ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

Performance of strong and weak link networks in opinion formations

Marc Vetter, Paul Sornette & Jean Baptiste Heimsoeth

Zürich
December 2010

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich diese Gruppenarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen-Hilsmittel verwenden habe, und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichen Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Gruppenarbeit nicht, auch nicht auszugsweise, bereits für andere Prüfung ausgefertigt wurde.

Marc Vetter Paul-Emmanuel Sornette Jean-Baptiste Heimsoeth

Agreement for free-download

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

Marc Vetter Paul-Emmanuel Sornette Jean-Baptiste Heimsoeth

Table of content

Introduction and Motivations

"In this world, the measure of power is connectedness"

Ann-Marie Slaughter
Director of the US Department for Policy Planning since 2009

We believe: The validity of this statement is not restrained to governments, institutions or organizations. It can be translated to the environment of an individual human being, at any time, under any circumstances.

Information is power, and connectedness represents highly paced access to information: So far so good. But connectedness is also profoundly human. There is nothing more innate to humanity than communication, which is both origin and aim of connectedness. However, the essence of communication is that it occurs primarily on a subconscious level: The amount of information purposefully relayed is dwarfed by what's eventually conveyed to and subconsciously processed by the receiver. In a face-to-face conversation, additionally to its content, we register speaking tone, silence spacing, choice of vocabulary, physical attitude and many more. The same works for written communication: Punctuation, spacing and message context tell us a lot. The point is: Sometimes, we will consciously or subconsciously, take over one of these elements and pass it over to the next person.

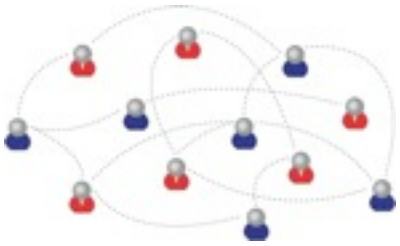
We have decided to analyze the pattern of propagation of such an element, a "pulse" originating in one specific individual, inside of a group, a "network", or individuals. The question we are asking is "How fast does a pulse propagate in a homogenous network, depending on the external factors of group connectedness, group average sociability, individual prominence and the degree of bilateral in?"

Means of communication
Average sociability: homogenous because same starting point, access to internet, landline phone, cell phone, daily activity & environment

In this paper, we will see the implementation of the model, it's translation into equations. We will discuss the results and finally we will show the entire Matlab code we used explained in detail.

Implementation

How to model a social network



In order to implement a social network, we first have to collect data. The data that is of interest for this study are the number of agents in our system and their social behavior.

We made the data by creating an agent standard type. We gave him a certain sociability, a certain connectivity rate and a belief rate (to what extend this agent accepts an idea once given the opportunity of choice). From this standard agent which we named Ben, we duplicated his entity but we randomly changed his personality parameters. We then got an entire population. Doing this steps a few times, we finally got a networks of twenty thousand agents. This is about the size of the population of ETH.

But with such a big network, our computer didn't manage to calculate the interactions. We then tried another approach that involved only a small network. It is made of sixty-four agents exactly. We had to make it very representative so that, this small population still includes the "intelligence" of the big network.

Then, we tracked the interactions between the agents (how many connections, how intense, how long, how often...) and make from it a simple mathematic model. So one for each agent. It is as we graded each agent according to it's sociability.

It order to be able to compare the agents, we had to weight their interactions. The importance of an agent comes from how many connections it as and/or how strong the connections are. To give an example, an agent with a network of 5 agents and strong sociability isn't equal to an agent with 100 agents and a weak sociability.

Agents are potentially connected to all others, but really connected to only a fraction of the population of agents. These connections are modeled by a connection matrix.

The agent based model we created gives us the entire spectrum of networking.

How to put into equations

We had to group interactions of a subject into categories. One can only have so much interactions, either a lot of connections which are of low importance (=weakly weighted) or just a few connections which are of high importance (=heavily weighted). It may also be the case that someone is not fully in either of these two groups such that he as a few strong connections(friendship) and at the same time weak connections/friendships.

More practically, we defined the number of agents, then we defined by 1 and 0 the presence or absence of a connection between two agents. Subsequently, each connection is weighed. The initialization for the connections and the weigh of the connections is made by the program CONNECTION_MATRIX. The program gives us a connection matrix which will be used later.

