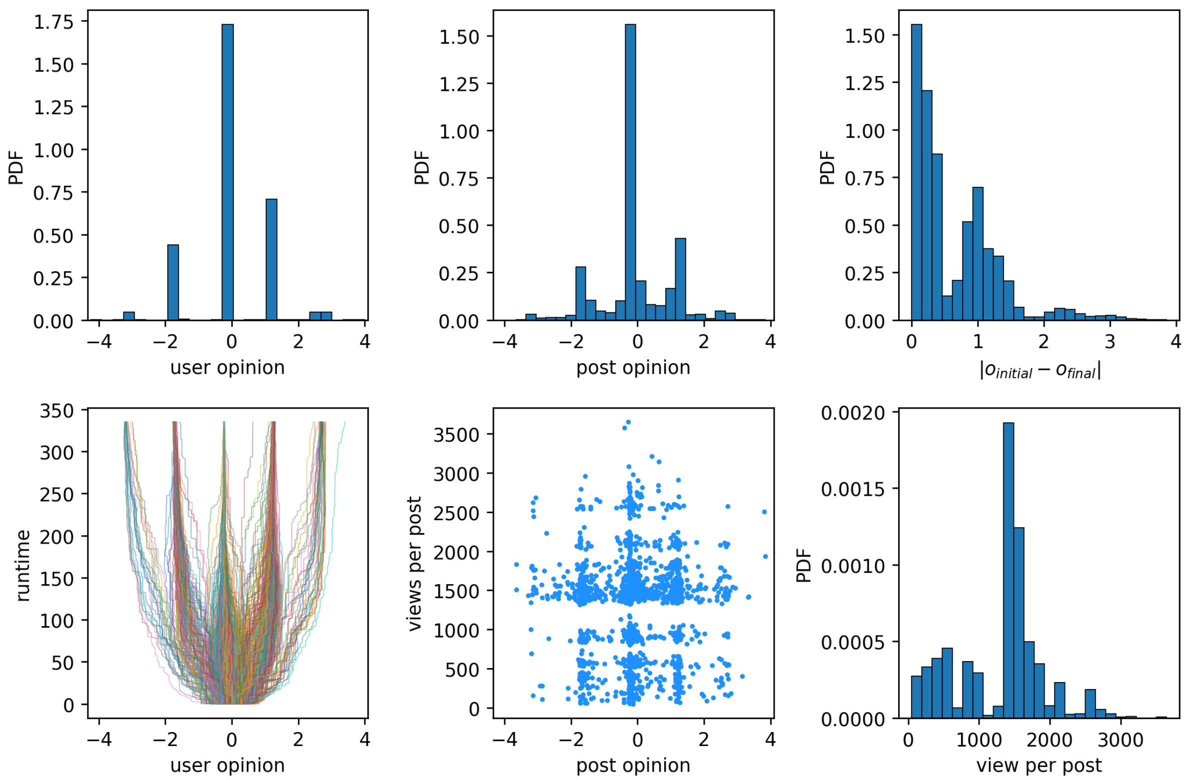


Figure 4.7.: Final state of the Telegram model after $t_s = 336$ iterations for a fixed network size $N = 10000$. User, post and initial vs. final user opinion distribution in the top row. Opinion timeline, views per post dependent on post opinion and views per post distribution in the bottom row. Final state of fragmentation with five peaks in the opinion distribution. Similar pattern in the views per post for every opinion peak. Posts from every opinion can get many views.

Looking at both final state overviews the first big difference is the amount of peaks in the user opinion distribution. For the Reddit model there are three peaks with the middle one being clearly the most present at a moderate opinion of approximately zero. The other two peaks occur around the opinions $o \approx \pm 1.8$. For the Telegram model five peaks are observed but the middle one at zero is not as big as before. The peaks beside this one are in the same region like in the Reddit model. On top of that the small outside peaks are located around $o \approx \pm 3$.

In the post opinion distributions for both models one can see a pretty similar shape like the user opinion distribution but with some additional bins filled in between the peaks. Since in the beginning of the simulation users have not separated yet, these are post being submitted at that time. Due to the relatively quick emergence of user subgroups this effect does not last long.

Comparing the distributions of the absolute distance of the initial and final user opinion only highlights again the characteristics of the final user opinion distribution. For both models the majority of users stay at their initial opinion. The amount of users with a bigger opinion difference regarding their starting point decreases until the absolute opinion distance between the middle and neighbor peaks is reached. Therefore there is a peak in this distribution as well. For the Telegram model there is an additional peak which reflects the mean of the absolute opinion distance between the inner peak and outer peaks.

Both opinion timelines show the separation of the users into their final subgroups. As in the Reddit model there are only three groups being formed, the distinction of these can be made at a runtime of approximately 150 iterations whereas in the Telegram model the clear separation happens later. Overall the opinion development shows a familiar shape like running the opinion model on a fully connected network. The big difference though is that the opinion changes can not only be explained by the parameters ε and μ but rather with how the users interact with the posts.

Speaking of this mechanism one important observable is the views per post dependence on the post opinion. The more views a post of a certain opinion has, the more users were influenced by it which is the crucial force for the opinion dynamics and strongly dependent on the network structure. For the Reddit model one can see that only posts with a moderate so the overall mean opinion have the chance of getting many views per posts. While these ones can get up to 9000 views, other ones with a slightly differing opinion have way less views and never even reach 1000 views. In contrast to that in the Telegram model posts of all opinions have the possibility of reaching many views as seen in the scatter plot. But the maximal amount of views seems to be limited at around 3500. In section 4.6 these observations are further explained and discussed.

For the Reddit model the upvote distribution shows the shape of the power law distribution wwith an exponent $k \approx -2$ between an amount of 10^2 and $5 \cdot 10^3$ upvotes. For the downvotes the power law fit does not produce any matching result as the region between $1 - 10^2$ upvotes is very flat. Overall there are many posts with a small difference in up and downvotes and a few with a bigger difference with more upvotes. There is a small peak at an even bigger difference. This is caused by the few posts that were seen the most.

4.5. Dynamical Network Size

For the dynamical network size all initial parameters were picked the same as above with the only difference that $\nu = 0.001$. Figures 4.8 and 4.9 show the resulting final states.

