Università degli Studi di Torino

Scuola di Scienze
Dipartimento di Fisica
Corso di Laurea in Fisica dei Sistemi Complessi

Opinion dynamics under bounded confidence on static and adaptive network

Lorenzo Betti

27^{th} **June 2018**

Contents

1 Introduction

In recent years we have witnessed the growing interest of physicists and computer scientists in the field of collective social phenomena such as opinion dynamics, in which a group of individuals has opinions on some topics and individuals can change idea through interaction in order to investigate how these opinions evolve in time.

The study of such systems borrows methods from statistical physics, trying to capture the essential features of complex social behaviour. To take advantage of the strong similarity with some physical system such as spin systems (for example the Ising model), consisting in a regular lattice of atoms in one of two spin states (up and down). By interaction, atoms can flip from spin up to spin down and vice versa and it is possible to observe situations in which all atoms in some particular region of the lattice, called domain, are aligned with the same spin. This spin system inspired the voter model that is one of the first and simplest model of opinion dynamics. It consists in a lattice where in each node there is an individual who can interact with a neighbor chosen at random taking its opinion with a certain probabily, with opinions expressed by a binary value meaning agreement or disagreement about some particular topic. By simulation one can observe the formation of groups sharing the same opinion as an emergent behaviour.

Other models have been developed during time all taking account of a fundamental result caming from social science, that people in a group exhibit hearding behaviour following local majority or following their neighbors.

In order to differentiate various kind of models, we will refear to discret and continuous opinion: the former means that opinions each individual can take is defined by a set of discrete values, while the latter means that opinions are values that span in a continuous interval. In taking account of more opinion an individual can have, vector of opinions can be used in which each component of the vector represents the individual's opinion about one topic. The vector containing opinions of all individuals is called opinion profile and it has as many component as the number of individuals in the system. The range of value associated with opinion depends on the model but in general the interval $[0, 1]$ is used.

There are also different ways of modelling individuals' interaction. In simple models two individuals are choosen randomly and they exchange opinions with a certain rule depending on the values associated with their opinion, otherwise other models consider individuals in a social network where they can interact only with neighbors. In the latter case it is possible to take account of the structure and the dynamics of the network differentiating between static and adaptive networks: static does not take account of the evolution of the network while adaptive considers the evolution of the network in a timescale comparable with timescale of opinion evolution making the influence that one process makes on the other interesting.

Such models permit to study emergent social behaviour as the reaching of consensus, when all individuals share the same opinion, polarization, when in equilibrium only few opinions survive, and fragmentation, when the number of opinions in the equilibrium state scales with the number of individuals.

In this work I will focus on continuous opinion dynamics under bounded confidence in static and adaptive network. I will consider a population with n individuals each one with his opinion at time t. The rule to update opinions consists in choosing randomly an individual i and choose randomly one of its neighbor j in the network, and let them interact only if their opinions are similar, that is if the difference of opinions is bounded. I will generate both random and scale free network giving them the possibility to evolve with certain rules or not and finally comparing results between static and adaptive models.

I am interested in the formation of comunities, defined as a subset of agents that share similar opinion. I will return on the definition of comunity later. I will study the case of adaptive scale free network with more attention with the aim of show when we can reach consensus varying the bound of the difference of opinions. In particular I will show how the number of comunities and the number of agents in the biggest comunity change varying the value of the bound.

2 Model

The model is composed by n agents in a social network, each one corresponding to a node, and social connection between two agents is represented with a link. Every agent has its own opinion x expressed by a real number $x_i(t) \in [0,1]$ meaning that agent i, with $i \in [1,n]$, has opinion x at time t. As initial condition, in $t=0$ each agent has a random opinion. The interaction rule is based on the Deffuand model [1] extended on a network. Agents i and j can interact only if there is al link between them and the update rule is:

$$x_i\,(t+1) = \begin{cases} x_i\,(t) + \mu\,(x_j\,(t) - x_i\,(t)) & \text{if } |x_i\,(t) - x_j\,(t)| < \epsilon_i \\ x_i\,(t) & \text{otherwise} \end{cases}$$

