



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

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

Project Report

---

**Filter Bubbles in Opinion Formation**

---

Adrian Hartmann, Beat Nairz,  
Leo Widmer, Luc Stoffer

Zürich  
December 2019

## Agreement for free-download

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

Adrian Hartmann



Beat Nairz



Leo Widmer



Luc Stoffer



# Contents

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Individual contributions</b>	<b>4</b>
<b>3</b>	<b>Introduction and Motivations</b>	<b>5</b>
<b>4</b>	<b>Description of the Model</b>	<b>5</b>
4.1	Model Network . . . . .	5
4.2	Opinion Evolution and Confidence . . . . .	6
4.3	Following and Unfollowing – Filter Bubbles . . . . .	6
<b>5</b>	<b>Implementation</b>	<b>7</b>
5.1	Initial state . . . . .	7
5.1.1	Opinion . . . . .	7
5.1.2	Trust . . . . .	8
5.1.3	Confidence . . . . .	8
5.2	Updating the network . . . . .	8
5.2.1	Updating opinion . . . . .	8
5.2.2	Updating trust . . . . .	9
5.2.3	Updating follower connections . . . . .	9
<b>6</b>	<b>Simulation Results and Discussion</b>	<b>10</b>
6.1	Initial Setup . . . . .	10
6.2	Influence of $p_{follow}$ and $p_{unfollow}$ . . . . .	11
6.2.1	Varying $\beta_u$ . . . . .	11
6.2.2	Varying $\beta_f$ . . . . .	12
6.3	Trust stability . . . . .	13
6.4	Merging of Opinion Bubbles . . . . .	13
<b>7</b>	<b>Summary and Outlook</b>	<b>14</b>
<b>A</b>	<b>Declaration of Originality</b>	<b>16</b>
<b>B</b>	<b>Further Analysis</b>	<b>17</b>
B.1	Influence of the confidence distribution . . . . .	17
B.1.1	Changing $\mu$ . . . . .	17
B.1.2	Changing $\sigma$ . . . . .	18

# 1 Abstract

In this project we study the formation of filter bubbles in social media. To tackle the topic, we create a model to predict the formation of agent opinions and the connections between different agents. We consider a weighted directed network of vertices. A vertex creates connections, "follows", other vertices with a probability depending on the opinion difference. It severs a connection, "unfollows", with a probability depending on the connection weight. With this model we run simulations - written in Python - for multiple parameter configurations and analyse the outcome in relation to bubble formation. For lower following probability for large opinion differences and higher unfollowing probability for low connection weight, filter bubbles are found to form more frequently. We observe behaviour for merging of opinion bubbles and find that there is a critical separation between bubbles for which they do not converge.

# 2 Individual contributions

All team members contributed equally.

*"[Technology such as social media] lets you go off with like-minded people, so you're not mixing and sharing and understanding other points of view ... It's super important. It's turned out to be more of a problem than I, or many others, would have expected."*

---

— Bill Gates 2017 in Quartz<sup>[1]</sup>

### 3 Introduction and Motivations

Inspired by social media and modern phenomena like influencers and Fake News, our goal was to study the formation of opinions. For this purpose we mainly adapted parts of the paper *Opinion Dynamics and Bounded Confidence* by R. Hegselmann and Ulrich Krause.

As the formation of opinions is a rather complex subject with many different aspects, we restricted our model to the formation of filter bubbles, especially in social media. With the rising impact of the internet on our all lives, being stuck in a filter bubble is as likely as never before. But first we need to talk about the troubles with Bubbles.

*"A filter bubble – a term coined by Internet activist Eli Pariser – is a state of intellectual isolation that allegedly can result from personalized searches when a website algorithm selectively guesses what information a user would like to see based on information about the user, such as location, past click-behavior and search history."* — Wikipedia<sup>[2]</sup>

If you are in a filter bubble, you will get isolated from other opinions. You will not encounter the point of view of the others anymore and at worst, you will only get positive feedback for your opinion, since everyone around you is thinking the same. This boost of your confidence can lead to big troubles. Even if you encounter an other opinion at some point, your confidence will just deny it. Some thesis even claim, that it is very likely that filter bubbles already have influenced elections i.e. for the president of the United States<sup>[3]</sup>.