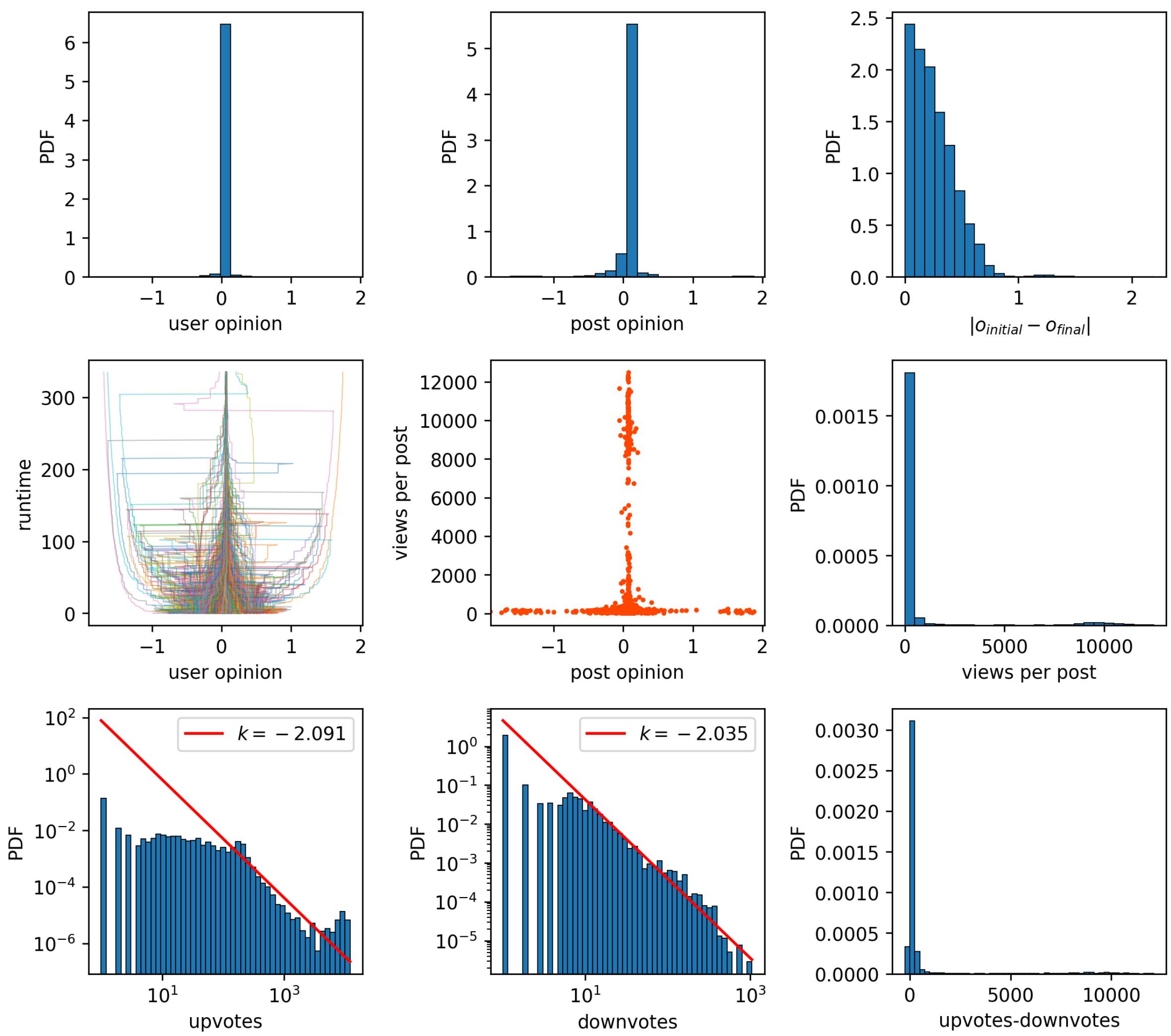


Figure 4.8.: Final state of the Reddit model after $t_s = 336$ iterations for a dynamical network size with $N_{final} = 13802$. User, post and initial vs. final user opinion distribution in the top row. Opinion timeline, views per post dependent on post opinion and views per post distribution in the middle row. Up- and downvote with their difference distribution in the bottom row. Additional power law check with resulting exponent k. Final state of consensus with one peak in the opinion distribution. Many views per post for moderate opinions. Jumps in the opinion timeline show that the user left and was replaced by a new one.

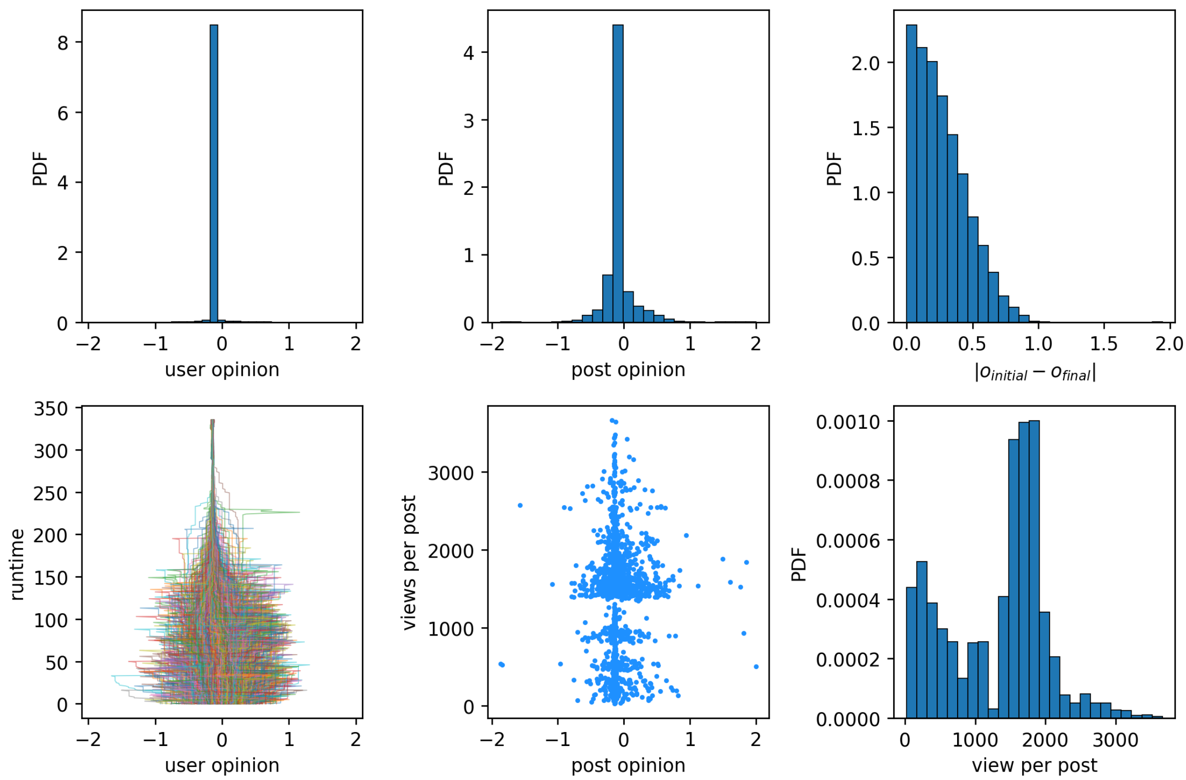


Figure 4.9.: Final state of the Telegram model after $t_s = 336$ iterations for a fixed network size $N_{final} = 13802$. User, post and initial vs. final user opinion distribution in the top row. Opinion timeline, views per post dependent on post opinion and views per post distribution in the bottom row. Final state of consensus with one peaks in the opinion distribution. Many views per post for moderate opinions with some fluctuations towards both extremes. Jumps in the opinion timeline show that the user left and was replaced by a new one.

Setting $\nu = 0.001$ means dynamical users are enabled and those which reject more than 90% of the posts they see during one iteration leave and get replaced by new ones. On top of that $\nu \cdot N$ users join to make the network grow a bit. The new users are drawn by the same initial distributions.

Looking at the final states of both models it gets clear that this leaving mechanism prevents the users from splitting up into multiple subgroups. Instead there is only one big peak in the user opinion distribution left with the moderate opinion zero. As a consequence of that the distribution of the absolute opinion distance between initial and final opinion is now strictly decreasing without having additional peaks as well.

Comparing both opinion timelines the leaving mechanism causes many users in both models to be reset at more moderate opinions which is seen by the big jumps in the timeline. One difference though is that more users are leaving in the telegram model and thus consensus is reached faster by the larger amount of more moderate users joining. This makes sense as Telegram users have a higher probability of seeing posts they reject due to the views per post characteristics observed before. In the Reddit model some extreme users still stay as they do not see many posts with extreme differing opinions as most posts seen have a moderate opinion.

For both models the views per post dependent on the post opinion show the same characteristics as before. For the Reddit model still only moderate posts are seen very often. Due to the increasing amount of users some posts get up to 12000 views. In the Telegram model theoretically posts of all opinions can be seen often by everyone. Since extreme users leave very quickly and therefore only few posts with such opinions are submitted, this is not observed very often.

The upvote distribution for the Reddit model is very similar to the one for a fixed network size. On the other hand the downvote distribution now can be fitted to a power law with the exponent $k \approx -2$ as well.

After doing the comparison for both models with fixed and dynamical network size, what does this mean for the real platforms? On one hand it gets clear that when users show a repulsive behaviour the community tends to split up into multiple subgroups with some of them having extreme opinions as well. If the users with differing opinions from the general consensus do not leave the community, they just split it up even more.

