Opinion polarization modeling in social networks

Table of Contents

Acknowledgement

We are really grateful to prof. Henrique Ferraz de Arruda who was the first author of [1], for constantly supporting us through mails and giving us idea to work on this the project to us in an easy to understand manner which has helped us complete our project easily and effectively on time.

We are also dearly obliged to Mr. Juan Fernández-Gracia [2] for giving us a supporting knowledge to work on this project which has provided valuable information about our Voter Model.

We have also consulted few more papers [3], [4].

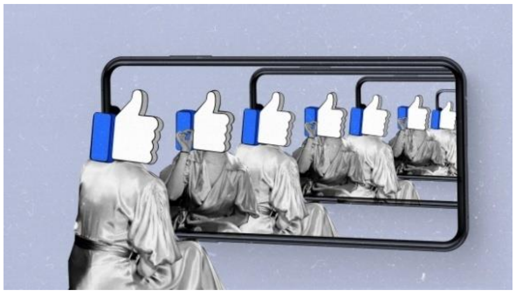
I would also like to thank our department of college and everyone who gave us the chance and inspired and helped in approaching and completing this project.

Introduction

Domain Description:

With the advent of the internet, many different online social networks have been created. In order to understand the impact of these networks on the users' opinions, different models have been proposed in the papers [4], [5].

Part of these studies considers that the dynamic is executed on a network structure, in which the nodes and edges represent people and their friendship, respectively. Several distinct characteristics of social networks have been studied in order to understand opinion dynamics. For example, we considered that when two or more people have the same opinion, it is more likely for them to convince others. In general, these studies consider static network structures. However, other studies have taken into account that people can change their friendship. In this case, edge rewiring is allowed, with might give rise to groups of connected people with similar opinions, called **Echo Chambers**.



One essential characteristic of these dynamics is how to represent the opinion. In many cases, the opinions are expressed only for two possible states. In other cases, a varied number of categories, or vectors can also describe the opinions. Another option is to express opinions as a continuous variable, which can express problems regarding negotiations. In this case, opinions are not categorical, and the individuals can have intermediate opinions. Promising results have been obtained from this type of dynamic. For instance, in, results obtained from simulations were found to be similar to the scenario observed in online social networks.

Although in real social networks, individuals typically have lots of friends, the authors considered that a person is not capable of interacting with lots of other individuals. For this reason, they adopted network models with low average degrees. In order to constrain the interactions between individuals, we considered two complementary mechanisms. The first represents the user's action of posting pieces of information, henceforth called **Post Transmission**. In contrast to other approaches, individuals can post something different from their own opinions. We also simulate the individuals' information transference, a mechanism that chooses if the data shall be delivered to other users. We named this mechanism as **Post Distribution**. This step mimics how social network algorithms could manage posts. Moreover, since the nature of the posts received could change opinions positively or negatively, we consider two kinds of interaction*:* *attraction* and *repulsion*, respectively. Finally, we also allow individuals to rewire connections for the cases in which the post repulses the individual’s opinion.

We analyzed the results obtained by employing two distinct and complementary measurements. The first consists of analyzing the resulting opinion distributions. We quantify how polarized and balanced the opinion distributions are. However, more information can be obtained if the network structure is considered. we measured the relationship between the node opinion and the average of the friends’ opinions.

Additionally, the bimodality of the opinion distribution does not guarantee that the dynamic would converge to echo chambers. We also compared our dynamic with networks obtained from Twitter. Interestingly, we found similarities between the results obtained with the proposed dynamics, including the level of bimodality and echo chambers.

Motivation:

Social networking is changing the way the world is doing everything, from the way people get information to the way people communicate, and, most importantly, the way people interact. Social media allow people to exchange ideas and opinions. Social media is different from traditional mainstream media in that their content can create everybody as well contribute into it or comment on it.

On this day, everyone is on social media and a part of the social network. An individual can come up with a post whose opinion can be attracted or repulsed. Furthermore, individuals that can be attracted to the post's opinion through making new friendships or repulsed by it can change their friendship through rewiring. With our project, Voter Model, we are attempting to comprehend only the randomness of human behavior. That is how things actually work with some dynamics. Due to these reasons, this field of work motivated us and it was interesting for us to do this project.

Scope:

The popularity of social networks has exploded in the past few years. There are several networks which currently have millions of users: Facebook, Myspace and several others. It is estimated that currently at least 300 million people use social networks, and that number is still growing. Social networks have existed in one form or another throughout human history, between families and friends. The Internet expanded this networking even before the current social networks: e-mails, newsgroups and discussion forums on the Internet have formed networks around specific topics. They have not been officially classified as social networks as they are defined today, but in practice they work on the same principles as the current networks. In future, it is very obvious that this topic would be definitely an interesting one and would worth working on it.

Background/review of related work

• Is the Voter Model a Model for Voters? - Juan Fernández-Gracia, Krzysztof Suchecki

The voter model has been studied extensively as a paradigmatic opinion dynamics model. However, its ability to model real opinion dynamics has not been addressed. We introduce a noisy voter model (accounting for social influence) with recurrent mobility of agents (as a proxy for social context), where the spatial and population diversity are taken as inputs to the model.

• Statistical physics of social dynamics - Claudio Castellano, Santo Fortunato, Vittorio Loreto

The concept that many laws of nature are of statistical origin is so firmly grounded in virtually all fields of mod ern physics that statistical physics has acquired the status of a discipline on its own. Given its success and its very general conceptual framework, in recent years there has been a trend toward applications of statistical physics to interdisciplinary fields as diverse as biology, medicine, information technology, computer science, etc. In this context, physicists have shown a rapidly growing interest in a statistical physical modeling of fields patently very far from their traditional domain of investigations Chakrabarti et al., 2006; Stauffer, Moss de Oliveira, de Oliveira, et al., 2006. In social phenomena, the basic constituents are not particles but humans, and every individual interacts with a limited number of peers, usually negligible compared to the total number of people in the system.

• Voter Model on Signed Social Networks - Yanhua Li, Wei Chen, Yajun Wang and Zhi-Li Zhang

Online social networks (OSNs) are becoming increasingly popular in recent years, which

have generated great interest in studying the influence diffusion and influence maximization with applications to online viral mar keting. Existing studies focus on social networks with only friendship re lations, whereas the foe or enemy relations that commonly exist in many OSNs, e.g., Epinions and Slashdot, are completely ignored. In this paper, we make the first attempt to investigate the influence diffusion and influence maximization in OSNs with both friend and foe relations, which are modeled using positive and negative edges on signed networks. In particular, we extend the classic voter model to signed networks and analyze the dynamics of influence diffusion of two opposite opinions.