An important remark: The matrix isn't symmetric: if agent A influences agent B, it doesn't mean agent B influences agent A.

Thus the total subset of agents (size) may be split into smaller groups of people interacting with each other.

We then implement an idea randomly in a few agents(dopesize) within the big network. It may be that in a small network, none of the agent have the idea, one has the idea or a few get the idea at the same time.

Using the model and the program

We described earlier how we get or create the data and equations we used. We explained how we got a model of the network. Now the research begins.

This hole study finally tries to answer a simple question: will an idea be adopted more rapid and spread reliably if it is implemented into a network where agents have a few strong connections as opposed to a network where agents have many relatively weak connections..

Now that we understand what is at stake, we can play with the number (coefficients and probabilities) to find some prediction in the results. Those will of course depend of coefficient we use for the data depending of the society, country, education and lots of other parameters.

We manage to get the results in 3D plots. It enable us to quickly see and decide which parameters are more dominant. For some plots, we used theses 3 axis: sociability, probability of connection, how long it takes for a percentage of the networks to be convince: we decided to take 20%, 50 % and 90% as the thresholds.

The probability of a connection between agents varies from 0.1 to 1. The strength of the connection, how social an agent is, how many good friends he as in life varies with some resemblance within a group or small network. For example, if one agent has 20 connections, maybe 4 of them are his close friends. His social influence will be close to 1/4 for the connections to those friends, but close to zero for the rest. The sociability varies for example from 1 to 20 . If an agent A has a sociability of 1 and has 2 connections (knows two people), than each of the connections will weigh on average w=0.5. We introduced a strong random element influencing every connections to allow agents to have diverse connections (from friendships to "just met once"). The total network sociability is held constant in order to be able to compare the networks. Together with the connection probability, it defines the social networks.

The model updates each agents state (0 idea not accepted , 1 idea accepted) by:
- checking which of his friends carries the idea.
- multiplying the idea strength (in our model : 1) with the strength of the agent-friend connection
- summing over all friends who carry the idea
- comparing the result to the required threshold (in our model : 0.9)
- changing the state to 0/1 if the result is smaller or larger respectively

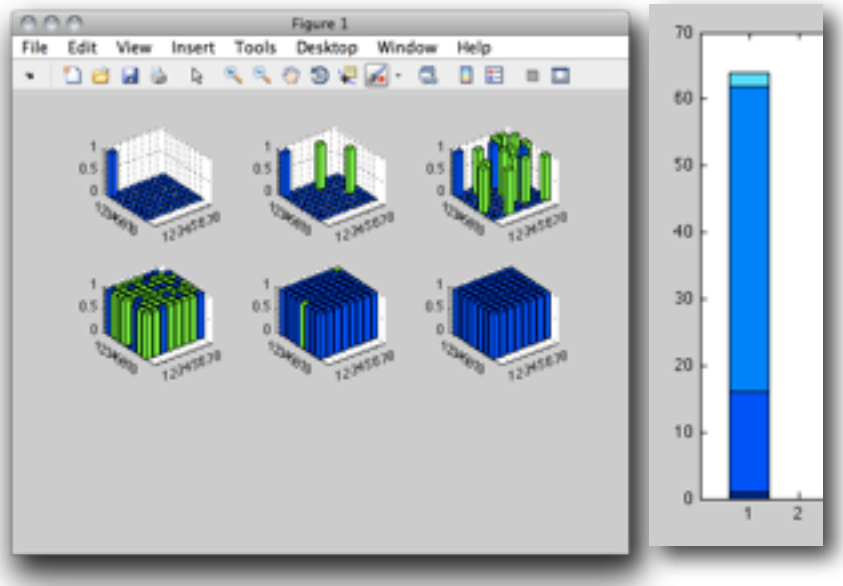
In addition each connection carries a random element with a strength of 0.5% (representing small scale changes in relationships from day to day) as well as an element trying to simulate agents who cherish or oppose ideas already accepted by several others (also with a strength of 0.5%).

It is important to note that "sociableness" is averaged. This means that is the input parameter sociableness=5 that all agents have sociableness 5. There is no age ; gender or other parameter which distributes this sociableness between values; it is easily implemented though.