$$x_j\,(t+1) = \begin{cases} x_j\,(t) - \mu\,(x_j\,(t) - x_i\,(t)) & \text{if } |x_i\,(t) - x_j\,(t)| < \epsilon_j \\ x_j\,(t) & \text{otherwise} \end{cases} \tag{1}$$

with $\epsilon \in R$ and $\mu \in [0, 1/2]$. This means that two agents can change their opinion if the difference of initial opinions is bounded by ϵ and their final opinions will be closer than before. The opinion change is regulated by a parameter μ in a way that the final opinion is included between the initial opinion ($\mu = 0$) and the intermediate opinion of the two angents ($\mu = 1/2$). For simplicity, I will consider the case $\mu = 1/2$, with both agents reaching the same opinion. On the other hand, the parameter ϵ has the role of tolerance and its value is central for the consensus reaching mechanism. I will use a constant value of ϵ for all agents in the interval $(0, 0.4]$ because for higher values the reaching of consensus starting from random opinions in $[0, 1]$ is trivial. This simplification excludes some kind of phenomena, such as the growt of extremism, meaning a group of agents with opinions near 0 or 1. Indeed agents with an extreme opinion are attracted towards more moderate opinion because of the limited opinion range. For exemple, if an agent with opinion 1 can interact, the value of its opinion can only decrease. For a complete analysis of the effect of different values of tollerance choosen from a certain distribution see [3].

In classical models of opinion dynamics, an agent can interact with all other agents with a certain probability [1] [2] . This assumpion is called *homogeneous mixing*. To take account that agents can't interact with every member of a population, other autors consider agents in a social network and this permits to limit the number of neighbors. This seems a more realistic assumption [3][4].

The model under study consists in a random or scale free network in the beginning. In particular I will consider two kinds of network, depending on the capability of agents to break one of its link and rewire that in order to link with another agent. This capability is ruled by a certain probability w. At each step, an agent i of the population and one of its neighbors j are randomly selected. With probability w, they break the link if their difference of opinion is not bounded by ϵ and i makes a link with another randomly selected node from agents that are

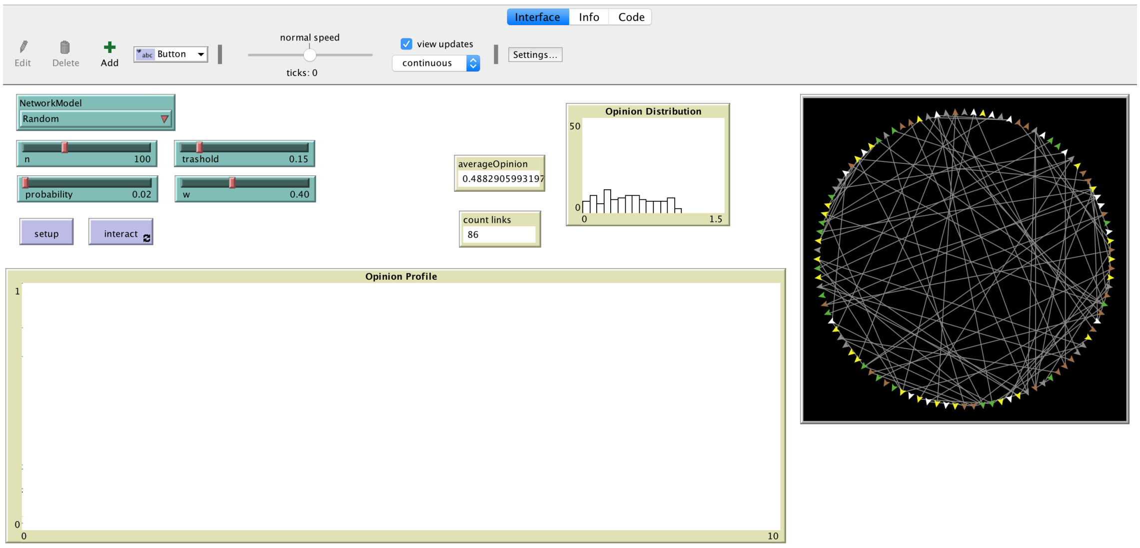


Figure 1: NetLogo interface of the simulation.

not in its neighborhood. If their difference of opinions is bounded by ϵ, the link remain. Otherwise, with probability $1-w$ they can comunicate and change opinion with the update rule (1). The parameter w permits to distinguish two cases: if $w = 0$ the network is static, meaning there is no rewiring and no breaking of links, while if $w > 0$ the network is adaptive and the topology of the network can evolve permitting to agents to find someone with opinion more similar.