If those users with an extremely differing opinion leave, fragmentation or polarization can be avoided as seen in both models ending up in a state of consensus. But it could also happen that users with a moderate opinion are in the minority and the community ends up at a state of consensus at an extreme opinion.

Since the whole simulation happened on an artificial timescale it is hard to say which situation reflects the real world the best. There are both big communities with a very dynamical user base but also smaller communities with a rather constant amount of users. On top of that is has to be taken into account how the platform specific post sorting effects the dynamics nonetheless.

4.6. Views per Post

Whenever a user views a post his opinion gets influenced. Therefore posts which are seen by many users play a big role for the opinion dynamics. The recent observations show that the network structure has a strong impact on the maximal possible amount of views and on the view distribution per opinion. Figure 4.10 directly compares these characteristics for both models with a fixed or dynamical network size.

(a) Fixed network size

(b) Dynamical network size

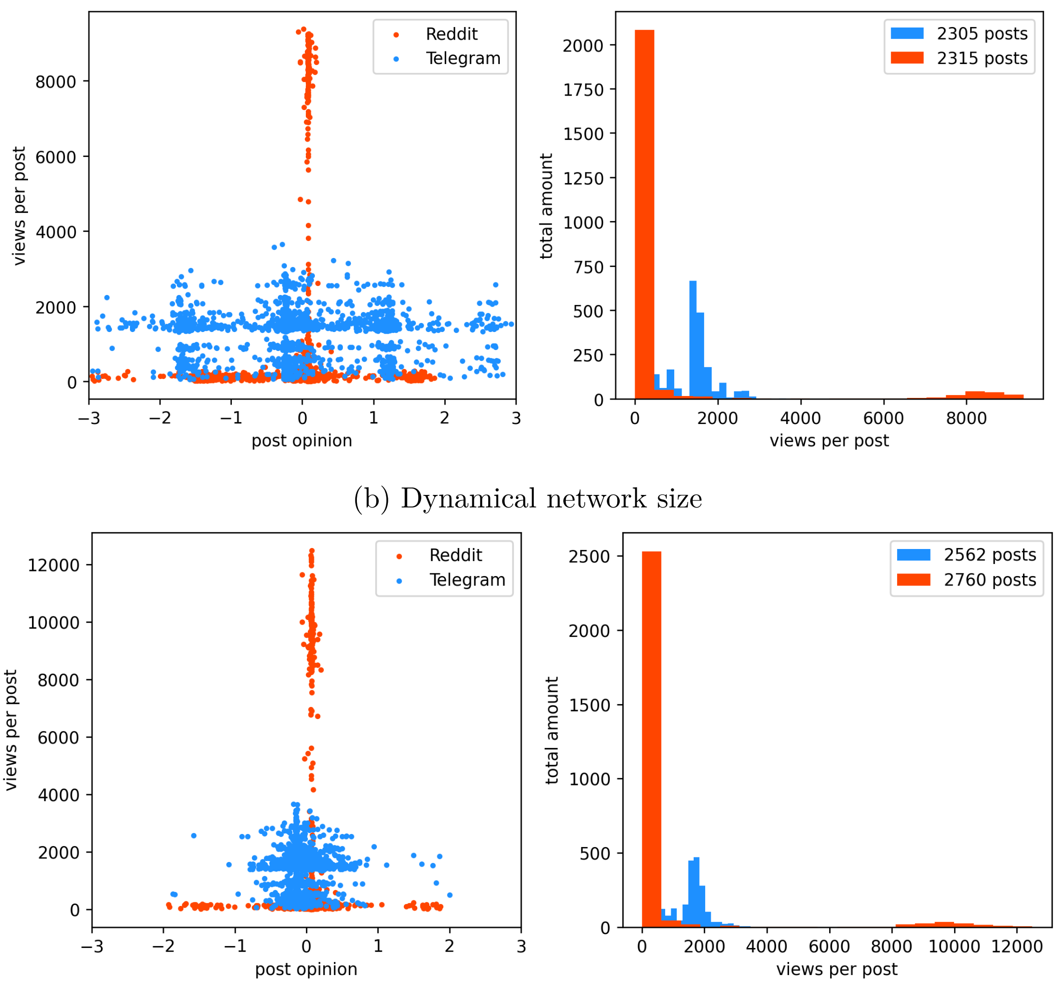


Figure 4.10.: Views per post dependent on the post opinion and views per post distribution for both models with fixed and dynamical network size. In the Reddit model only a few posts with moderate opinions have many views but overall most of the posts have less views. For Telegram posts of all opinions share a similar distribution but with overall less views than the maximal viewed ones in the Reddit model. Dynamical users reduce the creation of extreme posts but do not effect the views per post characteristics in general.

Looking at the comparison leaves some questions. Why exactly do some posts in a narrow region of opinions have so many views in the Reddit model while in the Telegram model posts of all opinions seem to have a similar view distribution with overall less views than the ones in Reddit? What role does the choice of parameters and initial conditions play in this and what is caused by the network structure itself?

The first thing one can say for sure is that the vote and ranking system of Reddit favors posts with opinions close to those of many users to be seen very often and also make them accessible for a longer period of time. Since the majority of users browse by “hot” and have a moderate opinion, the most upvoted posts will also have a moderate opinion. The crucial point here is that once a post got many upvotes it takes some time for it to disappear at the top of the hot section looking back at the score function (4.3).

While the most upvoted posts in the Reddit model can be seen over a few iterations, the posts in the Telegram always get replaced by newer ones due to the chronologically ordered system. On top of that there is no mechanism here which makes posts with a differing opinion from the general consensus less accessible. Thus the amount of views a post gets is not dependent on the posts opinion but rather on the activity of the users and the rate of new posts.

Looking back at the views per post distribution of Reddit there is a huge peak for posts with only $0 - 500$ views. This is even less than the majority of posts from the Telegram model. Therefore such posts are only seen by the users that browse by “new” or by those which scroll long enough in the hot section as there are always the few posts with more upvotes on top which stay for some iterations.

The dynamical network size causes top posts in the Reddit model to even have more views. This is simply due to the fact that more users browse trough hot during each iteration. On the other hand for the Telegram model it seems like the maximal amount of views a post can get did not increase even though there are more users which browse the feed every iteration. But more users also means more posts and thus one post gets replaced more quickly in the chronologically ordered system.

In summary this means the vote and ranking system of Reddit is responsible for amplifying the amount of views on posts by letting the most upvoted ones staying on top of the hot section for more iterations than in a just chronologically ordered system like Telegram. Furthermore it gets clear that the user browsing Reddit by “new” have a lot of impact on the overall opinion dynamics as they have the biggest influence with their votes what posts are displayed in the hot section. Therefore the opinion and activity distribution of those users is a big factor in the outcome of the dynamics. For the Telegram model the opinion model itself has the biggest impact on the outcome as the way posts are displayed can not be influenced by the user behaviour.

Of course the initial user opinion distribution highly effects the opinion dynamics. But the point is that nonetheless the views per post in a chronologically ordered system are only dependent on the amount of users, their activity (time being online and duration of browsing) and the probability of the creation of a post per iteration. In contrast to that the views per post in a vote and ranking system are dependent on the posts opinion, the user category and opinion distribution plus on the same parameters like before.

Knowing all of this the Reddit model should have a completely different dependence of the opnion on the views per post with another initial state. To test this statement figure 4.11 shows the observed dependence for the same initial conditions and parameter choices like in section 4.4 but with a different initial user opinion distribution.