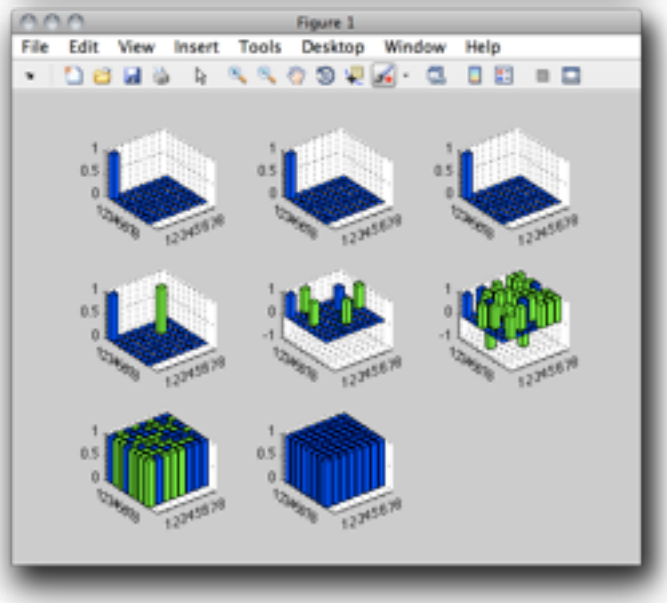
Simulation Results and Discussion

The Simulation yields some interesting results. As expected an idea travels faster the higher the sociableness of the agents within the network is. More surprising is that a network carries an idea faster and more reliable if the agent know less of the network eg. has a few good friends. Partly this can be explained by the initial condition where one agent carries the idea. If this agent does not influence at least one other agent sufficiently the idea will never spread. The option to control the amount of agents carrying the initial idea is already implemented in the code (dopesize) but has as off now not been used.

The Following Figure is created by “histproject” and shows a network of 64 agents. The Probability of two agents knowing each other was set to 15% and the sociableness to 10.

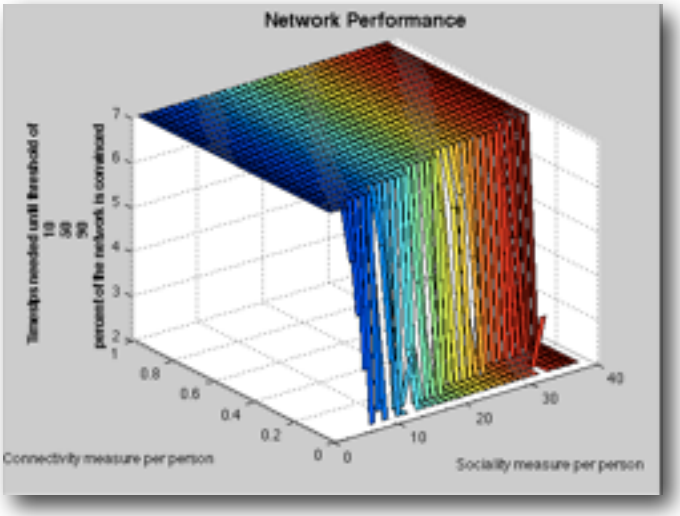


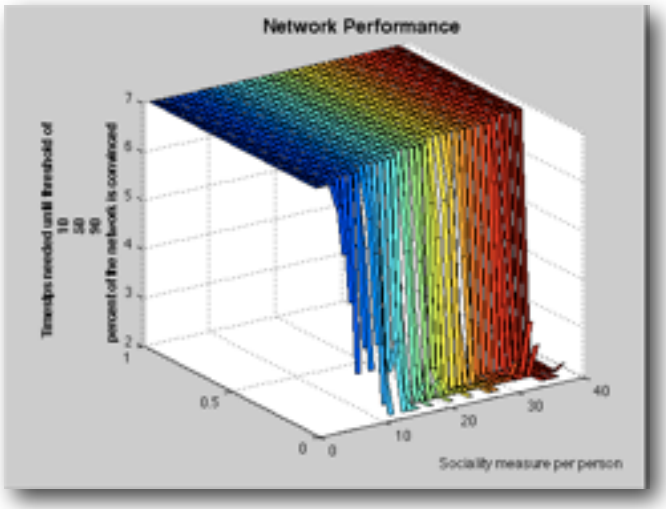
The six small images on the left show the agents which adopt the idea at each timestep. At the first step only the dopesize , in this case one agent , carries the idea. In the next step he convinces two agents of the idea until all agents are convinced. The graph on the right shows how many agents are convinced at each step.