In this model there are two dynamical processes regulated by the probability w. This parameter regulates the rate at which those two processes occur.

3 Simulation

I implemented the simulation of the model in NetLogo and in Python. The former is usefull because of its graphic interface with which we can make qualitative consideration, while the latter permits to make calculations and to extract statistically significant properties. I will explain the structure of the two codes in the next sections.

3.1 NetLogo

NetLogo is an agent programming lenguage and a integrated modelling environment with a graphic interface that permits to see an animation of what is happening while code is running.

The interface is shown in picture 1 and is composed as follows:

- a chooser with name *Network model* that permits to start with a random or a scale free network
- a slider with name n that sets the number of agents
- a slider with name *probability* that works only when a random network model is selected. This represent the probability to create a link from all possible links
- a slider with name *threshold* that is the value of ϵ in (1)
- a slider with name w that is the value of the relative rate of interaction or rewire dynamics
- a button with name *setup* that create the network and gives opinions to each agent
- a button with name *interact* that start the simulation
- a monitor with name *average opinion* that shows the mean value of the opinion of all agents
- a monitor with name *count links* that shows the number of links of the network
- an histogram with name *Opinion Distribution* that shows the dispribution of opinion between agents
- a plot with name *Opinion Profile* that shows the evolution of opinion of some representative agents choosen from the population in function of time
- the world on the right where we see the simulation. Arrows represent agents while lines represent links between agents.

First of all, we set the parameters in sliders and after we can push a button with which the part of code associated with the button strarts running. The code is written in another part of the environment called *code*.

With the *setup* button we create n agents and links following some rules. An integer number identifies each agent in a way that agent's ID goes from 0 to $n-1$. When we select random network, we select one turtle creating a link with another agent with probability p and we do that for all agents. We ask agents to create links only with turtles with greater ID number to forbid double evaluation of a link ($self > myself$)

```
to setup

  ca
  reset-ticks

  ;;;Create random network;;;

  if NetworkModel = "Random" [
  create-turtles n [set opinion random-float 1.0]
  layout-circle turtles (max-pxcor - 1 )
  ask turtles [create-links-with turtles with [self > myself and
                                              random-float 1.0 < probability] ]
                           ]

  ;;;Create network with preferencial attachment mechanism;;;

  if  NetworkModel = "Scale free" [
  create-turtles 2 [set opinion random-float 1.0]
  layout-circle turtles (max-pxcor - 1 )
  ask turtle 0 [create-link-with turtle 1]

  loop [
  if count turtles >= n [stop]
  let partner one-of [both-ends] of one-of links
  create-turtles 1 [set opinion random-float 1.0 create-link-with partner]
  layout-circle turtles (max-pxcor - 1 )

       ]

                                                ]


  colorate
  graph

end
```

Figure 2: Code of *setup* button.

. The expected number of links is $pn\,(n-1)\,/2$. When we select scale free network, we start creating two agents and a link between them. At each step we add an agent and it will make a link choosing randomly a link of the network and linking with one of the two agents linked with the choosen link. We stop when we have n agents. This mechanism is analogous with the preferential attachment and we finally have a scale free network [6]. With this algorithm we add just one link at each step and that is restrictive because we have no cycles in the network. Finally, we give an opinion as a random number in $[0,1]$ to all agents. In the last lines there are two functions, *colorate* and *graph*. The former gives a color depending on opinion and the latter makes the histogram that we can see in the *Opinion profile* plot. The *setup* code is shown in figure2.