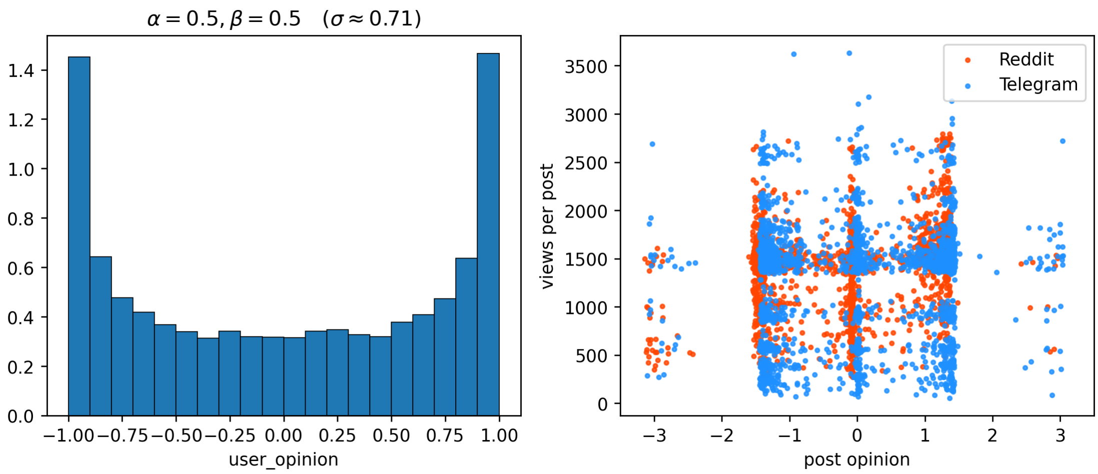


Figure 4.11.: Views per post dependent on the post opinion for a fixed network size and same initial conditions with a bi-modal initial user opinion distribution. The Reddit model shows a similar dependence like the Telegram model.

For the case of a symmetrical opinion distribution with two outside peaks at an extreme opinion both models end up in a state of fragmentation with five subgroups. In contrast to earlier observations the Reddit model now shows a pretty similar dependence of the post opinion on the views per post like the Telegram model. There is not one opinion favored which gets significantly more views than posts from other opinions. Since there is such big variance in the users opinions posts in the hot section of Reddit get many downvotes as well. This results in the hot section having a much higher rate of new posts with many different opinions on top and therefore being more similar to a chronologically ordered system even though users vote.

5. Summary

In the beginning of this thesis I introduced the basics of modelling opinion dynamics in social networks by presenting three different classes of one-dimensional opinion models, namely those with assimilative, similarity biased and repulsive influence. In the need of a model which belongs to the latter class and is able to produce a final state of either consensus, polarization or fragmentation while technically allowing opinions to exceed their initial range, I proposed using the model of social differentiation.

This model is able to explain the emergence of very distinct subgroups in small world networks and shows how opinion fluctuations make the outcome of it less sensitive to the initial conditions. However the shape of the used weight function seemed not to be fitting for the usage in another network model with a social media like structure.

Instead I introduced an adaptation for this weight function by scaling the repulsive term with an additional parameter and limiting the repulsive values with a hard cut to zero at a certain absolute opinion distance. A second idea was to use a Ricker-wavelet as the weight function due to its similar characteristics but without adding additional parameters besides ε and μ. On top of that it is a fully continuous function and automatically tends to zero for bigger absolute opinion distances.

Looking at only user to post interaction, I analyzed the dependence of ε and μ on the amount of peaks and stability at a final state for the initial, adapted and Ricker-wavelet weight function. All variants show a similar pattern for the amount of peaks at the given parameter space. Low values of ε result in two peaks which transitions into a small region of three peaks for increasing ε with stable states only for low μ. For even bigger ε there is only one peak left. Increasing μ generally causes more unstable states but might allow a different outcome for the same tolerance parameter.

After this analysis it got clear that the use of a Ricker-wavelet as the weight function for an opinion model produces known outcomes. It has big advantages like being continuous and opinions do not need to be truncated by an additional mechanism even though it is an unbound confidence model.

The next step was to apply this opinion model on a network with a social media like structure and to compare the opinion dynamics between a vote and ranking system and a chronologically ordered system for the way of displaying posts to the users. Due to the similar user size, the modelling process was motivated by the social media platforms Reddit and Telegram.

In order to compare both structures, the platforms had to be reduced to a level reflecting one community (subreddit or group) and the models had to be implemented in a similar way besides the different post sorting. In the end I developed a compact toy model for those type of networks with the possibility of assigning different user types ("lurker", submitting content), user activity which scales to the amount of seen posts and controlling network specific parameters like the rate of new posts or joining users.

Using the exactly same initial conditions and parameters both models were analyzed for a fixed and dynamical network size. The first important observation was that when users do not leave the network, the Telegram model ended up having small groups of users at more extreme opinions than the Reddit model. This was caused by the fact that in Telegram posts of every opinion have the same chance of being seen by many users and thus cause more repulsion.

In contrast to that the amount of views a post from Reddit gets is dependent on the posts opinion. In general many posts are not seen by the users as they only browse in the hot section where posts are ordered by their score while being visible for a longer period of time when having many upvotes. Most new posts do not exceed the score of the older top posts.

But in order to get many upvotes the users browsing in new have to upvote posts with an opinion that is similar to the one of many other users. Whenever those appear on top of the hot section it is very likely for them to get significantly more views than ones with a differing opinion as they trigger a process of preferential attachment (posts with many upvotes, automatically get more).

This shows that the opinion and activity of users browsing new posts play a big role in the opinion dynamics as well. For future research it might be interesting to look deeper into the impact of different opinion and activity distributions for the users browsing new. Besides that it could be analyzed how the ratio between different user categories effects the characteristics of the vote and ranking system as well.

Generally speaking, this vote and ranking system of Reddit amplifies the views per post on content with an opinion close to opinions from many others, if the post manages to appear on top of the hot section after being upvoted by users in new. Furthermore it prevents posts with a differing opinion from most other users to be seen often as if those posts were visible in the hot section, they would get downvoted and disappear.

On the other hand if the community has a big variance in its opinion distribution or most users browsing new posts have an extreme opinion, the post sorting would become more similar to a chronologically ordered system like Telegram. This is due to the users browsing hot rejecting and downvoting more posts which decreases the score and thus there is a faster flow of newer posts.

In a chronologically ordered system there is no dependence of the posts opinion on its views. Only the rate of new posts, amount of users and their activity influences the view numbers. In communities with a constant amount of users this can lead to the emergence of multiple subgroups of moderate and extreme opinions as every user sees posts of all opinions with the same probability (disregarding the eventually different ratio of subgroup sizes). In further research one could analyze how this mechanism impacts dynamical communities in general since it was observed that it could also cause more users to leave.

Taking the foundation of these platform models one could also try to further modify the network specific post sorting. For example it would be really interesting to see the Reddit model with a different function for the score, perhaps with a linear instead of a logarithmic term. Another fun thing to do might be only allowing up- or downvotes. Furthermore there is still a lot of mathematical analysis of the whole parameter space left. Since the whole network dynamics rely on stochastic processes it should be possible to predict mean values for some of the observables dependent on the network parameters.

In the end, transfering model insights to the real world should only be done very carefully. Opinion dynamics in social media are a complicated process where even in one platform the community size, opinion distribution and user behaviour is completely different. Besides that the Reddit and Telegram model only describe one community whereas it is also possible for users to browse a feed with content from multiple communities. On top, the Reddit model neglects the features of a comment section and the Telegram model the possibility of giving an answer to a submission.

Another important factor for the opinion dynamics in Reddit are moderators or group administrators in telegram. Those can arbitrarily remove users or content and are therefore very powerful inside communities. The occurrence of bots can be relevant as well. But these things are a completely different topic on their own.