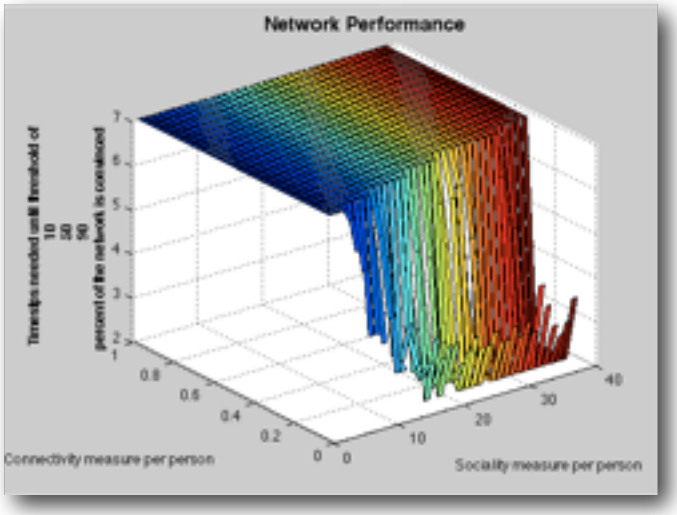


The following figures belong to a simulation with the exact same parameters. Here we see the interesting behavior that for three timesteps the idea does not spread. Then it manages to spread to a second agent upon which the idea manages to spread throughout the whole network. Another interesting occurrence is the fact that two agents switch back to not carrying the idea at timestep six.

These differences show the level of complexity in our model and justifies the use of multiple





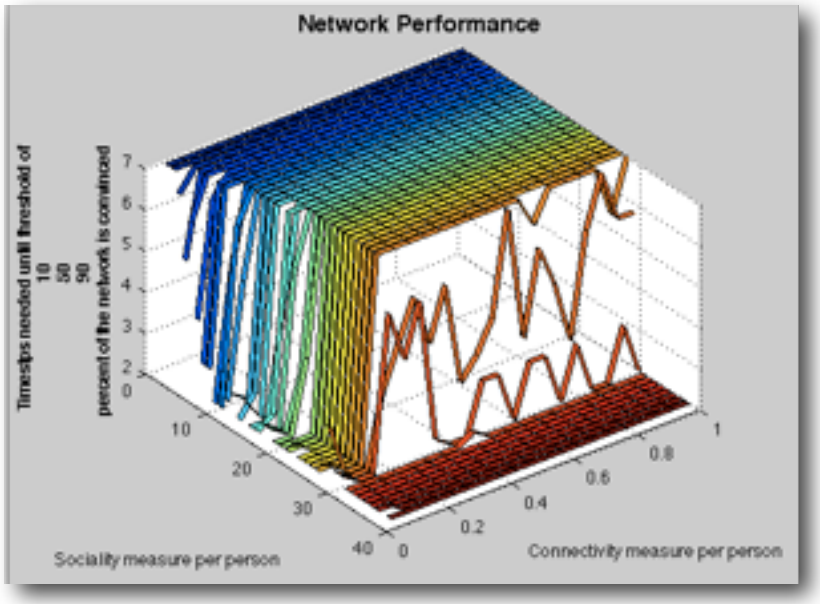


simulations with the same parameters in the subsequent examples.

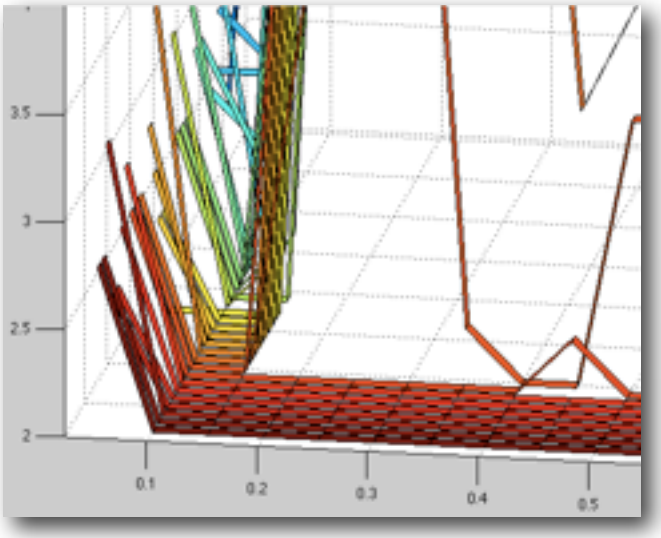
These figures are created by datathresh and shows a network of 30 people. Each plot shows the amount of timesteps needed for the idea to be spread to 10% - 50% - 90% of the network from top to bottom respectively. The higher the value the slower the idea propagation.