Dynamics description and methodology used behind the analysis:

In simple words, here, at the very beginning of our analysis, we consider a random post is uploaded by some unknown on any social media platform which is based on any random topic no matter what it is but somehow a bit controversial or something on which a debate would be much probable on that particular platform. After it became visible on everyone's news feed, we are introducing our first dynamics or we can say an operation which is named as "**Post Transmission**". Post transmission is nothing but the mechanics by which it is determined if the post is reaching to an individual and once it is reached to some people, we bring our second dynamics named as "**Post Distribution**" which is performed by the individuals who look forward to share the post among his connections or to the friends in simple terms trying to convince his/her own beliefs about the post. Now, the question arises if the connected individuals would get convinced or not and due to this particular reason at this step, two important dynamics start to play their role which are "**Post Attraction**" and "**Post repulsion**" which means the friends of that individual can get attracted or totally repulsed depending on their own thought process. Up to this, there was nothing crisp about the dynamics and in order to bring some new ideas we introduced two more steps which can take us to more interesting turn in this analysis. As, we have mentioned this is a social networking analysis and being a stochastic network, anythingcan happen anytime and moreover, human nature is the most unpredictable thing ever. Therefore, after attraction and repulsion people may change their friendship (i.e. they can unfriend people or they can make new friends over there) due to the difference between their opinions which we called "**Rewiring**".

New section: By analysing and watching the real scenario of social media these days, we are totally aware of some unethical things happening over there such as hacking,spamming, creating fake ids,etc. One of the most frequent things these days are creating fake ids by some users which are nothing but spam accounts perfuming some fishy actions throughout. That's why we brought a new idea or an additional dynamics called "Spamming".

In mathematical or technical terms, the above process will move forward as follows

A real number is associated with each user representing its opinion. The dynamics starts with a user that has contact with a random opinion, and, according to a given probability function, this individual can post this opinion. This step is henceforth called post transmission. In the next step, called post distribution, another probability function is employed to select the user's friends that could see the post. Post transmission and distribution represent the user and the social network algorithm, respectively. If an individual has contact with a post, its opinion can be attracted or repulsed. Furthermore, individuals who are repulsed can change their friendship through a process called “rewiring“. These steps are executed various times until the dynamics converge.

(I) **Proposed framework**:

Our model is based on a social network, in which the users (individuals) produce a posts.In the following, the social network algorithm (post distribution) selects the neighbors (friends)that will receive the post. Finally, the opinions of the selected neighbors are changed, andthe users that strongly disagree can change their friend-ship (rewiring). Each node, I, represents an individual that stores an opinion, bi, as a real number, in which is bi∈ [-1,1]. The network edges represent the individual's friendships. The dynamics start with the opinions randomly initialized by following uniform probability distribution. For each iteration, a node ‘I’ is randomly selected, and a post P is created (see Figure 1 (a)). In order to define the post's opinion, a number,θ, is randomly generated with uniform probability θ∈ [-1,1]. In the following, a transmission probability, Pt(x), is used to define if the individual’i’ will post P. This probability function is computed according to the difference between the post and the individual's opinion (x= | θ – bi |). The functions are employed in this analysis are described in the section ‘A’. In the following, if the individual ‘i’ produces a post (Figure 1 (c)), there is a probability function,Pd(x), defined for all of the ‘i’ neighbours to receive the information. In this case, the function is calculated for all edges,(i,j), connected to ‘i’, where (x = | bi – bj |). One example of this action is shown in Figure 1 (d) and (e). This action is associated with how the social network algorithm acts. The probability functions are shown in Section ‘B’. After this action, another probability could be associated with the individuals to define if they are active in the social network. However, here we consider this probability as one. In other words, we considered that the users take a look at all received posts. The opinions of the individuals that receive the post can be attracted or repulsed. More specially, for each individual,j, the probability of being attracted is

- ε j (θ, bj)= 1 - | θ - bj | / 2 ………… (1)

As observed, people update their opinions on subjects after interacting, or in a discussion, and can become more polarized while doing so. Thus, in our model, if the individual is attracted, its opinion bj turns to be bj + Δ when x = | θ – bj |, otherwise bj turns to be bj - Δ. High values of ‘Δ’ strongly affect the dynamics because it leads the distributions to be less well-defined.Furthermore, with low values, the dynamics delays much more in converging. As the last action of our dynamics, we allow an individual to unfollow a given friend and connects to another at random (see Figures). This step is henceforth called rewire. The unfollowing in social networks have been extensively studied. For instance, the study developed by indicates that Twitter users are less likely to unfollow friends who have acknowledged them. With the basis in this study, here, if an individual is repulsed by a neighbor, the rewire can happen according to a given function, Prewire(x), for x = | bj – bi |. Our employed strategies are described in Section 'D'. Another initial node is randomly selected, and all the process is repeated some 'n' iterations, in which n should be big enough to lead the dynamics to reach a steady state.

So, as mentioned above, we divided our dynamics into **6 main parts** which stated as follow:

- Post Transmission
- Post Distribution
- Post attraction
- Post repulsion
- Rewiring
- Spamming

A) **Post Transmission**: Transmission refers to some information or a post carrying some thoughts or opinion is being transmitted or carried to some individual. To measure the value or the chance of this step taking place in real, we assigned a probability function which calculates further the probability of the post being transmitted to each of the individuals.

We considered the users that have a much higher probability of posting information similar to their own opinions and cannot post contrarian information, which can be modelled as:

A(i):

Pt(sim) = cos^2(x π/2),if x≤ 1
Otherwise, pt(sim) = 0

The third tested strategy is the uniform probability, as follows:

A(ii):

Pt(uni) = 1

This probability simulates the cases in which the users produce posts without taking care of the information. More specifically, all the created posts are spread by the users. This case can also be used as a null model.

B) **Post Distribution**: The individuals to whom the post is transmitted distributes to his friend's circle or the connected individuals. we also took into account some possibilities of probability functions in this operation which measures the amount of chance the post would be distributed to each individuals in his circle. The functions are stated as follows:

B(i)

$$\boxed{Pd(x) = \cos^2(x\,\pi/2 + \Phi)}$$

where the parameter 'Φ'is a real number that controls the starting point of the cosine-squared function or we can say it's a scaling factor and $x = | b_i – b_j |$, in which bi and bj are the opinions of the individual 'i' and its given neighbourj, respectively. We considered another version, in which the probabilities vary smoother than in equation which is stated as follows:

B(ii)

$$\boxed{Pd(x) = \cos^2(x/2 * \pi/2 + \Phi)}$$

the relationship between the parameter and the functions of probability. For both cases,
the functions can represent a range of algorithms that spread information from a polarized to depolarized. As the third case, we also employed a null model, that transmits uniformly the information, which is defined as:

B(iii)

$$\boxed{Pd(x) = 1}$$

C) **Reactions by the individuals**:

i) **Post Attraction**: In this phase, it is determined whether the individuals to whom the post is being distributed by the selected main individual are liking the post or getting attracted towards that post because of the similarity between the thought of that individual and that particular post. The assigned probability function to this operation or dynamics is stated as follows:

$$\varepsilon j (\theta, bj) = 1 - | \theta - bj | / 2$$

ii) **Post Repulsion:** In this phase, it is determined whether the individuals to whom the post is being distributed by the selected main individual are disliking the post or getting repulsed away from that post because of the difference between the thought of that individual and that particular post. The assigned probability function to this operation or dynamics is stated as follows:

$$\varepsilon j (\theta, bj) = 1 - (1 - | \theta - bj | / 2)$$

D) **Rewiring**: After the individuals who are connected with the distributor individual got attracted or repulsed, they can change their mind such as avoiding that particular person or starts a new friendship with some other person due to randomness in their behaviour and therefore we are assigning a probability function in this step also to measure the degree of chance of an individual getting rewired.