In this project we went on a journey with numerical simulations on which we tried to find the cause of these bubbles and a way to prevent them.

## 4 Description of the Model

### 4.1 Model Network

We want to consider a network  $V$  of agents, called vertices. The structure should reflect the connection structure of social media.

There are two main types of networks; directed, where a vertex  $u$  can be connected to  $v$  without  $v$  being connected to  $u$ , or undirected. Both types are present for popular social media platforms. For example, Instagram and Twitter have a directed follower - following relationship, while on Facebook being 'Friends' is undirectional. A simple illustration of this difference can be seen in Figure 1.

A network is weighted, when the connections are assigned weights.

The approach we choose is describing the system with a weighted directed network. Suggestively, we say that  $u$  is **following**  $v$ , if  $u$  is connected to  $v$ . A vertex  $v$  is then influenced only by the vertices it is following. The weight that is assigned to the connection

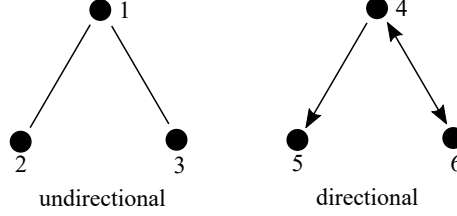


Figure 1: Schematic of directed versus undirected networks.

is called the **trust** of  $u$  in  $v$ .

The trust measures how much  $u$  is influenced by  $v$ , and is a dynamic variable.

## 4.2 Opinion Evolution and Confidence

The opinion of a vertex evolves in a similar way as is described by Hegselmann and Krause<sup>[4]</sup>. A vertex adheres to his opinion with a certain *degree*. We call this degree the **confidence** that  $u$  has in its own opinion,  $c(u)$ . The evolution of the opinion is then calculated as a weighted mean over the opinions of the vertices that  $u$  follows. The confidence induces a latency in the change of opinion; a high confidence means that the opinion change is smaller. This is explained in more detail in Section 5.

For a more realistic model, an alternative interpretation of a vertex' opinion is the opinion the vertex perceives to be the consensus. In a filter bubble, an agent believes that everybody holds a similar opinion to his own.

## 4.3 Following and Unfollowing – Filter Bubbles

The connections of the network are dynamically updated.

A vertex  $u$  can sever a connection to  $v$ , as well as create a new one with  $v'$ . We say  $u$  **unfollows**  $v$  and **follows**  $v'$ .

We model the following and unfollowing process to be stochastic. The probability of  $u$  unfollowing  $v$  is dependent on the trust that  $u$  has in  $v$ . If the trust is low, unfollowing becomes more likely.

The probability of  $u$  following  $v'$  is modeled to be dependent on the opinion difference between the two vertices.

This step is where filter bubbles can be implemented and is therefore the 'heart' of this model. By regarding the probability of  $u$  following  $v$  as the probability that the social media platform *proposes* to  $v$  to follow  $u$ , and this probability diminishes if their opinions are further apart, this can lead to a separation of opinions, where one group does not in

any way communicate with the other. For a visualisation, see Figure 2. This approach was chosen to model algorithms by social media, which analyse a user’s opinions and personalise content and user suggestions to predominantly reflect these opinions.

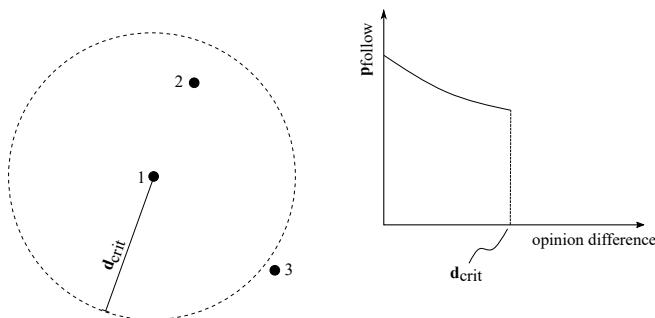


Figure 2: Three vertices with their respective distances visualising the opinion difference. A following probability  $p_{\text{follow}}$  which depends on the opinion difference between the vertices can result in filter bubbles. In this case, vertex 1 cannot be influenced by 3, since the probability to follow 3 is 0.