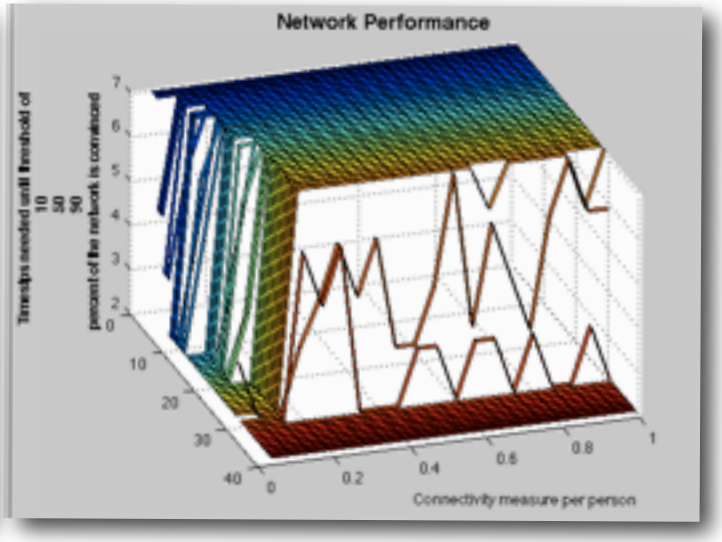
The sociality measure (sociableness of each agent) is in fact only half of the indicated value thus reaching from 1.5 to 20. As intuition would imply the speed by which the idea spreads here forth referred to as the network performance increases with the sociableness of the agents. A much stronger measure of network performance is however the connectivity measure per person. In the last plot (timesteps for 90% of the network to be convinced) we can see that an optimal performance is reached between a connectivity of 10-15%. Thus an idea spreads faster and more reliably if agents in a network know less people but therefore know them better. Even with a relatively low sociableness of an agent a complete idea propagation throughout the network can still be reached if the connectivity measure is sufficiently low.

These figures show the exact same thresholds but for a network of 20 agents. This is a critical comparison because we assumed that a typical sociableness lies between 1.5 and 20. Within a network of 20 agents this still allows for a connection with everyone with a relatively strong weighting of 1.



At a network of this small size - of the order of the individual sociableness, the sociableness's role becomes as significant as the connectivity's. As implied however the connectivity does not loose its importance. Remarkably this network is showing the highest performance at the same interval of connectivity (10%-15%) as the larger network.





Summary and Outlook

As expected, the higher the sociability, the faster an idea travels in a network. Interestingly the lower the connectivity (meaning the better I know my friends) the higher and more reliable the performance. This effect outweighs the effect of sociableness. A similar result was found in a recent study by the MIT (check references for the link)

Dopesize

The dopesize represents the amount of agents within a network which carry the idea at the time zero. Within all sizes of networks it will be interesting to see what influence this parameter has on network performance. With a larger dopesize the influence of the connectivity is believed to diminish. The question wether an idea propagates from one initial carrier or is put into a network is important to answer. An advertising campaign will fall under the latter while a new idea can only ever originate in one agent. The code is already written and can be activated.

Ideastrength

An idea can carry an initial weight. The code is already written and can be activated. Several ideas may also be pitted against each other.

Initial inclination

Agents can have an initial inclination to accept particular ideas. For example some agents might be willing to accept an idea related to health and nutrition because of an overall interest for the subject and new findings while they can have an aversion towards technological ideas / behaviors. The code is already written and can be activated.

Bigger networks

To validate the findings it will be helpful to run the simulations as they are on more powerful machines. The code has already been altered as to run in parallel (parfor) but the desktop computers at our disposition where not concluding the simulations at network sizes of 50 agents or larger.

Agent sociableness

Agent sociableness varies in reality but is approximated in the model. It is straightforward to implement networks with agents who have different sociableness and will make the model more accurate.

Individual Contributions & Acknowledgements

We would like to begin with thanking Giovanni Luca Ciampaglia, Karsten Donnay and Stefano Balietti for the extensive help and support they have given us the possibility to take advantage of during the course of our project. We understand and appreciate the effort they have put in supervising it. To the exception of the assistance we received from the moderators of the MatLab Documentation forums, no further advisors and consultants have been needed ; Without the generous help those individuals and our faculty examinors team, this paper would not have been possible.