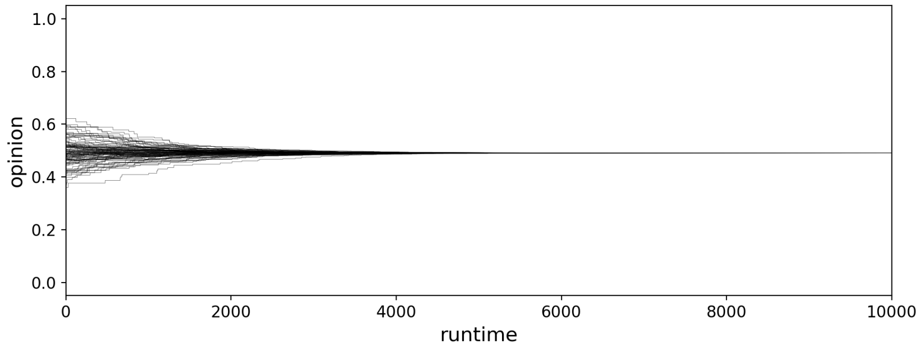
Even though the modelling process underlies a lot of assumptions and simplifications, one can definitely see the differences between a vote and ranking and a chronologically ordered system for displaying content. The first system could be useful for preventing extreme opinions to be visible for a lot of users in moderate communities but it is also capable of amplifying the amount of views on a majority opinion whether it is extreme or moderate. Besides, users voting new content have more influence on the displayed content than the rest.

The second system could have a good use for communities with a bigger opinion tolerance or only moderate opinions as it does not bias the influence of a certain post with votes. It rather equally displays posts of all opinions until new content replaces them. But this might be a danger as it also allows extreme opinions to get a lot of views independent of the overall opinion and therefore trigger a faster process of fragmentation or polarization.

After all it can not be said which system is better or worse. Both have advantages and possible dangers but whether these occur are highly dependent on the user base. I personally think that it is just necessary to be transparent about the impact of platform specific mechanisms on opinion dynamics and reinforce the user awareness of such topics. Of course there has to be a parallel ongoing research which helps to understand all processes and motivates reducing the possible risks of certain mechanisms.

A. Additional Plots

(a) $N = 100$, $\mu = 0.01$, initial opinion normal distribution with $\sigma \approx 0.05$.



(b) $N = 100$, $\mu = 0.1$, initial opinion normal distribution with $\sigma \approx 0.2$.

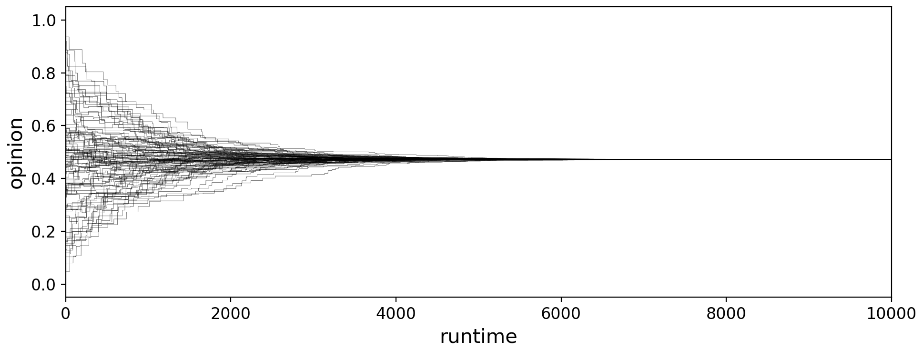
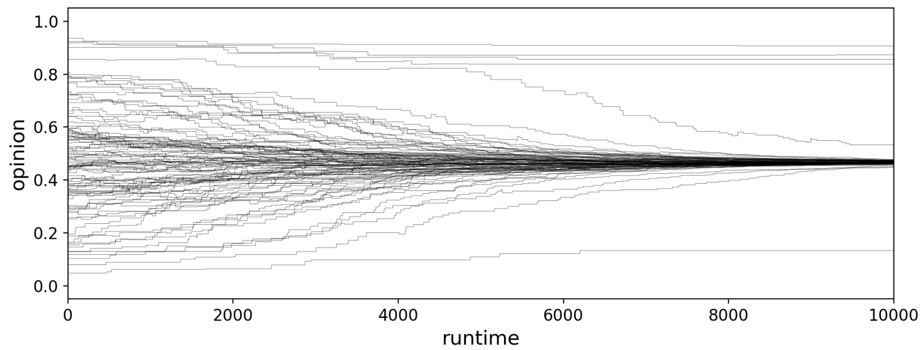
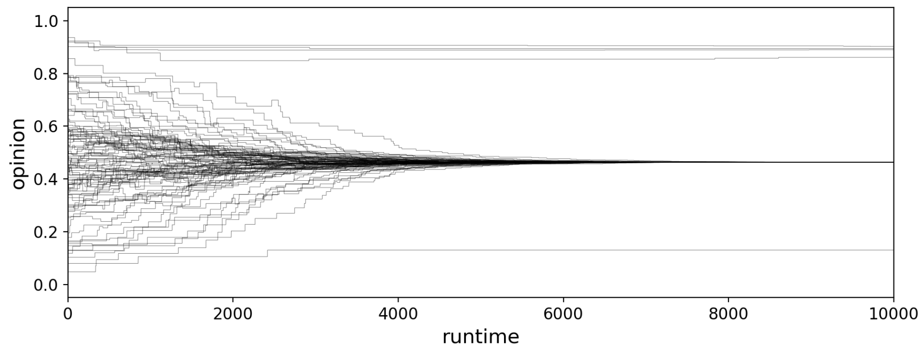


Figure A.1.: Model with assimilative influence. Opinions of all agents dependent on runtime for a different network size N, convergence parameter μ and initial opinion distribution with standard deviation σ.

(a) $N = 100$, $\varepsilon = 0.3$, $\mu = 0.05$ and initial $\sigma \approx 0.2$.



(b) $N = 100$, $\varepsilon = 0.3$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.



(c) $N = 1000$, $\varepsilon = 0.3$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.

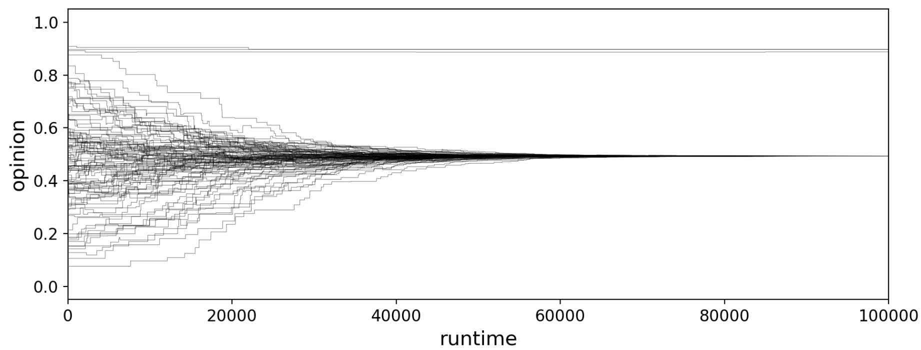
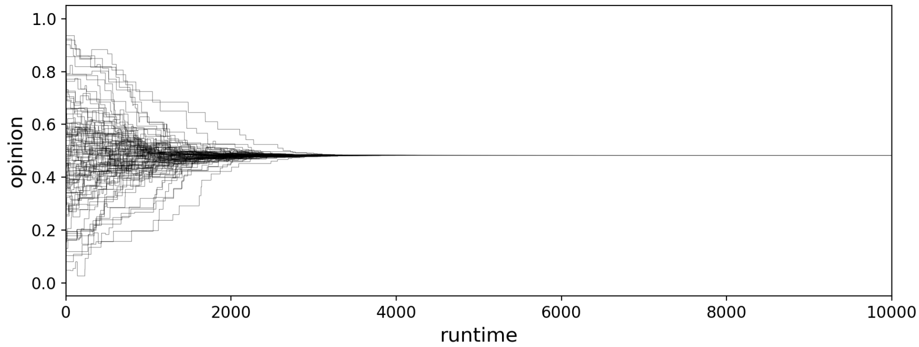
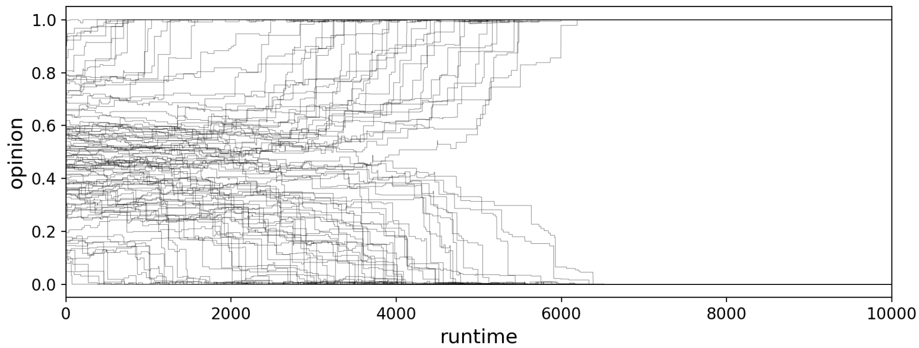


Figure A.2.: Model with similarity biased influence. Opinions of all agents dependent on runtime for a different network size N, tolerance parameter ϵ, convergence parameter μ and initial opinion distribution with standard deviation σ.