In order to rewire only connections between individuals that strongly disagree, we adopt the following rewiring probability function The function is stated as follows:

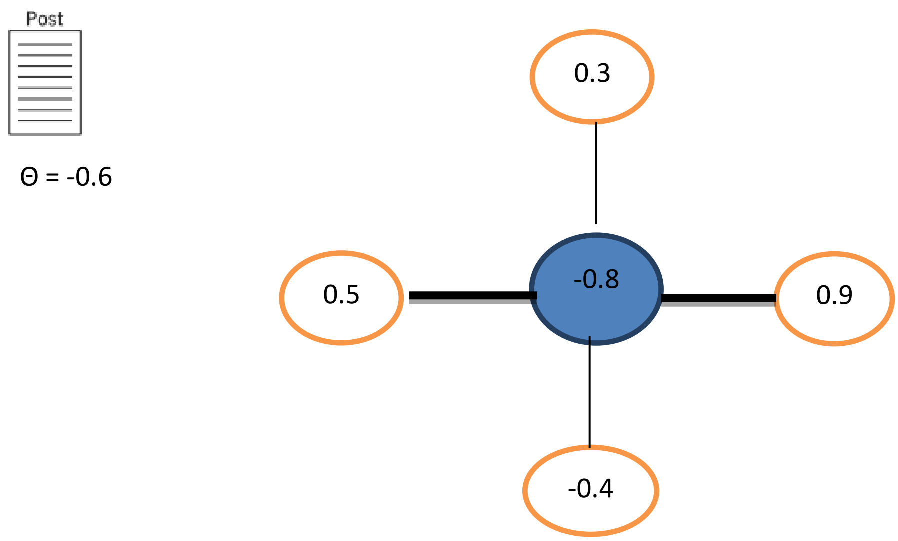
$$Pt(sim) = \cos^2(x\ \pi/2), \text{if } x > 1$$

Otherwise, pt(sim) = 0

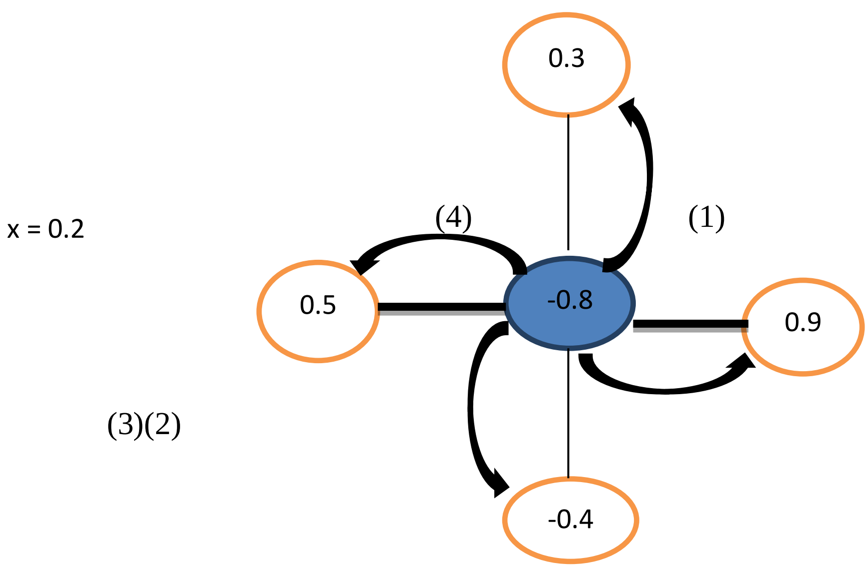
where x is defined by the difference between the opinions of the individuals i and j, x = | bj – bi |.We also considered the dynamics without the possibility of rewiring (Prewire(x) = 0).

E) **Spamming**: In this step, we are fixing some nodes suppose let's say 'n' nodes as spammers who will spam nodes up to level - 2 in respect to a tree at some fixed iteration values such that the rate of spamming remain stable and visible during results .

Figure 1 illustrates one iteration of the proposed dynamics .