The *interact* button is the core of the simulation. It starts choosing an agent and one of its neighbors. If at least a neighbor exists, with

```
to interact

;;; Choose randomly an agent (i)

  let i one-of turtles

;;; Choose randomly an agent's link (j) and define an agentset with agents that are not i's neighbors (notNeig)

  ask i [let j one-of my-out-links  ;j diventa un link che appartiene al nodo i
         let r random-float 1.0
         let notNeig turtles with [not link-neighbor? myself and self != myself]

;;; If a link exists, let agents sharing link j interact with probability 1-w and let i try to rewire the link with probability w

          if j != nobody [ifelse r > w [ask j [if abs([opinion] of end1 - [opinion] of end2) < trashold [     ;;first if: nodes interact if possible

                                              let tmp ([opinion] of end1 + [opinion] of end2) / 2
                                              ask end1 [set opinion tmp]   ;;; they can interact changing opinion
                                              ask end2 [set opinion tmp]                             ]
                                             ]

                                    ] [ask j [if abs([opinion] of end1 - [opinion] of end2) > trashold [ask i [create-link-with one-of notNeig] die ] ]

          ]
                          ]
  ]


  layout
  colorate
  graph
  tick

end
```

Figure 3: Code of *interact* button.

probability $1 - w$ they interact following the update rule (1) and with probability w the first agent tries to rewire. It will link with an agent that is not in its neighborhood (an agent in *notNeigh* agentset). In the last lines there are the same funcions, *colorate* and *graph*, and another function called *layout* that change the positions of all agents in the worls in order to permit a better visualization of the network. The *interact* code is shown in figure 3.

NetLogo is a very usefull environment because of its graphic interface with which we can see directly the evolution of the simulation to have a rough idea of the role of parameter in the dynamics of the sistem. In spite of this, it has a limited computational power and data manipulation and analysis is not easy.

3.2 Python

Python is a programming lenguage and I developed the code in the web application Jupyter Notebook that allows to create documents containing code. The structure of the code is perfectly analogous with the one in NetLogo previously described except for some built in function in the NetworkX library. In this case I will make data analysis searching for how comunity forms in dependence of the tollerance parameter, focusing on scale free adaptive network.

First of all, I define with more rigour the notion of comunity. Two agents i and j belongs to the same comunity if there is a path in the network where each consecutive agent in the path has an opinion within the tolerance value of the previous agent. This means that there is a channel of comunication between agents belonging to the same comunity

```python
# Here is the code to detect comunities. At the end of the simulation, i search links connecting two nodes with
# distant opinions and i delete that. This ensures me to count the real number of comunities.
# I will do this operation with a copy of the original network.
# After that, i remove all nodes with no links

    H = G.copy()

    for i in list(H.edges()):

        # if the opinion difference is not bounded, the link is broken
        if abs(H.node[i[0]]['Opinion'] - H.node[i[1]]['Opinion']) > threshold :

            H.remove_edge(i[0], i[1])

        # nodes with no links are removed
        H.remove_nodes_from(list(nx.isolates(H)))

# I define the connectend_comunity as the list in which the j-th element contains nodes of the j-th largest component
    comunities = list(sorted(nx.connected_components(H)))

    return G, opinionProfileS, comunities
```

Figure 4: Comunity detection code in Python.

[3].

At the end of the single run of the simulation, I make a copy of the final network searching for links between agents whose difference of opinion is not bounded and I break those links. After that, all nodes without neighbors are deleted. Finally, the function *connected_ components* makes a list with agents belonging to connected components of the network and this is one of the output of the simulation. This part of the code is shown in figure 4.

There are other algoritms that searchs comunity in a network but I choose this one because of its simplicity and because comunty will be perfectly disconnected for values of the probability of rewiring w small enough and for long enough simulation. Agents with no neighbors are not interesting for my purpose because they would produce noise in detecting comunity when consensus is reached. Those kind of agents are not present because of an anti-social behaviour, but because an agent break a link with a neighbor with only one link and in a scale free network there are a lot of agents with only one link.

4 Results and discussion

In this section I will report results, in particular I will report separately four cases: static and adaptive random network (qualitative description) and static and adaptive scale free network (qualitative description of the former and more rigourous description of the latter).