## 5 Implementation

### 5.1 Initial state

Our network  $(V, A)$  with the set of vertices  $V$  and set of edges  $A$  will be represented as an adjacency list.

In the initial state our network has no connections. By running the simulation for a while, the network will naturally build connections between mostly like-minded agents. This way we do not have to make initial artificial random connections which may be relatively unrealistic.

All our functions, namely opinion, trust and confidence, map to  $[0, 1]$  for reason of simplicity. They are concretely implemented as follows.

#### 5.1.1 Opinion

$$o : V \rightarrow [0, 1]$$

Each vertex will have an opinion value.

In our simulation we give uniformly random initial opinions but one could easily do it with an other distribution or give deterministic values.

### 5.1.2 Trust

$$w : A \rightarrow [0, 1]$$

In the adjacency list, each vertex will have a set of vertices it follows. We will store a trust value for each of these vertices. If a vertex  $u$  is not connected to another vertex  $v$ , i.e.  $(u, v) \notin A$ , we implicitly assume the value to be 0.

If a new connection from vertex  $u$  to  $v$  is built, the trust value will get initialized according to

$$w(u, v) \leftarrow \frac{1}{\max\{1, \deg^+(u)\}}$$

where  $\deg^+(u)$  is the number of vertices  $u$  follows (including the new one).

### 5.1.3 Confidence

$$c : V \rightarrow [0, 1]$$

Each vertex will have a confidence value.

These values will not change over time. They are initialized randomly over a truncated normal distribution with bounds  $a = 0.01, b = 0.99$  and different values for the mean  $\mu$  and standard deviation  $\sigma$ .

## 5.2 Updating the network

Our network will change over time. In the simulation we iterate over a defined number of time steps. Therefore we need to update our functions (opinion and trust) and the connections between agents in each time step of our simulation. The next sections describe how this is implemented.

### 5.2.1 Updating opinion

The opinion of a vertex  $u$  is updated as follows.

$$o(u) \leftarrow c(u) \cdot o(u) + (1 - c(u)) \cdot \overbrace{\sum_{(u,v) \in A} (w(u, v) \cdot o(v))}^{\text{other opinions}}$$

Firstly, we determine how opinions of other agents can influence a certain agent  $u$ . The only opinions that the agent  $u$  considers are the opinions of those it follows. The vertex is more heavily influenced by vertices it trusts more. Therefore, we weight each opinion by the respective trust. Then, we take the weighted mean of opinions.

Secondly, we also take the opinion of  $u$  itself into account. This is done by multiplying the opinion of  $u$  by its confidence. Then we add to this the weighted sum of the other



opinions multiplied by one minus the confidence (susceptibility). This means that, the higher the confidence of  $u$ , the more it takes its own opinion into account and ignores the opinion of the ones it follows.

### 5.2.2 Updating trust

For each vertex  $u$  we update the trust in a vertex  $v$  it follows according to

$$w(u, v) \leftarrow w(u, v) + (1 - \alpha) \cdot (c(v) - |o(u) - o(v)|)$$

for  $0 < \alpha < 1$ . We call  $\alpha$  the **trust stability**. The higher the trust stability, the slower the trust values change.

How the trust of  $u$  in  $v$  gets changed is dependent on the confidence of  $v$  and the opinion difference between the two agents. The more confident  $v$  is the more trust it will get from its followers, e.g.  $u$ . But the higher the opinion difference is the less trust  $u$  will have in  $v$ .

Furthermore, we normalize the trust values of each set of neighbours. We do this because we want to do weighted sums with the trust values of a vertex (see Updating opinion).

### 5.2.3 Updating follower connections

In this section we change the actual structure of the network.

In each time step we update the vertices a vertex follows. This is done for each vertex  $u$  by the following procedure.

Firstly, we do the unfollowing. This is simply done by iterating through all agents  $v$  which  $u$  follows and unfollowing them with a certain probability

$$\Pr[u \text{ unfollows } v] = \exp(-\beta_u \cdot w(u, v)), \beta_u \in \mathbb{R}_{\geq 0}.$$

Secondly, we build some new connections. We define an upper bound for how many new agents someone can follow in a time step. Then we iterate through all vertices  $v$  in a random order. If  $u$  does not yet follow  $v$  then  $u$  will follow it with a certain probability

$$\Pr[u \text{ follows } v] = \exp(-\beta_f \cdot |o(v) - o(u)|), \beta_f \in \mathbb{R}_{\geq 0}.$$

We see that  $\Pr[u \text{ follows } v]$  depends on the opinion difference between the two agents. This should represent the algorithm of any social network that recommends new people to a user. The idea is that we want users to find people which they find interesting.