(a) $N = 100$, $\varepsilon = 0.55$, $\mu = 0.3$ and initial $\sigma \approx 0.2$.



(b) $N = 100$, $\varepsilon = 0.35$, $\mu = 0.1$ and initial $\sigma \approx 0.2$.



(c) $N = 100$, $\varepsilon = 0.35$, $\mu = 0.1$, initial $\sigma \approx 0.2$ and bigger opinion range $o_{i,t} \in [-1, 2]$.

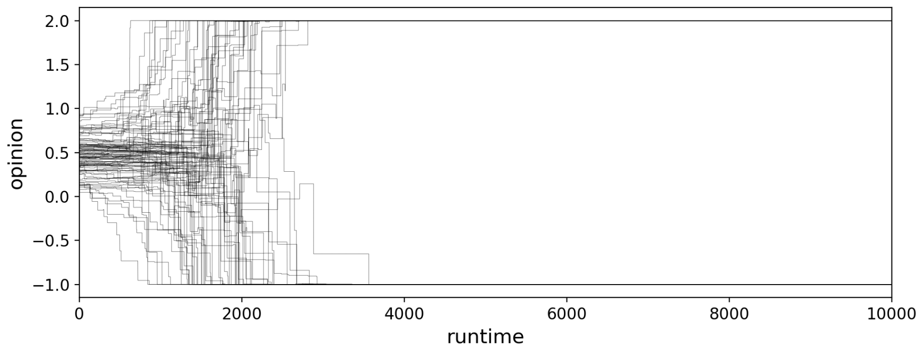
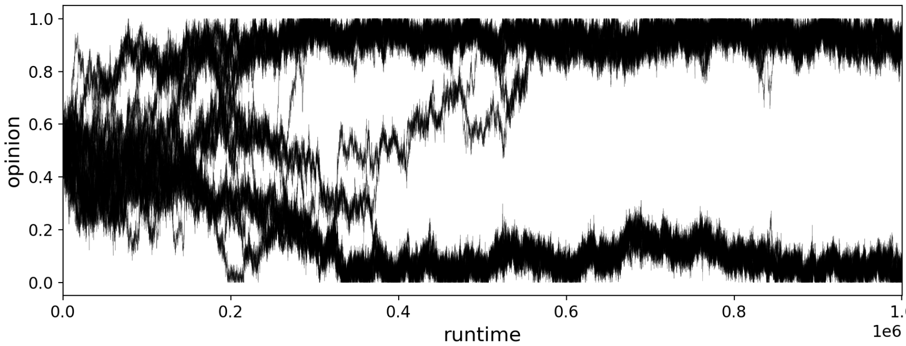


Figure A.3.: Model with repulsive influence. Opinions of all agents dependent on runtime for a different network size N, tolerance parameter ϵ, convergence parameter μ and initial opinion distribution with standard deviation σ.

(a) Allowing $|f_w| < 10^{-5}$.



(b) Allowing $|f_w| > 0.075$.

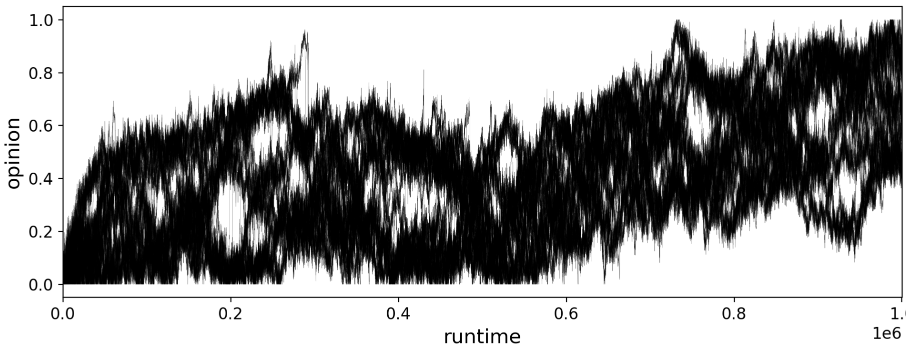


Figure A.4.: Model of social differentiation. Opinion timeline for initial distribution with $\sigma = 0$, $N = 100$ and parameters $\varepsilon = 1$, $s = 0.025$, $a = 100$. Two cases for different model behaviour without limiting the weight function f_w.

B. Peak Find and Stability Check Algorithm

In the following part I will explain the algorithms that I used in section 4.1.1 for finding the amount of peaks of a given opinion distribution and for measuring the amount of users between these peaks as an indicator for the stabilitys. At first since I am only using one dimensional distributions I tried using the function `signal.find_peaks` from the python library SciPy [23]. This function takes an one dimensional array X as an input and returns the indexes of all local maxima as an array M by comparing neighboring values of the array. So without setting other criteria for the classification, a value $x_i \in X$ will be classified as a peak if $x_{i-1} \leq x_i \geq x_{i+1}$. The following figure B.1 shows one exemplary application.

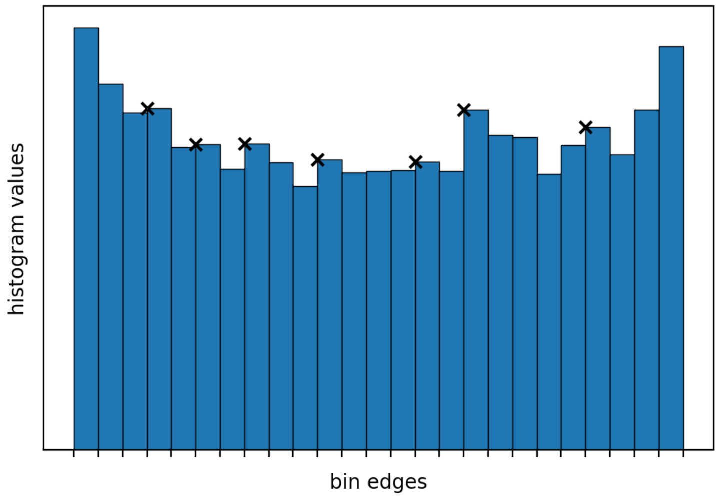


Figure B.1.: Finding peaks of a histogram with `scipy.signal.find_peaks` without setting additional criteria. Black crosses indicate the returned position for the peaks.

At first glance there are a few things which are not ideal for my purposes. The returned indexes for the peaks of a histograms are the bin edges but the midpoints of each bin are relevant for further calculations as they reflect the opinions around the users concentrate. Thus I would prefer having the x-coordinate of each peak shifted to the midpoints which can be done by just adding one half of a bin length on top of each coordinate. Another problem is that the outside histogram values are not recognized by the algorithm due to the missing value to the left/right to compare if it is a peak. But since these peaks would represent a state of bi-polarization they definitely should be included. In order to solve that problem other criteria for the classification must be determined.