References

http://en.wikipedia.org/wiki/Social_network

http://www.mathworks.com/

http://mathworld.wolfram.com/CausalNetwork.html

http://thesituationist.wordpress.com/2008/03/03/social-networks/

http://web.mit.edu/newsoffice/2010/social-networks-health-0903.html

Matlab code

We wrote these five programs: **findthreshold, project, histproject, connectionmatrix, datathresh.**

We will now look at how they work.

```
function[steps]=findthreshold(ideapropagation,absthreshold,timesteps)

steps=timesteps;

for i=1:length(ideapropagation)
   if ideapropagation(i)>=absthreshold
      steps=i;
      break
   end
end
```

```
%agent based model of network
% 12000 people work at ETH ; 8000
undergrads ; 180 graduates at MAVT Dep.

%Initial State

function[ideadata,ideapropagation]=project
(size,P,S,timesteps)
%size = number of people
%P = Which percentage of the group does
the Subject know
%S = Social Interaction per agent


time0=0;
dt=1;
timeend=timesteps;
ideapropagation=zeros(1,timesteps+1);


%number of Nodes
reqstrength=.9;
ininclination=0;
ideastrength=1;


%total Network social interaction
%S=S*size;

%average probability of connection between
nodes
%P=P/size;

%connection matrix
connection=connectionmatrix(size,P,S);
%Initial State of Agents
% Idea = 1 ; Stupid = 0

idea=zeros(size,size);
ideadata=zeros(size,size,timesteps+1);
strength=ones(size,1)*ininclination;
%[~,zeile]=max(connection);
%[~,spalte]=max(max(connection));
%strongestinfluence=connection(zeile
(spalte),spalte);
%idea(zeile(spalte),spalte)=ideastrength;

%dope 1/100th of the subjects with the idea
%dopesize=round(size/100);
%if dopesize==0
   dopesize=1;
%end
idea(1:dopesize,1:dopesize)=ideastrength;
ideapropagation(1)=sum(sum(idea));


%time loop

for t=time0:dt:timeend

   ideadata(:,:,t+1)=idea;

   %agent loop
   for agent=2:size


      %idea loop
      %for i=1:length(idea)

      %Strength loop ; calculate strength of
idea around agent
      for friend=1:size
         sign=0;
         %generate random sign
         sign=0.5-rand;
         if sign<0
            sign=-sign/sign;
         else
            sign=sign/sign;
         end
```