The probability  $\Pr[u \text{ unfollows } v]$  is dependent on the trust that agent  $u$  has in  $v$ . The higher the trust the lower the probability of unfollowing.

## 6 Simulation Results and Discussion

The goal of the following section is to investigate the results obtained by changing the basic parameters of the model. These are the values  $\beta_u$  and  $\beta_f$ , which determine the probabilities for following and unfollowing, *confidence\_mean* and *confidence\_std*, which determine the probability distribution of the confidence and *trust\_stability*, which affects how fast the trust values change.

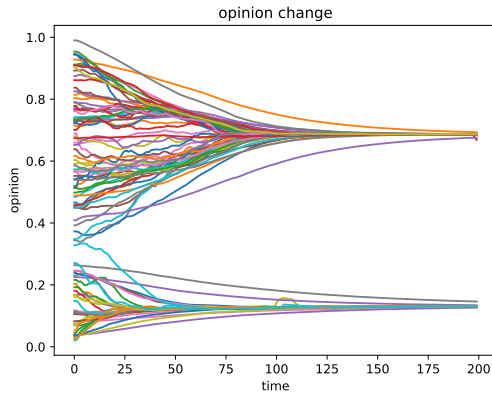
### 6.1 Initial Setup

As the initial setup the basic parameters for the model are set as follows:

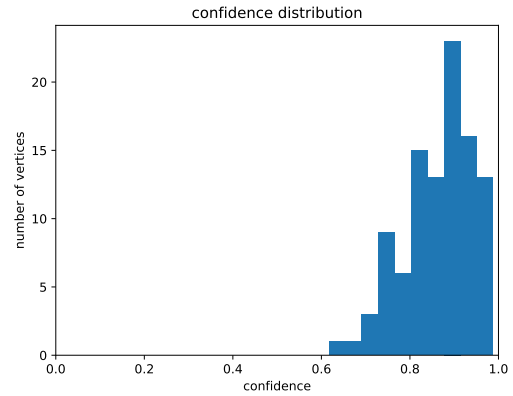
```
bu = 20
bf = 20
confidence_mean = 0.9
confidence_std = 0.1
trust_stability = 0.99
```

The initial opinions are uniformly distributed over the interval  $[0, 1]$ .

This initial setup is used for all further investigations of the model. Running the simulation on the initial setup results in two filter bubbles.



(a) Opinions ending up in two filter bubbles



(b) Distribution of confidence

Figure 3: In (3a), the changes of opinions, resulting in two filter bubbles at 0.65 and 0.18, are plotted over time. (3b) displays the distribution of confidence.

Different simulations with the same initial model differ only in the opinions around which the filter bubbles build up.

## 6.2 Influence of $p_{follow}$ and $p_{unfollow}$

The amount of filter bubbles at the end of a simulation is highly dependent on  $p_{follow}$  and  $p_{unfollow}$ . In the model, these two values depend on the parameters  $\beta_f$  and  $\beta_u$ . How these two probabilities are calculated is described in subsection 5.2.3.

Increasing  $\beta_u$  means that the the probability to unfollow some other vertex gets smaller if the trust in this vertex has not changed. In other words the trust in another vertex has to decrease in order to unfollow this vertex if  $\beta_u$  has increased.

Increasing  $\beta_f$  decreases the probability to follow some other vertex if the opinion difference between the two vertices has not changed. In other words, the opinion difference between two vertices has to get smaller to accept a follow suggestion if  $\beta_f$  has increased.