4.1 Static random network

In this case I set $n = 150$, $w = 0$, meaning that agents can not rewire links, and $p = 0.02$, meaning that the expected number of links in the

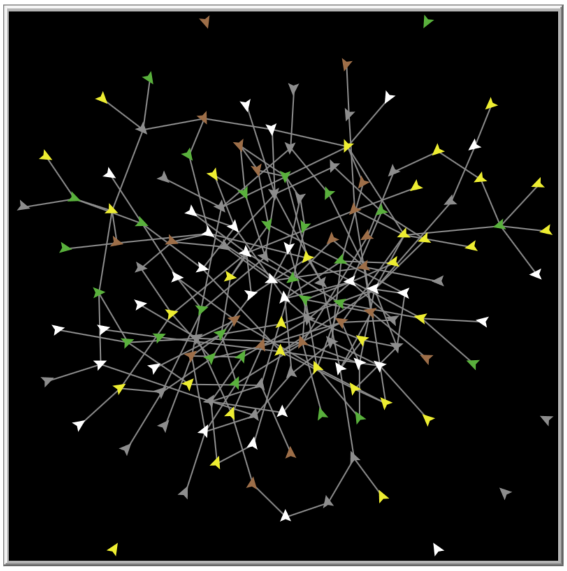


Figure 5: A realization of a random network with 150 agents and with $p = 0.02$ in NetLogo.

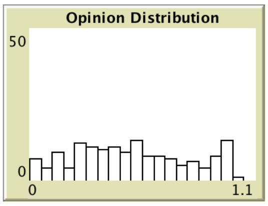
network is about 250 and the mean degree of a node is about 3 . Figure 5 shows a possible configuration of a random network of this kind where agents with different colors correspond to agents with different opinion. In static network, an agent can not change opinion if their neighbors have distant opinion and it breaks a possible channel of comunication. In random network, the probability that an agent has no links is greather than zero, indeed in the immage we see isolated agents. Figure 6a shows an example of an initial opinion profile.

As we change the value of the threshold we can observe different behaviours. For values of ϵ small enought ($\epsilon < 0.10$) we have a final opinion profile not different from the initial, as in figure 6b for example, characterized by an homogeneous distribution because of the small tollerance and the small number of neighbors. This is a fragmentation state. For higher tollerance ($0.10 < \epsilon < 0.20$) two or more comunities with a not negliable fraction of agents emerge as a consequence of the higher possibility to interact, reaching a state of polarization. Finally, for one more time higher tollerance ($\epsilon > 0.20$) a macroscopic comunity emerges (in figure 6d it has 60% of agents) coexisting with smaller comunities.

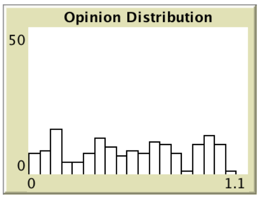
In static random network we will not have complete consensus because of the presence of isolated agents who can not comunicate with others. Despite this, agents in the connected component can reach consensus.

Different values of p cause variation in number of links in the network and in the degree of each node. The effect of an increase of p consists in an higher possibility of interaction in spite of a lower value of the tollerance because each agent has an higher number of neighbors, so its probability to find someone with similar opinion is higher. For example, with $p = 0.10$, meaning a mean degree of each node of 14, and $\epsilon = 0.08$ a finite number of comunities emerges as we can see in figure 6e. Another effect is the faster reaching of a stationary state because of the more channel of comunication. In varying p we have to pay attention because we have no giant component if the mean degree of a node is smaller than 1 [6].

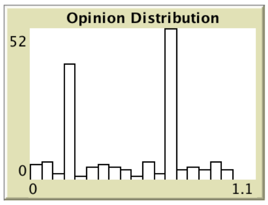
In static network, two comunities sharing the same opinion could not be linked because they coud emerge in different region of the network, so they contribute separatedly in the count of the number of comunities as defined in the previous section. This means that we may have *false* polarization states. For example in figure 6c we see two dominant opinions but we do not know if those correspond to two or more comunities.



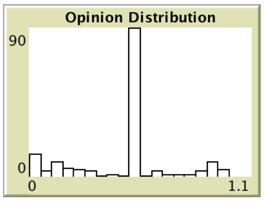
(a)



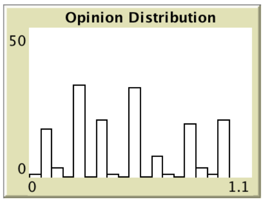
(b)



(c)



(d)



(e)

Figure 6: Opinion profiles of a static random network in different conditions: 6a is an example of an initial opinion profile, 6b is a final opinion profile with $\epsilon = 0.10$, 6c is a final opinion profile with $\epsilon = 0.18$, 6d is a final opinion profile with $\epsilon = 0.22$ and 6e is an example of a final opinion profile of a random network with $p = 0.10$ and $\epsilon = 0.08$.