1.a



1.b

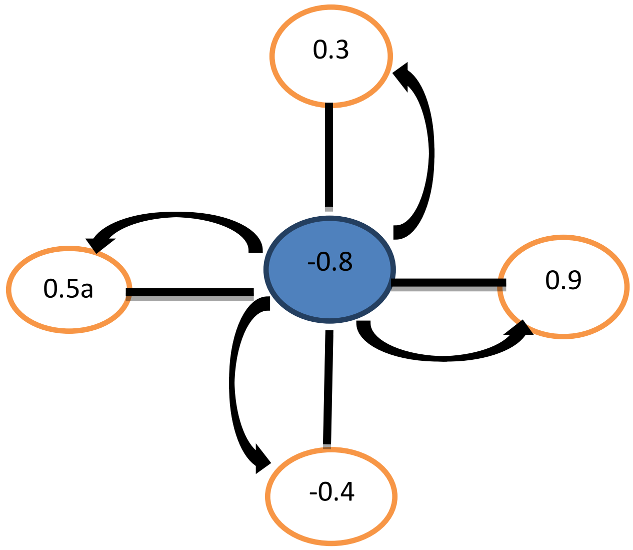
Pd(x1) = 0.42

Pd(x2) = 0.05

Pd(x3) = 0.90

Pd(x4) = 0.27

×



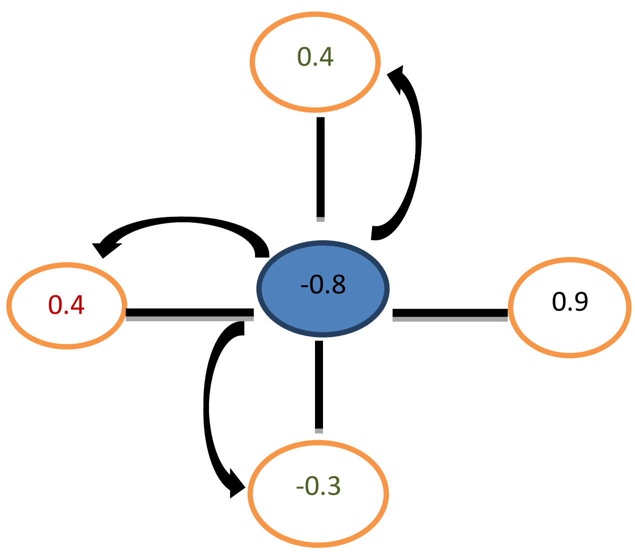
1.c

For attraction ,

X1(|-0.4-0.3|)=0.7

X3(|-0.4-(-0.4)|)=0

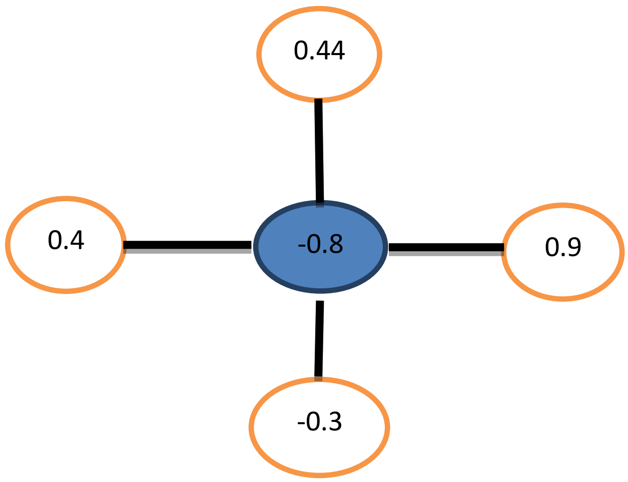
X4(|-0.4-1.0|)= 1.4



1.d

✕

Prew (x=0.9) = 0.1



<u>1.e</u>

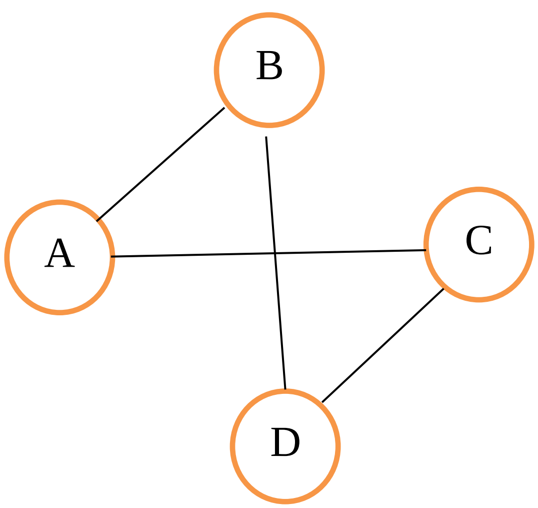
Figure – 1

Implementation

Here we discuss about how we implemented our problem or the analysis. Basically we have implemented our problem in python. For these we have used various python libraries and modules. The followings are the used python modules and libraries:

- NetworkX
- NumPy
- SciPy
- Matplotlib
- Random
- Math
- TextBlob
- Tweepy

Firstly, we collected a data set from a website [1] for analysing. Now we read the data set and have drawn the network or a graph 'G' using NetworkX .NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Using networkx we can understand, store, modify and illustrate complex networks. Then we created an adjacency matrix from the data set using NumPy and NetworkX for detecting the connection of a particular node with its neighbouring nodes. NumPyis a Python library used for working with 'n dimensional arrays '. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. We created the adjacency matrix for indicating the connection or friendship of each node. If the value is 1, it indicates that there is connection or friendship with a specific node and if the value is 0, then it indicates that there is no connection or friendship. For example, we consider a small graph given below:



Adjacency Matrix of the given graph:

$$\text{Mat} = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{pmatrix} A & B & C & D \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Here node A has connection with node B and C that's why the value of Mat(AB) and

Mat(AC) = 1.

Now we iterated through the graph and initialized each node with a random opinion value. We use the Random module for generating a random opinion between -1 to 1. We assigned unique IDvalues to each node of the graph which we considered as individuals and helps to identify each individual's connection list separately from the adjacency matrix using NumPy, ndarray and flatten function. We created a class for keeping all these data and created an object for every node and assigned individuals and opinion value, a unique ID and neighbours connection list for each node. Then we put the objects within a list.

Now, we introduced a new idea to our analysis as mentioned above which is about spamming fake IDs up to level 2(in reference to tree) from a randomly selected spammer node which is very common nowadays on any social media platform to make some particular opinions or ideas dominant all over the platform gradually or we can say just to put some biased opinion on the social media wall.

Mathematically, we fixed some spammer nodes randomly from the initial network in our analysis using Random module which generates the ID of the node who is going to spam and each of those spammers spammed '6' fake ids/nodes. In our analysis, we also added some dynamics taking place including a rewiring dynamics by which a node/individual can disconnect themselves from their neighbouring/friendly nodes. We put the IDs of the spamming node within a list for easy recognition.

Now we again created a new adjacency matrix because the graph got updated now. After this, we assigned the opinion value to the spammed nodes same as the producer node's opinion. Again we separated the connection list of a node with its neighbouring nodes from the adjacency matrix. We updated these in the objects we created previously and also added the spamming nodes to the object list.

Now, we selected one random actual user (not fake ID) from the network and we performed the proposed dynamics including transmission, distribution, attraction, repulsion and rewiring respectively starting from that particular node or individual in real life and calculated the probabilities according to the mentioned functions. We iterated the program for 1000 times and every time a random actual user/node was selected and we performed all the dynamics.

We created some lists for storing the difference between the post's and selected individual's opinion, the difference between the individual's opinion and its neighbour's opinion, the result of the transmission function for the selected nodes, the result of the distribution function for the selected nodes and the result of the rewiring function for the selected nodes for every iteration.

Here, we assumed that the value of Φ to be 0 and the value of Δ as 0.1 for performing the dynamics.

Firstly, we selected a node which is an actual user by generating a random user ID. Then we found its neighbouring nodes by iterating the connection list and stored the ids of the neighbouring nodes into another list of neighbouring node IDs.

Now we generated a value of θ (post's opinion),between -1 and 1 using the random function for calculating the transmission function.After that, we calculated the difference between the post's opinion and selected individual's opinion and then we called the transmission function and passed the value of the difference through the transmission function for calculating the value of transmission.

Now, we checked that if the post was transmitted or not. If the difference between the post's opinion and selected individual's opinion was less or equal to 1, the post was transmitted and we further proceed for distribution .

In case of distribution, we basically calculated the difference between the individual's opinion and its neighbour's opinion by fetching the neighbouring nodes IDs from the list neighbouring node IDs we created previously. After that, we passed the value of that difference and Φ through the distribution function for calculating the value of distribution.

Again, we checked that if the post was distributed or not. If the probability of the distribution was greater or equal to 0.1, Then we assumed that the post was distributed and further proceed for attraction, repulsion and rewiring .

We calculated absolute value of the difference between the post's opinion and the opinion of each neighbour and compared if it is less than 1 or equal 1 which means almost similar opinion and that's why now we updated our bj by bj + Δ and rewired that individual towards friendship by updating its connection value in adjacency matrix by 1.0 , otherwise we assigned bj as bj – Δ because there was a big difference and rewired it away from that node by updating its connection value in adjacency matrix by 0.0 .

The rewiring has been done according to the proposed function and the difference between the neighbours' opinion and the opinion of the individual as mentioned in dynamics description section 'D'.

The above dynamics in the code section is done by choosing the following equations from the dynamics part :

A(i) , B(ii) , c(i) , D(i) .

Furthermore , we calculated the bimodality constant according to the given formula in the dynamics description part using the in-built skewness and Kurtosis functions from Scipy library of Python and we also separated two sets of people comparing with our fixed value 0 and also evaluated the balance factor .

We plotted the density map using Kernel Density Estimation logic and used meshgrid for this .

At last , we again have drawn the graph G using networkx spring layout to have a view of the final network after spamming as well as rewiring .