### 6.2.1 Varying $\beta_u$

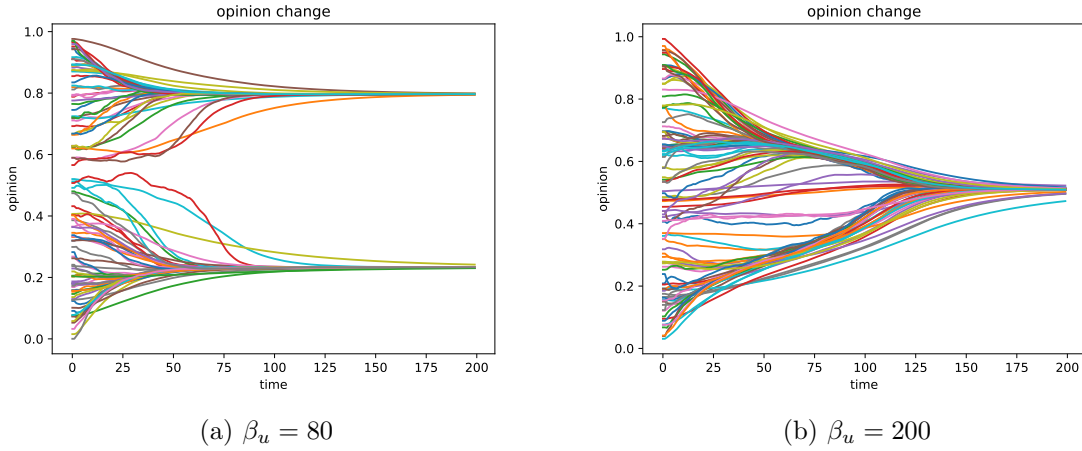


Figure 4: Increasing  $\beta_u$  to 80 (4a) respectively 200 (4b) while leaving  $\beta_f$  at 20.

Increasing  $\beta_u$  while leaving  $\beta_f$  at a value of 20 leads to the fact that it is less likely to unfollow another vertex. Hence the opinions settle in two respectively one filter bubble. Since one agent is influenced by every other agent it follows, this result makes perfect sense. After some time every agent follows every other vertex whose opinion is in some range. In this case the range depends on the value of  $\beta_u$ . Increasing  $\beta_u$  also increases this range. At the end all agents in one range agree to the same opinion and therefore a filter bubble comes into existence. For  $\beta_u = 80$  there are two such ranges and for  $\beta_u = 200$  there is only one.

### 6.2.2 Varying $\beta_f$

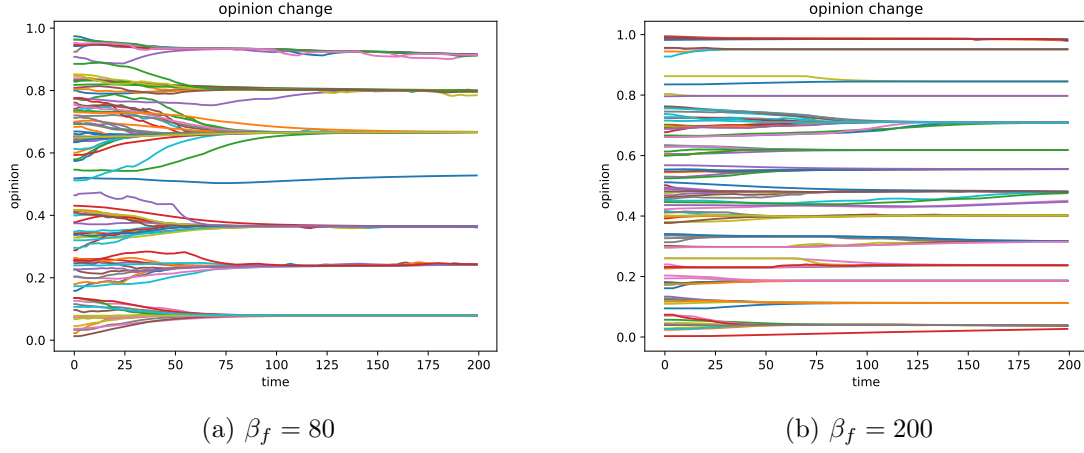
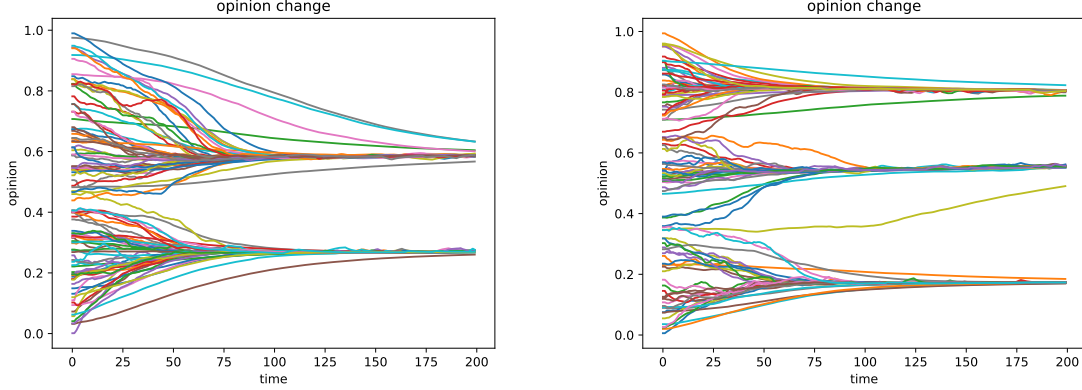