Rest of the code from the page above

```
%including random element and contra
feeling (people influenced by group
decision / either reinforcing or repellent)
      strength(agent)=strength(agent)
+connection(agent,friend)*idea(friend,friend)
*(1+0.005*randn)+sign*0.005*sum(sum
(idea))/size;
      end %end strength loop

      %end %end idea loop


   end %end agent loop

   %2nd agent loop
   for agent=2:size

      if strength(agent)>reqstrength
         idea(agent,agent)=1;
      else
         idea(agent,agent)=0;
      end
   end %end 2nd agent loop


   %reset strength to idea carrier
   strength=ones(size,1)*ininclination;
   strength(1)=ideastrength;

   if t>=1
   ideapropagation(t+1)=sum(sum(idea));
   end

   %stop uninteresting samples if condition
"last 3 states all equal" is
   %met
   if t>=3
   if ideapropagation(t)==ideapropagation
(t-1)&&ideapropagation(t)==ideapropagation
(t-2)
      ideapropagation(t:timeend+1)
=ideapropagation(t);
      return
   end
   end


end %end time loop
```

```
function[]=histproject(size,timesteps,P,S)
%use rectangular size

if sqrt(size)~=round(sqrt(size))
   display('choose a rectangular size')
else


[ideadata,ideaprop]=project
(size,P,S,timesteps);


%transform into rectangular matrix
network=zeros(sqrt(size));

for t=1:timesteps
   i=0;
   for a=1:sqrt(size)
      for b=1:sqrt(size)
         i=i+1;
         network(a,b,t)=ideadata(i,i,t);
      end
   end
end

tobeplotted=zeros(sqrt(size));
tobeplotted(:,:,1)=network(:,:,1);
for t=2:timesteps
   tobeplotted(:,:,t)=network(:,:,t)-network
(:,:,t-1);
end

%these are for colors
name='bg';

%plot the whole thing
figure
subplot(round(sqrt(timesteps)),round(sqrt
(timesteps)),1)
bar3(tobeplotted(:,:,1),name(1))

for t=2:timesteps

   subplot(round(sqrt(timesteps)),round(sqrt
(timesteps)),t)
   bar3(tobeplotted(:,:,t),name(2))
   hold on
   bar3(network(:,:,t-1),name(1))

   if sum(sum(network(:,:,t-1)))==size
      break
   end

end

%how many switched opinion each step
oestep=zeros(length(ideaprop));
oestep(1)=ideaprop(1);
for i=2:length(ideaprop)
   oestep(i)=ideaprop(i)-ideaprop(i-1);
end

%plot it
figure
bar(oestep','stacked')
end
```

```
function [connection]=connectionmatrix
(size,percentofnetworkknown,sociability)

P=percentofnetworkknown*100/size;
totalnetworksociability=sociability*size;

%Initialize random connectivity matrix
connection=rand(size,size);

%compare to get IO Matrix
for i=1:size
   for j=1:size
      if connection(i,j)<=P
         connection(i,j)=1;
      else
         connection(i,j)=0;
      end
   end
end

%make diagonal zero -> self influence zero
connection(1:size+1:size*size)=0;

%find number of connections
%numberofconnections=sum(sum
(connection));

%average weight of connection
%weigh=totalnetworksociability/
numberofconnections;

%weighmatrix with .3 noise
%weighmatrix=weigh*(ones(size,size)
+0.3*randn(size,size));

%weighted connectionmatrix multiplied
elements wise
%connection=connection.*weighmatrix;

%connection=connection/(sum(sum
(connection))/size);

%loop to weigh all connections
for i=1:size
   %find total connections to this node
   nrconnections=sum(connection(i,:));
   for j=1:size
      if connection(i,j)==1;
         %weigh connection depending on
how many other friends the agent
         %has , how sociable he is and a
random element of 5%
         connection(i,j)=(sociability-sum
(connection(i,1:j-1)))/nrconnections*
(1+0.05*randn);
         nrconnections=nrconnections-1;
            if connection(i,j)<0
               connection(i,j)=0;
            end
         %for average connections
         %M=sum(sum(connection));
         %weigh=totalnetworksociability/M;
      end
   end
end

%make sure the total network sociability is
correct
realnetworksociability=sum(sum
(connection));
f=totalnetworksociability/
realnetworksociability;
connection=connection*f;
```

```
function[]=datathresh
(size,timesteps,n,threshold)

%calculate absolut threshold
absthreshold=threshold*size;

%initialize global variables
rows=20;
columns=38;
numberofplots=length(threshold);

data=zeros(rows,columns,n,numberofplots);
avg=double(zeros
(rows,columns,numberofplots));

for example=1:n
   %initialize row variable to keep track; for
writing the result into data

row=0;

for P=0.05:0.05:1
   %initialize column variable & update row
variable to keep track; for writing the result
into data
   column=0;
   row=row+1;

   for S=1.5:0.5:20
      %update column variable to keep track;
for writing the result into data
      column=column+1;
       %execute network / first element would
be "networkideapropagation"
      %if gathering of that data is enabled
      [~,k]=project(size,P,S,timesteps);

      %find the threshhold
      for i=1:numberofplots
         numberofsteps=findthreshold
(k,absthreshold(i),timesteps+1);

         data(row,column,example,i)
=numberofsteps;
      end

   end
end

end

%get the average over all examples
for i=1:numberofplots
for row=1:rows
   for column=1:columns
      avg(row,column,i)=sum(data
(row,column,:,i))/n;
   end
end
end

for i=1:numberofplots
figure
X=1.5:0.5:20;
Y=0.05:0.05:1;
%surf(X,Y,avg(:,:,i))
%hold on
%waterfall(X,Y,avg(:,:,i))
ribbon(Y,avg(:,:,i))
xlabel({'Sociality measure per
person'},'fontsize',10,'fontweight','b')
ylabel({'Connectivity measure per
person'},'fontsize',10,'fontweight','b')
zlabel({'Timestps needed until threshold
of',threshold*100,'percent of the network is
convinced'},'fontsize',10,'fontweight','b')
title('Network Performance','fontsize',
14,'fontweight','b')
end
```