4.2 Adaptive random network

In this case I use the same setting of the previous case, except the probability of rewiring for which I use $w = 0.4$, meaning that each agent can try to rewire one of its links with probability of 40% when its neighbor has distant opinion. There is no difference in the starting network with the previous case.

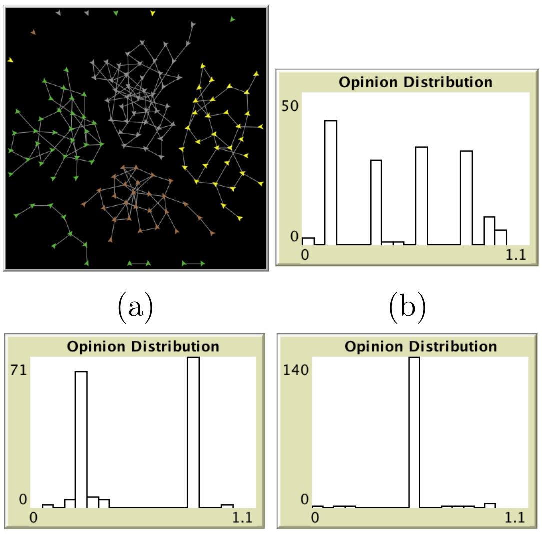
The sistem behaves in a different way because of the rewiring property, especially when the tollerance is low. Indeed, an angent tends to find someone with similar opinion favouring the formation of comunities with no links between different comunities. The network at the end of the simulation is shown in figure 7a and its relative opinion profile in figure 7b in which I used $\epsilon = 0.10$. We can see four main comunities, some isolated agents and some smaller comunities and a different opinion profile respect to the analogous case with a static random network shown in figure 6b. In particular four dominant opinions emerge and the opinion profile looks more similar with the one in figure 6e where each agent has a greater number of neighbors. The difference between static and adaptive network in this case could be attributed to the higher possibility to comunicate, because of a greater number of neighbors or because of the possibility to link with other agents, that permits the growt of comunities despite the small value of the tollerance.

As the tollerance increases, the system approach consensus while the rewiring property tend to form comunities that prevent a complete consensus in an unique comunity. As a consequence, we need a greater tollerance to reach consensus with an adaptive network as compared to a static network. In [3] there is a complete analysis of static and adaptive random networks.

4.3 Static scale free network

Scale free networks have different properties respect to random networks as we can see in the comparison between figure 5 and 8a . First of all, in the former all agents have approximately the same number of neighbors while in the latter there are few agents with a lot of links, called hub (about two in my simulation with around 20 neighbors), and a lot of agents with few neighbors. The degree distribution of the nodes is a power law while random networks have Poisson distribution [6]. This means that the probability to select at random one of the links of the hub is high, so the work of the hubs is crucial for the dynamics of an opinion. This role is magnified by the tree structure of the network so a branch of agents with distant opinion with respect to the hub is excluded from the comunication with other parts of the network. As explained in section 3.1, we construct a network with no cycle.

The sistem reachs consensus for an higher value of the tollerance respect to the static random network because of the previous consider-



(c) (d)

Figure 7: 7a is a network structure of a single run at the end of the simulation for an adaptive random network with $\epsilon = 0.10$ and $w = 0.40$ with its relative opinion profile in 7b. Other figures represents final opinion profiles, 7c with $\epsilon = 0.20$, and 7d with $\epsilon = 0.30$, all with $w = 0.40$.