Figure 5: Increasing  $\beta_f$  to 80 (5a) respectively 200 (5b) while leaving  $\beta_u$  at 20.

Increasing  $\beta_f$  leads to a much lesser tolerance of the agents, meaning that the opinion difference between an agent and a suggested other agent has to be much smaller for the agent to accept the suggestion. In the end, this as well leads to ranges in which it is possible for agents to follow each other. However, in contrast to  $\beta_u$ ,  $\beta_f$  is direct proportional to the amount of ranges. In conclusion, increasing  $\beta_f$  leads to more filter bubbles.

If  $\beta_f$  is too large, the agents tend to not accept any suggestions and therefore keeping their initial opinion.

### 6.3 Trust stability



(a) Changes in opinion for  $trust\_stability = 0.5$  (b) Changes in opinion for  $trust\_stability = 0.5$

Figure 6: Changing the value of  $trust\_stability$  can cause slower convergence of opinions (6a) or it can even increase the amount of filter bubbles (6b).

Reducing the  $trust\_stability$  can have two effects on the opinion formation. In general the opinions converge slower. However, it is also possible that the amount of filter bubbles increases. That is because if the trust stability is smaller, the trust values for the different agents change more.

### 6.4 Merging of Opinion Bubbles

We can observe critical behaviour for the merging of two opinion strands. When initialising, opinions are grouped into two strands, separated by a certain number (separation). The probability of the strands to merge is recorded for different separations and different coefficients  $\beta_f$ , determining the following probability

$$\Pr[u \text{ follows } v] = \exp(-\beta_f |o(u) - o(v)|).$$

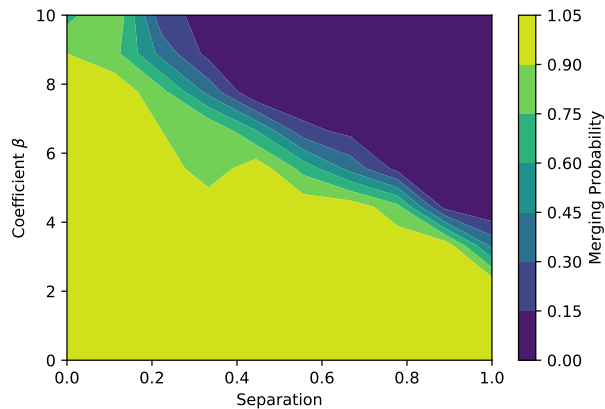
Results are shown in Figure 7. We observe that there is a critical strand separation, at which the probability of the strands to merge falls from 1 to 0 quickly. For larger  $\beta_f$ , this critical separation is lower. For  $\beta_f = 5$ , the merging probability is fitted with a Fermi-Dirac distribution function  $f(s) = \frac{1}{\exp((s-a)/b)+1}$ . The fitted parameters are