When looking at typical final distributions two peaks never emerge very close to each other. This is just given by the properties of the opinion dynamics. So if by coincidence a bin nearby another peak to which the opinions wander is classified by the initial criteria, I want to exclude this case. Fortunately there is an argument for the function, the `distance`, that passes an additional criteria for the classification, the required minimal horizontal distance in samples between peaks. As I am using 25 bins for most of the time, I set the argument to $\texttt{distance} = \frac{\#\text{bins}}{5}$.

Another useful criteria to set is the required height of peaks. In cases of the formation of multiple peaks it could happen that most of the users are around the peaks and in between the bins have the height zero. But there are states in which there is always a slow movement from the inner peak to the two outside ones resulting in some of the bins in between having a value larger than zero. Without setting a threshold for the minimal peak height, these bins would also be classified as peaks. By passing the function the argument $\texttt{height} = 0.25$ I ensure that this can not happen.

In order to solve the problem with the outside peaks, I first let the function classify the peaks with the new criteria. Afterwards I manually check if the first and last value of the histogram fulfill the same criteria for distance and height but only compare if its greater or equal to the right/left value. Figure B.2 illustrates how the new classification works.

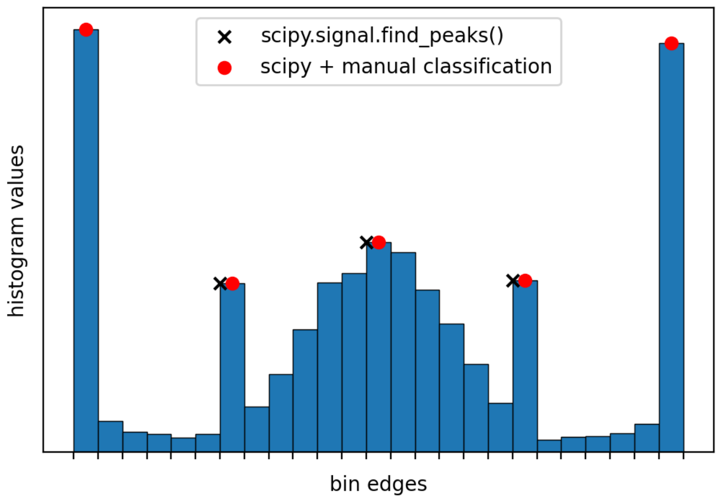


Figure B.2.: Comparing both peak classifications. The manual method recognizes every peak and return the position in the middle of the bin.

For the stability check of a current state I thought of having a metric in form of a measurement for the user movement on the opinion space. Taking into account that with the Ricker-wavelet weight function the system always forms one or multiple peaks as a consequence of the specific repulsive and attractive behaviour, the amount of users within a certain range of a peak should reflect how stable the system is. If there are many users in between peaks and do not form a peak by themselves, it means they are still being attracted or pushed away by the posts opinions and the system needs more time to stabilize.

My stability check function basically takes the current distribution, uses the just explained peak finding algorithm and calculates the absolute opinion distance from every user to every peak. The function sums up the amount of uses that do not have an absolute distance below a certain threshold and returns the value. In the figure B.3 below a few cases are shown for a distance threshold of 0.25.

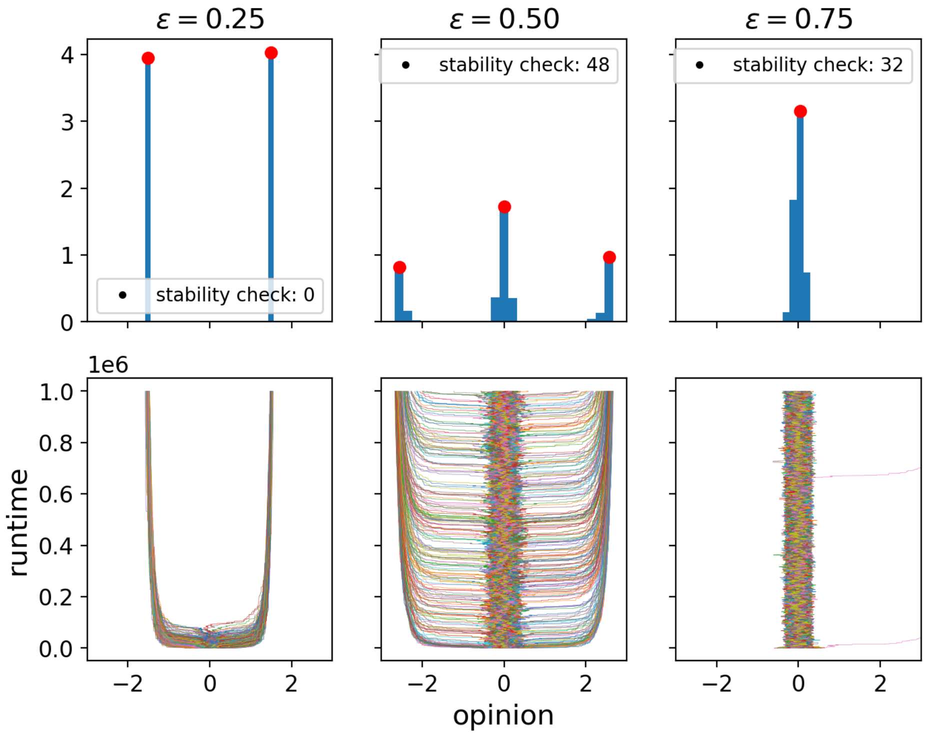


Figure B.3.: Final distributions, opinion timelines and stability check for $N = 1000$, $\mu = 0.5$ for different values of ε after a runtime of 10^6 iterations. Initial user and post beta distribution with $\alpha = \beta = 10$. All states show a different stability as a result of user movement or peak width.

The left state is the most stable as two peaks were formed with all users being concentrated very dense around them. There is no and will not be any further more movement away from the peaks. This is the case when the tolerance parameter is set very small, in this case to $\varepsilon = 0.25$, and in combination with the current post distribution ($\alpha = \beta = 10$) all interactions are very distant and the weight function only returns values close to zero. One can see how in the beginning of the simulation all users are repelled pretty quickly from the posts. After that they only move very slowly towards the outside until the weight function tends to zero.

The middle state is the most unstable one. There are three peaks of nearly about the same size with the inner one having the biggest width. Looking at the opinion timeline already gives a hint that there is much user movement during all times which the algorithm captures. In fact setting $\varepsilon = 0.5$ the repulsive interactions are now less probable for the given post distribution but still causing a constant flow of users towards the outside. Thus in In the long run this state will end up having two peaks and is unstable at the captured moment. Note how the outside peaks are having a slightly bigger distance. This is simply caused by the Ricker-wavelet weight function being stretched by ε and therefore approximating zero at bigger opinion distances than in the first case.

The right state with one peak is a bit more stable than the former one. There are a few users not within the given distance threshold which is caused by setting $\mu = 0.5$. All users get attracted by the posts and form a peak but constantly shift their opinion by such an amount that the peak does not looses width. Instead it looks like a fluctuation between a certain opinion range that exceeds the distance threshold. Notice how during the simulation two users are being pushed away. This is not enough to form an own peak so they get repelled until they are distant enough and the repulsive values become approximately zero. Of course they are also counted for the stability check.

C. Implementation of the Reddit and Telegram Model

In this part I want to talk about a few technicalities of the implementation of the social media model in Python. Basically the whole code is structured like described in the model overviews in section 4.2.1 and 4.3.1. For the model initialization, I use one 1D-array with length N each for the user opinion, activity and category. Doing it this way optimizes the runtime and is easier to handle in numpy functions instead of using a 2D-array with the shape $(N, 3)$.

For the posts though I had to use a 2D-array with flexible shape $(N_{\text{posts}}, 7)$ as the array gets sorted during every iteration and it would be more difficult to sort one of the seven separate arrays without getting index trouble. The seven attributes of each post are the opinion, score, duration online, upvotes, downvotes, number, views. For the Telegram model the post array is reduced to the shape $(N_{\text{posts}}, 3)$ with the three attributes opinion, duration online and views.