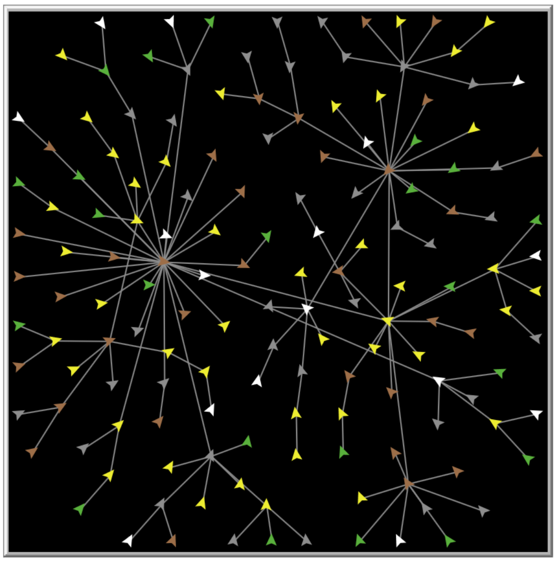
ations. In figure 8b there is an example of the final opinion profile with a static scale free network with an high value of the tollerance for which a static random network reach consensus. We can see that only about an half of agents is divided in two dominant opinions.

This kind of social network is not realistic because no one knows "my friend's friend" and we can expect a different behaviour with an algorithm that takes account of different numbers of link added at each step.

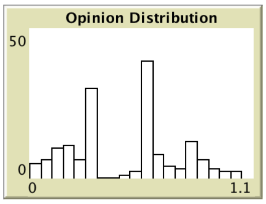
4.4 Adaptive scale free network

Let us now consider an adaptive scale free network. In this case I made a number of simultions that permit to obtain statistically significant results spanning different values of the tollerance. In particular the behaviour of the system with 1000 agents is simulated for one hundred values of the tollerance in the interval $[0.05, 0.4]$, one hundred times for every value. Each run consists in 1.5×10^5 cycles of interaction. I calculated the mean number of agents belonging to the largest and second largest comunity (figure9a) with relative standard deviation and the number of comunities in function of the tollerance (figure 9b).

As a result we can distinguish three different region refering to figure



(a)



(b)

Figure 8: 8a represents a realization of a scale free network. 8b is an opinion profile of a single run simulation of a static scale free network with $\epsilon = 0.30$.

9. For $\epsilon < 0.20$ the largest and second largest comunities grow as the tollerance increase while the number of comunities decreases reaching the value of two. This means that the system reaches a situation in which almost all agents belong to two dominant comunities. This is a state of fragmentation/polarization.

For $0.20 < \epsilon < 0.25$ the number of agents in the largest comunity do not grow while the number or agents in the second largest comunity continue to grow as long as only two comunities survive.

For $\epsilon > 0.25$ the system changes behaviour abruptly and all agents belong to a unique comunity reaching a consensus state.

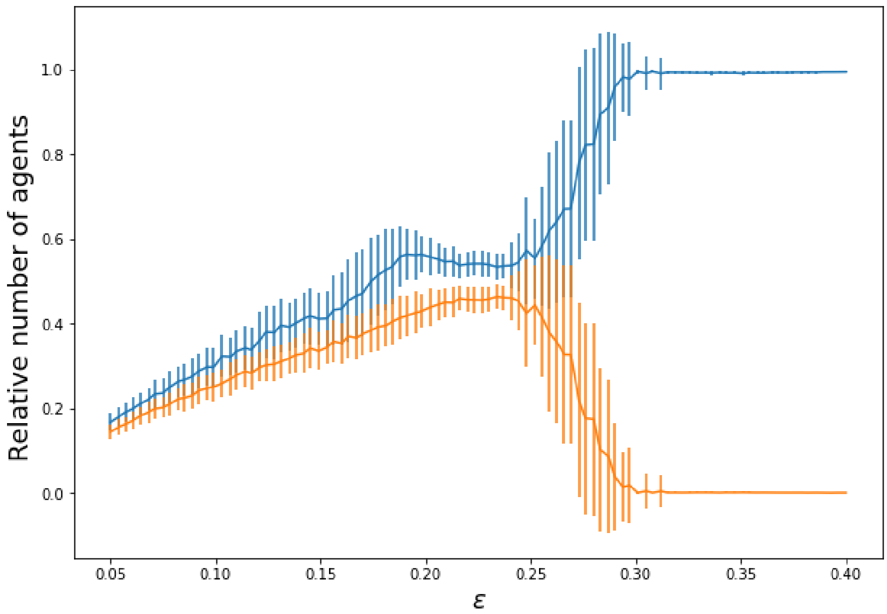
For tollerance in the interval $[0.25, 0.30]$ we can think that there is a phase transition from polarization to consensus as we can see in figure 9a. To prove this, we should try to reproduce the experiment with a greather number of agents and see if the transition becomes sharper. Another interesting feature we can see in fugure 9b consists in the fact that the number of comunities in function of the tollerance behaves as $1/2\epsilon$, or rather, as the inverse of the probability an agent has to select randomly another agent with similar opinion.