$$a_{\text{fit}} = 0.711 \pm 0.007, \quad b_{\text{fit}} = 0.048 \pm 0.007.$$

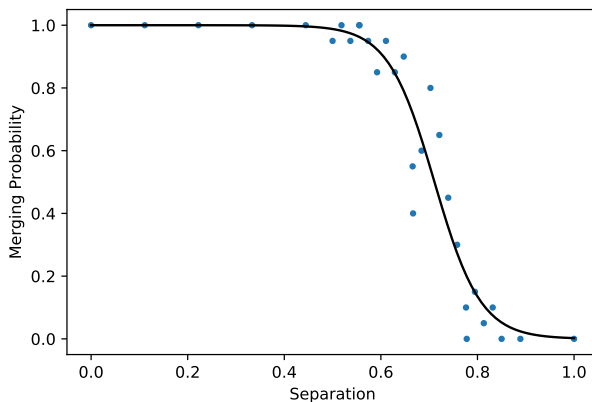
In the "thermodynamic" limit, the dependence might be better described by a function similar to percolation models,

$$\Pr[\text{merging of strands}] \propto 1 - |s - s_{\text{crit}}|^\gamma, \quad s > s_{\text{crit}},$$

where  $s$  is the strand separation, and  $s_{\text{crit}}$  is a critical separation. A thorough investigation of such a relation is limited by computation times.



(a) The merging probability for different following probabilities, characterised by  $\beta_f$ .



(b) Fitted merging probability for  $\beta_f = 5$ .

Figure 7: The probability of two strands to merge for different initial strand separations. In 7a, the probability is plotted for different  $\beta_f$  coefficients. In 7b, the curve is shown in more detail for  $\beta_f = 5$  and fitted with a function of the form  $f(s) = \frac{1}{\exp((s-a)/b)+1}$ .

## 7 Summary and Outlook

We have implemented a relatively simple agent-based model to study filter bubbles in opinion formation, motivated by a social media phenomenon.

We were able to extract qualitative statements about the effect of various parameters on the formation of filter bubbles.

Notably, we studied the effect of different following and unfollowing probabilities. We found that for low following probabilities for large opinion differences, the formation of filter bubbles becomes more likely. This is because communication between vertices of different opinions becomes unlikely. A high unfollowing probability for low trust also results in formation of bubbles, as a vertex is more likely to unfollow a vertex it disagrees with. By varying model parameters such as trust stability and confidence, we were able to manipulate the convergence rate of filter bubbles.

We observed behaviour for the merging of different opinion bubbles, and found that there is a region critical opinion separation, where the merging probability falls sharply from 1 to 0. This is reminiscent of basic percolation behaviour, and could be studied in more detail.

It would also be interesting to consider different network structures, where a vertex does not have the opportunity to in principle communicate with any other vertex. Our approach was chosen to model social media, however, geographic separation, age, and other factors also have an effect on the connections in realistic social media networks. If other network structures were to be considered, we would expect to see more complex behaviour.

It is a possibility to use real-world data and study, which aspects of the dynamics are captured by our model. While it is unlikely that the model is able to realistically describe the system, it might give valuable insights, both into social media dynamics as well as improvement of the model.

## References

- [1] Kevin J. Delaney (February 21, 2017). "Filter bubbles are a serious problem with news, says Bill Gates". Quartz. Retrieved 15 November 2019.
- [2] Wikipedia, Wikimedia Foundation. Last accessed 15 November 2019.  
[en.wikipedia.org/wiki/Filter\\_bubble](https://en.wikipedia.org/wiki/Filter_bubble)
- [3] Medium, A Medium Corporation. Last accessed 20 November 2019.  
[medium.com/@willrinehart/the-election-of-2016-and-the-filter-bubble-thesis-in-2017-51cd7520ed7](https://medium.com/@willrinehart/the-election-of-2016-and-the-filter-bubble-thesis-in-2017-51cd7520ed7)
- [4] Hegselmann, R., Krause, U., JASS **5**, 3, 2002,  
*Opinion Dynamics and Bounded Confidence*

## A Declaration of Originality



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

### Declaration of originality

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

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

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

**Title of work** (in block letters):

Filter Bubbles in Opinion Formation

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Hartmann  
Widmer  
Nairz  
Luc

**First name(s):**

Adrian  
Leo  
Beat  
Stoffer

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

**Place, date**

Zürich, 02.12.2019

**Signature(s)**

A. Hartmann  
L. Widmer  
Beat Nairz  
L. Stoffer



## B Further Analysis

### B.1 Influence of the confidence distribution

In the initial setup the confidence values get initialized randomly over a truncated normal distribution with the mean  $\mu = 0.9$  and standard deviation  $\sigma = 0.1$ .