Implementation of density map is performed in the following steps : -

we take two lists as input.

Step 1 :-

(Grid Size and Radius)

While creating heatmap using KDE we need to specify the bandwidth or radius of the kernel shape and output grid size. For this case, I use radius 10 m and grid size 1 m.

Step 2 :-

(Getting X,Y Min/Max to Construct Grid**)**

To construct grid we use mesh grid. Therefore we need to find x,y minimum and maximum to generate a sequence number of x and y. These sequence numbers then will be used to construct mesh grid. To include all the dataset coverage with a little bit more space, I subtract x,y minimum with radius and add it up for x,y maximum

Step 3 :-

(**Calculate Grid Center Point**)

After constructing mesh grid. Next we calculate the center point for each grid. This can be done with adding x mesh and y mesh coordinate with half of grid size. The center point will be used later to calculate the distance of each grid to dataset points

Step 4 :-

(**Kernel Density Estimation Function**)

To calculate a point density or intensity we use a function called kde_quartic. We are using Quartic kernel shape, that's why it has "quartic" term in the function name. This function has two arguments: point distance(d) and kernel radius (h).

<u>Step 5</u> :-

(**Compute Density Value for Each Grid**)

This is the hardest part of this post. Computing the density value for each grid. We are doing this in three looping. First loop is for mesh data list or grid. Second loop for each center point of those grids and third loop to calculate the distance of the center point to each dataset point. Using the distance, then we compute the density value of each grid with kde_quartic function which already defined before. It will return a density value for each distance to a data point. Here we only consider the point with a distance within the kernel radius. We do not consider the point outside the kernel radius and set the density value to 0. Then we sum up all density value for a grid to get the total density value for the respective grid The total density value then is stored in a list which is called intensity_list.

<u>Step 6</u> :-

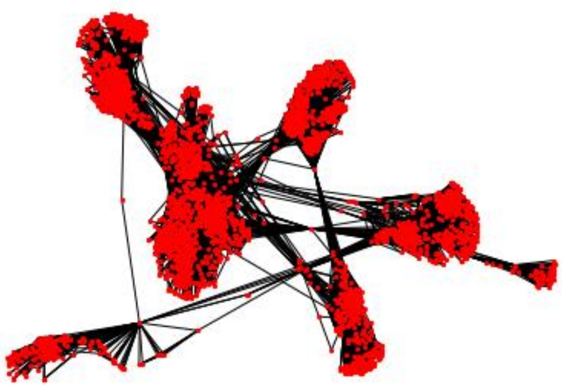
(**Visualizing The Result**)

The last part we visualize the result using matplotlib color mesh. We also add a color bar to see the intensity value .

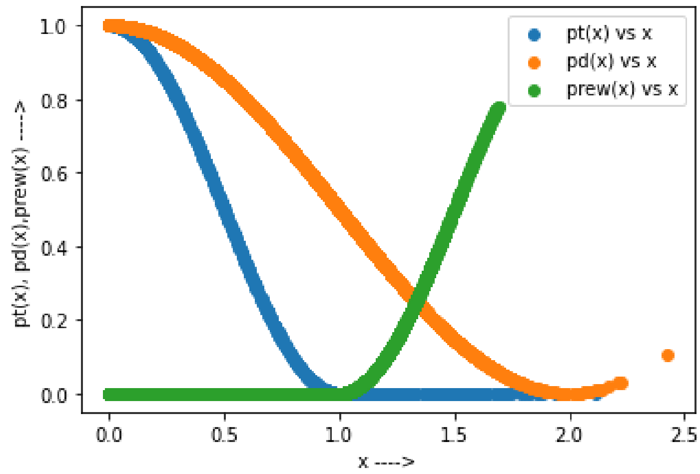
Datasets and Results

Next , we present our results, starting from analysis of the opinion distributions, and varying the post transmission parameters, reception, and rewiring. We also analyse a real scenario in terms of the proposed methodology .

Firstly, we have read the dataset and have drawn the original network using the spring layout of NetworkX. The diagram we got is shown below:

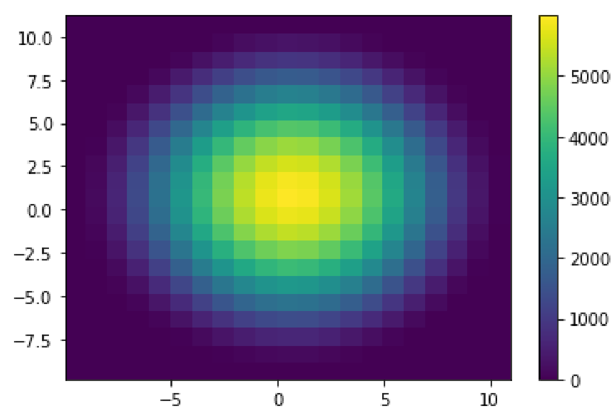


Now , we applied our dynamics of spamming per 100 th iteration and we selected $1/100^{th}$ spammer individuals or nodes of total nodes in network and produced fake nodes upto level – 2 as per the proposed model . After we got the network containing the original as well as fake nodes , we started our steps of transmission , distribution , reaction and rewiring according to the dynamics and generated the probability values by choosing the suitable functions as mentioned in implementation section . After completion of the 1000th iteration we plotted the graphs for the probability functions for the main 3 steps versus the difference values of the opinions calculated for evaluation of the functions . The plots are given below :

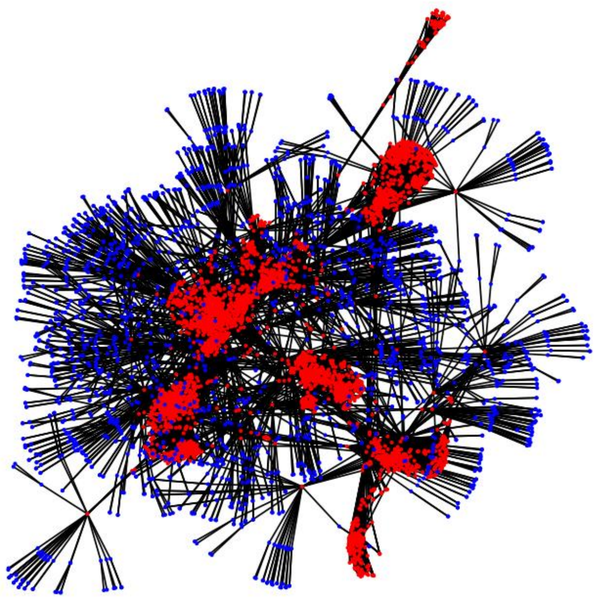


Now at the end of the dynamics , we calculated the bimodality constant and as per our dataset we are getting somewhat around 0.5 which is a standard value and we set a value 0 and divided the nodes on the basis of their opinion value greater and lesser than this value . We are getting groups divided in the ration of approximately 2:3 every time we run our program . we also calculated a balance factor which we got in the interval [0.5 , 1.0] .