5 Conclusion

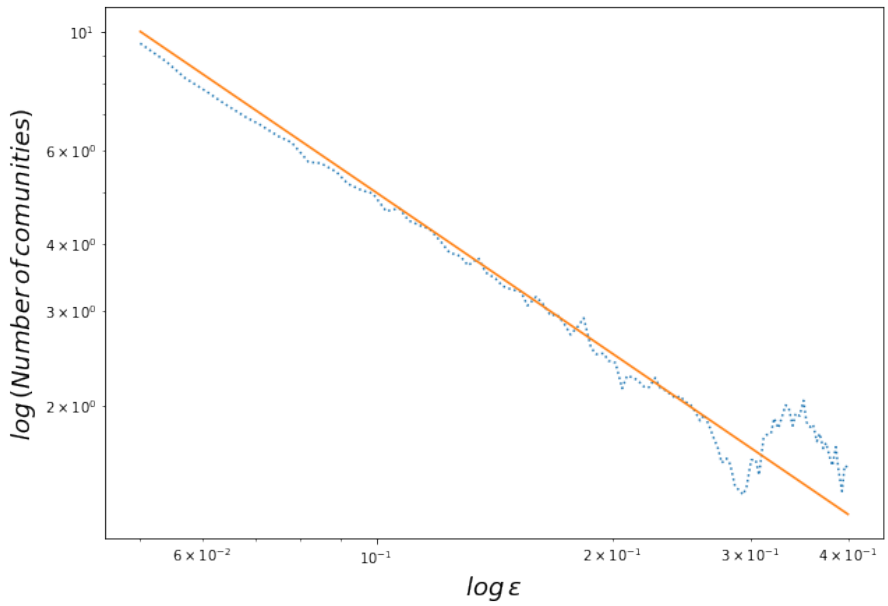
In this work I have studied how consensus emerges in network with different topological structure, random or scale free, and with different dynamics, namely static or adaptive.

In static networks, links are freezed and two agents can try to excange opinions only if they start linked. In this way two agents can comunicate only if there exists a path of agents who have similar opinion, otherwise they can not comunicate. In adaptive network an agent can rewire a link with someone with distant opinion and search for someone in the network with a similar opinion. This property permits the formation of larger comunities but it prevents the reaching of consensus for which the system needs an higher tollerance compared to the static case.

Comparing random and scale free network I found no considerable differences, except for the static case. Indeed the scale free network behaves in a different way needing great value of tollerance to reach consensus because hubs tend to obstruct the comunication between different region of the network. It may be interesting to investigate how this behaviour changes with different generating algorithms that permit to control the number of links added by a new node. As a conclusion, I found no difference between an adaptive random network and an adaptive scale free network in the opinion dynamics. I think it may be usefull to think about the reasons of that consideration, studying how topological structure of different adaptive networks changes with rewireing property for example.



(a)



(b)

Figure 9: Results of the simulation. In 9a there are the mean relative number of agents in the biggest (blue) and in the second biggest (orange) comunity in function of the tollerance with relative standard deviation as vertical bars. In 9b there is the mean number of final comunities in function of the tollerance, plotted with blue dots in $log-log$ scale, while in orange the function $y = 1/2\epsilon$ is plotted.

References

[1] G. Deffuant, D. Neau, F. Amblard, G. Weisbuch; Mixing beliefs among interacting agents 2000

[2] R. Hegselmann, U. Krause ; Opinion dynamics and bounded confidence models, analysis, and simulation, 2002

[3] B. Kozma, A. Barrat; Consensus formation on adaptive networks, 2007

[4] Y. Yu, G. Xiao, G. Li, W. P. Tay, H. F. Teoh; Opinion diversity and community formation in adaptive networks, 2017

[5] M. E. J. Newman; Networks, an introduction, 2010

[6] A. Barabasi; Network science, 2014