To assign each user an opinion, I draw a random number $B \in (0, 1)$ from a beta distribution with numpy [24] and then calculate $\tilde{B} = 2B - 1$ to receive a value $\tilde{B} \in (-1, 1)$ as the used Ricker-wavelet weight function is symmetrical along the y-axis and moderate opinions are reflected by zero. For the activity assignment the method of inverse transform sampling is used to generate a a random number $U \in (0.1, 1)$ from a bounded power law distribution [25]. The minimal activity is chosen with 0.1 to ensure that even the most inactive users see some posts when browsing.

The categories are chosen by drawing a random number from a uniform distribution with numpy and if the number is below c_0 the user gets assigned to the first category. If the number is between c_0 and c_1 to the second category and if it is above c_1 to the last one. This mechanism is used for generating events in the simulation as well. The last thing that needs to be initialized is a dictionary for each user which consists of all the posts the user has interacted with to ensure that he will not get affected twice by a post.

In the iteration cycle the new posts are created if the user is online and belongs to the right category. Since it would be unrealistic that a user posts every time he is online, I added the posting probability p. So the event of the creation of a new post is only triggered if the activity is greater than a randomly drawn number from a uniform distribution, if another randomly drawn number is smaller than p and the user belongs to the right category. This means user with a greater activity have a bigger chance of submitting content. Greater p instead increase the overall probability of the event.

The new post then is initialized with the users opinion who submits it, one upvote by default, the posts number and one view. The score, duration online and downvotes are zero by default. Furthermore the posts number gets attached to the dictionary entry for this user to ensure he will not get affected by his own post later on.

After going through this process for every user the next step of the iteration cycle is reached and all users browse by new posts. First the post array gets sorted by their age using the numpy function `argsort`. After that the program again checks if a user is online and belongs to the category that browses by new. The browsing process itself is simulated by assigning each user a value which determines the range of the loop that iterates through the post array and applies the opinion model on the user and posts. The value is calculated by multiplying the user activity by the proportionality constant d. E.g. a user with activity $\gamma_i = 0.5$ sees 25 posts during one iteration when $d = 50$.

Inside the loop before applying the opinion update rule it has to be checked that the user has not interacted with the post yet. On top of that if an interaction happens, the rejections for the leaving process are counted and it is decided if the user leaves an up- or downvote. The next step is to calculate the score of every post by using the given formula (4.3) and increasing the duration online by one. Now the post array can get resorted by the score ready for the users to browse by "hot". The browsing and interaction is exactly like before with the only difference that the user category does not need to be checked, only if the user is online.

At last the indexes of the users which leave the subreddit are passed to a function which re-initializes those users by the initial opinion, activity and category distribution. Overwriting the array entries has the advantage of saving computational time but limits the model in not allowing more leaving than joining users. On the other hand it is possible to make additional users join. The parameter ν determines how many users should join at the end of each iteration. In fact $\nu \cdot N$ users will be initialized and added to the array. This can be done using the numpy function `concatenate` which allows to extend pre-allocated arrays.

For the data visualization I wanted to check if the up- and downvote distribution follows a power law. This was done by using the SciPy function `curve_fit`. In order to plot the according double logarithmic histogram, I had to use the numpy function `logspace` to transform the bin size of the histogram.

Bibliography

[1] Weiguo Fan and Michael D. Gordon. The power of social media analytics. *Communications of the ACM*, 57(6):74–81, June 2014.

[2] Katrin Weller. Trying to understand social media users and usage. *Online Information Review*, 40(2):256–264, April 2016.

[3] Ekaterina Zhuravskaya, Maria Petrova, and Ruben Enikolopov. Political effects of the internet and social media. *Annual Review of Economics*, 12(1):415–438, August 2020.

[4] Filippo Menczer, Santo Fortunato, and Clayton A. Davis. *A First Course in Network Science*. Cambridge University Press, January 2020.

[5] Christopher Musco, Indu Ramesh, Johan Ugander, and R. Teal Witter. How to quantify polarization in models of opinion dynamics, 2021.

[6] Mason A. Porter and James P. Gleeson. Dynamical systems on networks: A tutorial, 2014.

[7] Andreas Flache, Michael Mäs, Thomas Feliciani, Edmund Chattoe-Brown, Guillaume Deffuant, Sylvie Huet, and Jan Lorenz. Models of social influence: Towards the next frontiers. *Journal of Artificial Societies and Social Simulation*, 20(4), 2017.

[8] Wendy Wood. Attitude change: Persuasion and social influence. *Annual Review of Psychology*, 51(1):539–570, February 2000.

[9] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, August 2001.

[10] Raymond S. Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, 2(2):175–220, June 1998.

[11] Elaine Hatfield, John T. Cacioppo, and Richard L. Rapson. Emotional contagion. *Current Directions in Psychological Science*, 2(3):96–100, June 1993.

[12] Milton E. Rosenbaum. The repulsion hypothesis: On the nondevelopment of relationships. *Journal of Personality and Social Psychology*, 51(6):1156–1166, December 1986.

[13] Michela Del Vicario, Antonio Scala, Guido Caldarelli, H. Eugene Stanley, and Walter Quattrociocchi. Modeling confirmation bias and polarization. *Scientific Reports*, 7(1), January 2017.

[14] Ro'ee Levy. Social media, news consumption, and polarization: Evidence from a field experiment. *American Economic Review*, 111(3):831–870, March 2021.

[15] Michael Mäs, Andreas Flache, and James A. Kitts. Cultural integration and differentiation in groups and organizations. In *Perspectives on Culture and Agent-based Simulations*, pages 71–90. Springer International Publishing, October 2013.

[16] Yanghua Wang. Frequencies of the ricker wavelet. *GEOPHYSICS*, 80(2):A31–A37, March 2015.

[17] Reddit by the numbers. `https://www.redditinc.com/press`. Accessed: 2022-12-20.

[18] Francesco Cauteruccio, Enrico Corradini, Giorgio Terracina, Domenico Ursino, and Luca Virgili. Investigating reddit to detect subreddit and author stereotypes and to evaluate author assortativity. *Journal of Information Science*, 48(6):783–810, December 2020.

[19] Reddit archive. `https://github.com/reddit-archive/reddit`. Accessed: 2022-12-21.

[20] Noella Edelmann. Reviewing the definitions of "lurkers" and some implications for online research. *Cyberpsychology, Behavior, and Social Networking*, 16(9):645–649, September 2013.

[21] Lev Muchnik, Sen Pei, Lucas C. Parra, Saulo D. S. Reis, José S. Andrade Jr, Shlomo Havlin, and Hernán A. Makse. Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Scientific Reports*, 3(1), May 2013.

[22] World pupulation review. `https://worldpopulationreview.com/country-rankings/telegram-users-by-country`. Accessed: 2022-12-21.

[23] Scipy signal documentation. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html`. Accessed: 2022-12-12.

[24] Numpy documentation. `https://numpy.org/doc/`. Accessed: 2022-12-12.

[25] Bounded power law. `http://up-rs-esp.github.io/bpl/`. Accessed: 2022-12-14.

Acknowledgements

I thank Dr. Joao Pinheiro Neto, Riccardo Carlucci and Robin Danek for their exchange of ideas, providing me a lot of useful comments on my work and helping me to realize this project.

Erklärung Ich versichere hiermit, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die von mir angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder sinngemäß aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht. Die Richtlinien zur Sicherung der guten wissenschaftlichen Praxis an der Universität Göttingen wurden von mir beachtet. Eine gegebenenfalls eingereichte digitale Version stimmt mit der schriftlichen Fassung überein. Mir ist bewusst, dass bei Verstoß gegen diese Grundsätze die Prüfung mit nicht bestanden bewertet wird.

Göttingen, 2. Januar 2023

(Vincent Christoph Brockers)