This section determines the influence of changing  $\mu$  and  $\sigma$  for this distribution.

#### B.1.1 Changing $\mu$

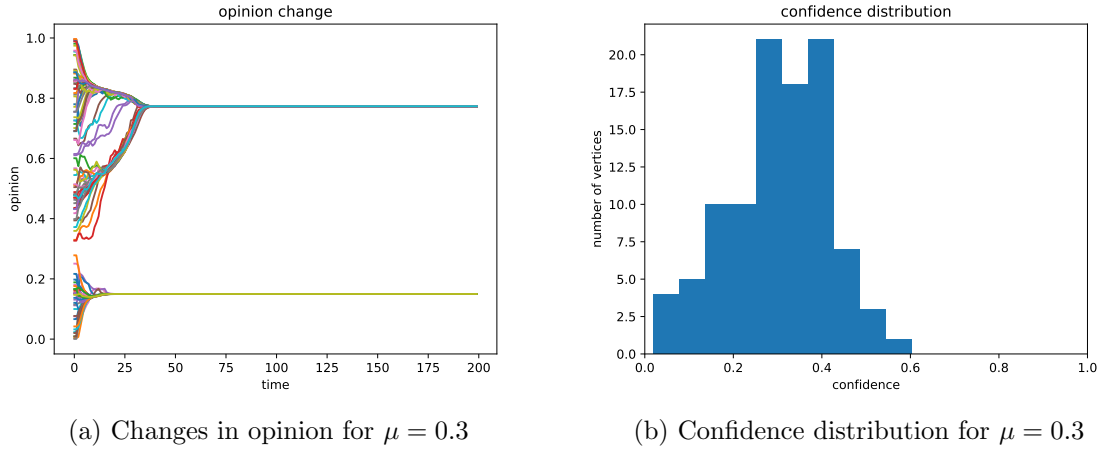
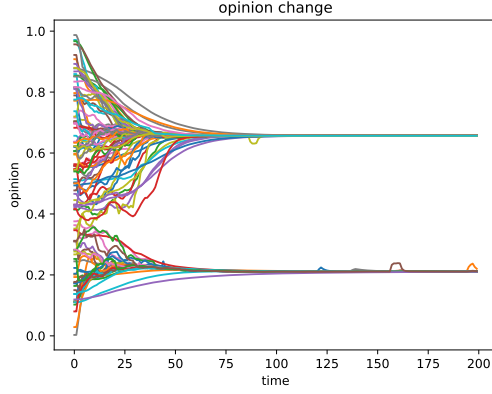


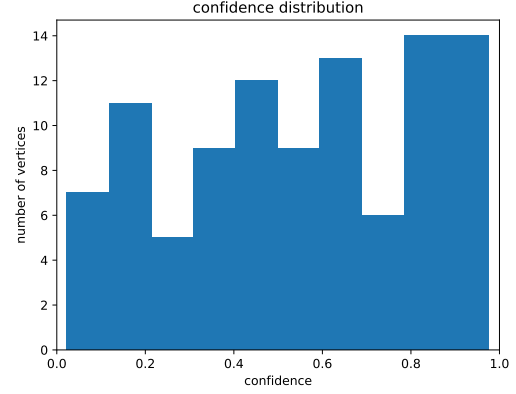
Figure 8: Changing the value of  $\mu$  from 0.9 to 0.3

Changing  $\mu$  from 0.9 to 0.3 leads to lower confidence of the agents. And therefore the agents change their opinions faster, the opinions converge faster and the filter bubbles get established faster.

### B.1.2 Changing $\sigma$



(a) Changes in opinion for  $\sigma = 0.7$



(b) Confidence distribution for  $\sigma = 0.7$

Figure 9: Changing the value of  $\sigma$  from 0.1 to 0.7

Changing the value of  $\sigma$  while keeping the value of  $\mu = 0.9$  influences the speed of convergence. Because the higher  $\sigma$ , the more agents there are, which have little confidence. Furthermore, these agents change their opinion faster and therefore support a faster convergence of opinions.