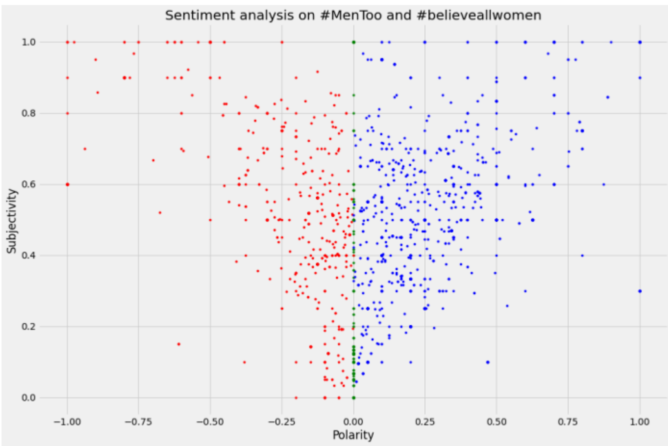
We also drew the density plot at the using Kernel Density Estimation technique as mentioned in the implementation part to see if the opinions of all the individuals at this stage are almost similar to each other or there are large differences among them . The density plot we got indicates its consensus in nature . The plot is given below :

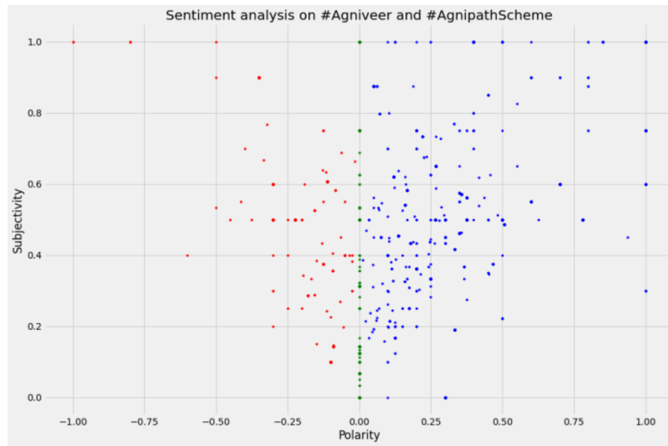


At the end , after rewiring and spamming we wanted to visualize the final structure that’s why we have redrawn the network which is shown below :



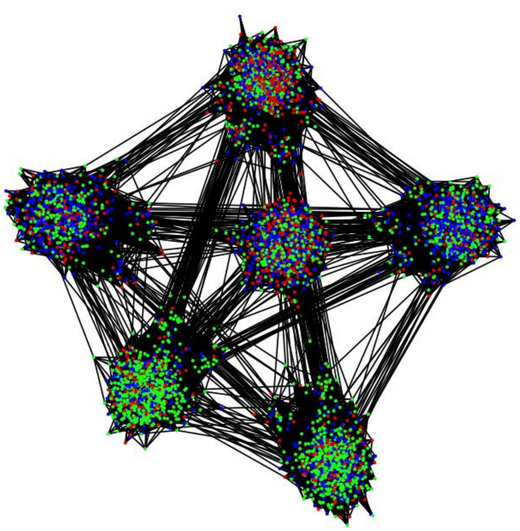
The above results are seen for two datasets collected by performing sentiment analysis using textblob from the tweets which we fetched using twitter API on the basis of some popular hashtags that went viral during some specified time . The sentimen analysis is given below :



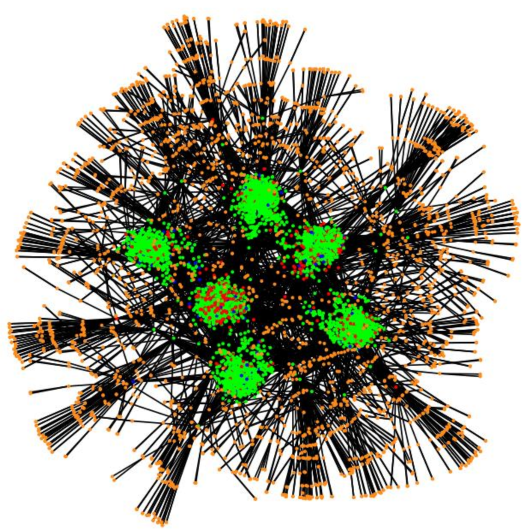


Results (#MenToo, #believeallwomen, #AmberisnotAlone)

Initial Network

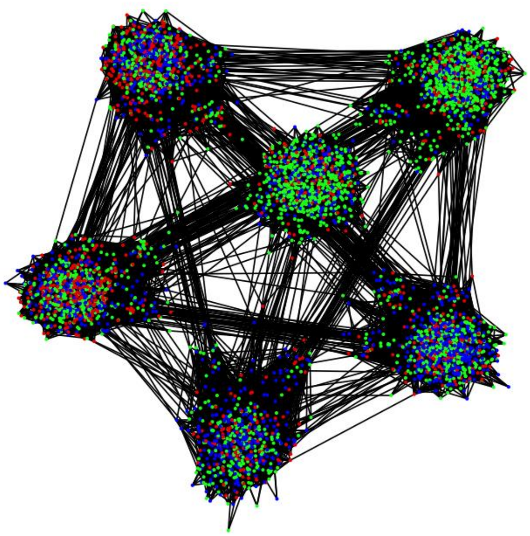


Final Network

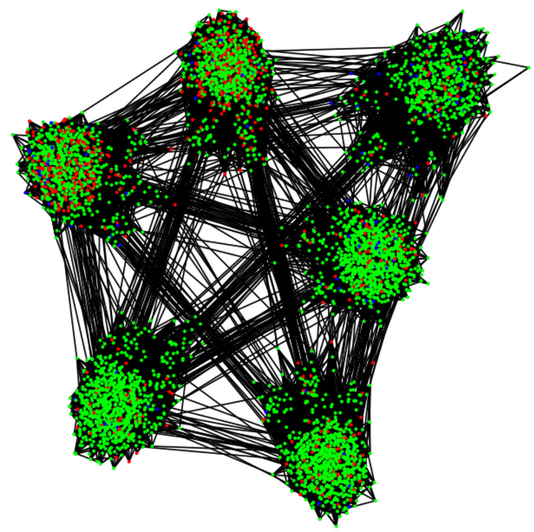


Before and After rewiring without spamming

Initial Network

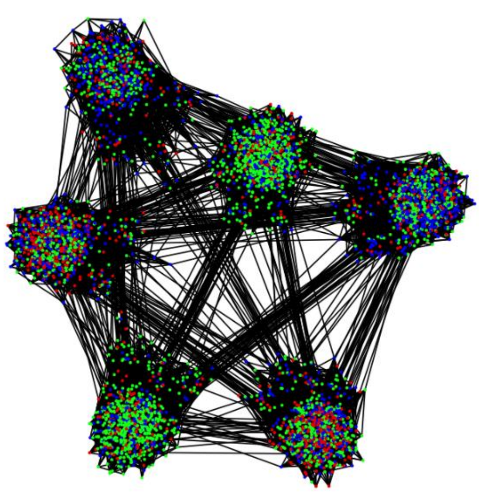


Final Network

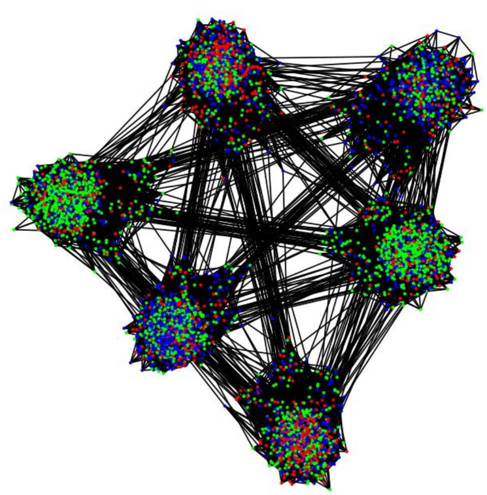


Before and after without rewiring and without spamming

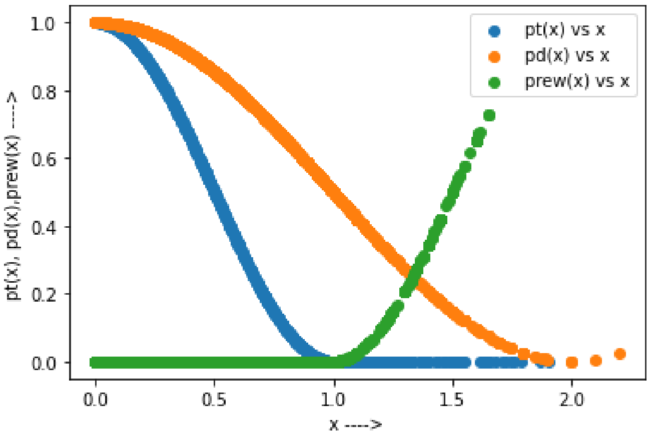
Initial Network



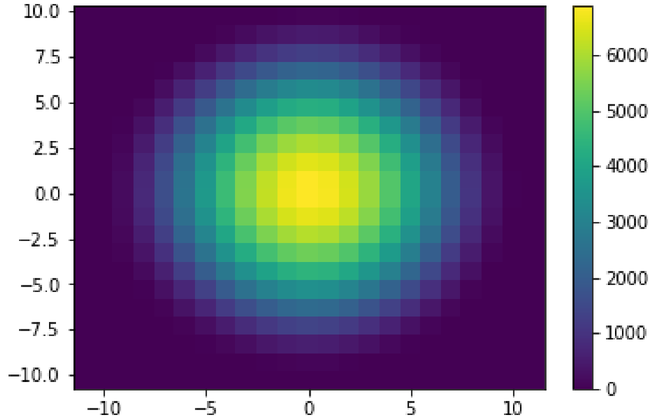
Final Network



Probability plots

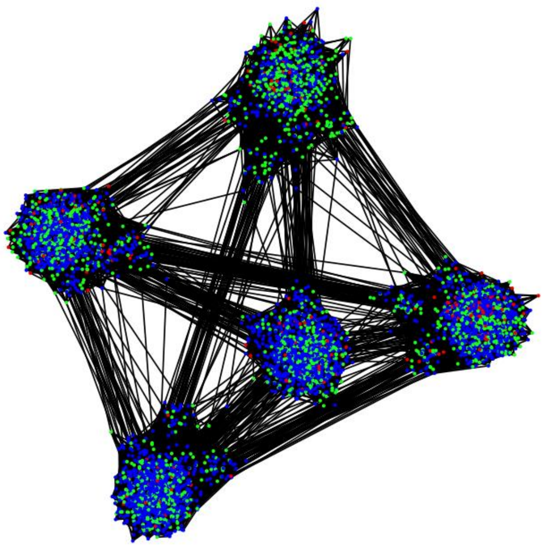


Density map

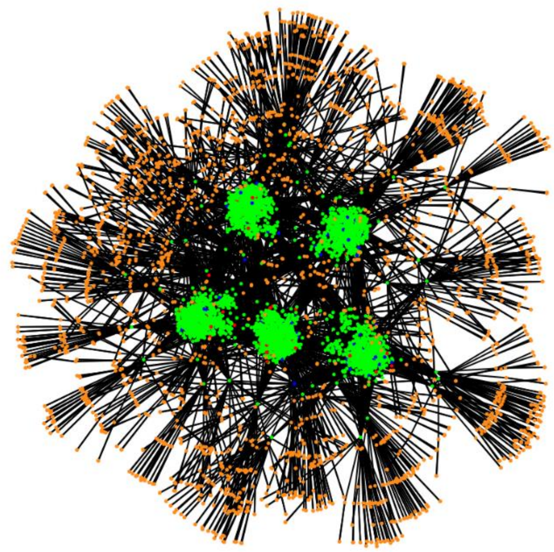


Results (#Agniveer, #AgnipathScheme)

Initial Network

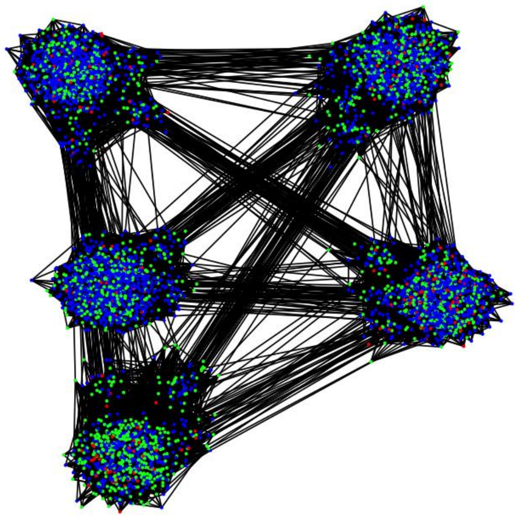


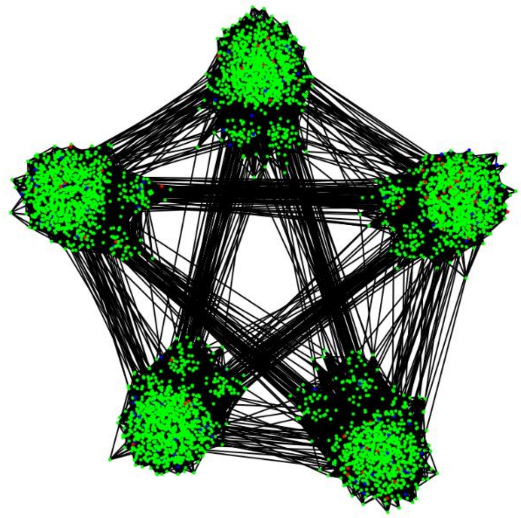
Final Network



Before and After rewiring without spamming

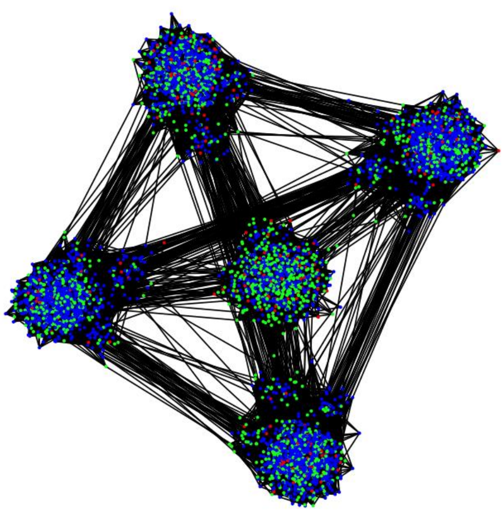
Initial Network | **Final Network**

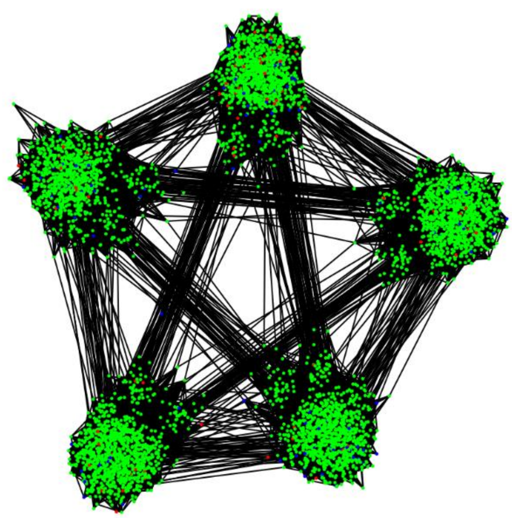




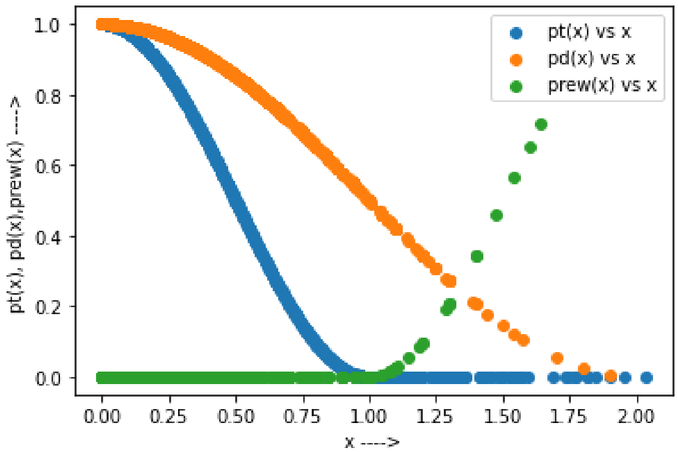
Before and after without rewiring and without spamming

Initial Network | **Final Network**

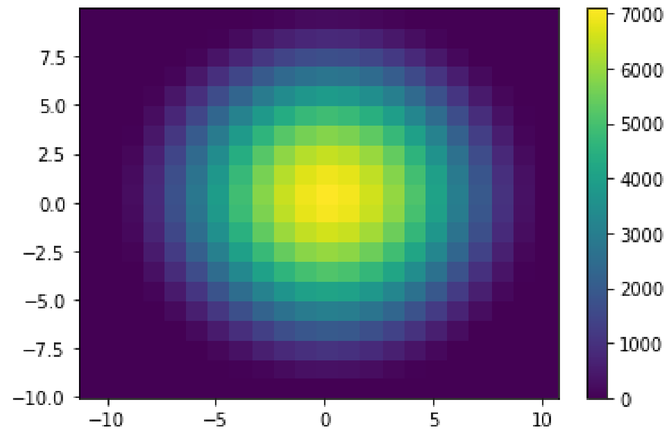




Probability plots



Density map



Discussion

- In this section , we can say that for our dynamics we chose squared trigonometric cos function , instead we could have chosen absolute cos function and we could have used sin function as a result of which we would have got phase changed plots .

- We assumed the value of the scaling factor ‘Φ’ in the distribution part to be ‘0’ which can be changed to see further change and we considered the small change of opinion value ‘**Δ**’due to attraction or repulsion very less which is 0.1 . It can be increased to see if more rewiring. takes place .

- we would like to discuss about the random number generations in which we used random module of python but we took the values from Gaussian Distributions . For further change the random values can also be taken from Poisson Distribution which will change the results as it will generate not only positive but also discrete values .

- In the spamming part , we selected a fixed no of spammers and level of spamming according to our choice , further it can be controlled and changed by calculations in the code section .

- After spamming , we saw the final network structure which was pretty much similar with protein protein interaction networks which took us to a interesting conclusion but it has to be further studied why this happened if there is similarity behind this analysis and the theory or dynamics behind the PPIN .

- We modelled the dynamics initially and checked the results and latern on, added the feature of rewiring to make it more interesting .

- Lastly , we considered 1000 iterations to perform this dynamics . It can be decreased or increased according to choice so that any change in the results may take place .

Conclusion

As nowadays the use of social networks has increased , researchers have been studying their opinion dynamics . In this analysis , we proposed a model to study the conditions that can give rise to different behaviours . Our dynamics is based on some divided sections or and it is limited to simulating how new facts are generated as well as the individual's friends' reactions . Importantly, this study does not only contribute to a static model but also proposes a new type of analysis. Here, we considered that the network structure is not the single aspect that limits the communication between individuals .

An interesting outcome was observed in our case .Though , we modelled our implementation in such a way that we could see the outcomes of this dynamics without rewiring but as we added a major operation of spamming in this dynamics and we chose the probability functions considering that the individuals are aware of what he/she is posting in this social media wall , even after rewiring of some individuals , we got consensus as the result when we mapped the density of individual's opinion vs the average of opinions of its neighbours .

At the end of the dynamics , as the calculated bimodality constant value according to our dataset and analysis is something around 0.5 and the balance factor is something between 0.5 and 1 , the groups we got were almost divided in the ratio of 2:3 .

Further analysis can be done by randomizing the spamming dynamics instead of doing it for fixed iteration values and introducing a dynamics of rejecting a particular post or deactivation of accounts .

Last but not the least , one of the most interesting observation we must mention is that after spamming the network took a shape just like a protein protein interaction network which can be further understood and analysed in future work , all we can say that may be some behind mathematics or some features which are similar .

We would like to conclude that it was interesting to work technically on this topic of social networking analysis which has a real life meaning and understanding the human opinions and thoughts around a random post or information in various way .

References

[1] *Modeling how social network algorithms can influence opinion polarization* - Henrique Ferraz de Arruda, Felipe Maciel Cardoso , Guilherme Ferraz de Arruda , Alexis R. Hern´andez , Luciano da Fontoura Costa, and Yamir Moreno

[2]J. Fern´andez-Gracia, K. Suchecki, J. J. Ramasco, M. San Miguel, and V. M. Egu´ıluz, Phys. Rev. Lett.112, 158701 (2014).

[3] K. Sznajd-Weron and J. Sznajd, International Journal of Modern Physics C 11, 1157 (2000).

[4] S. Moscovici and M. Zavalloni, Journal of Personality and Social Psychology 12, 125 (1969).

[5]S.Kokoska and D. Zwillinger, CRC standard probability and statistics tables and formulae (Crc Press, 2000).

- **Websites follows** :

  https://numpy.org/

  https://networkx.org/

  https://matplotlib.org/