



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

Insert Title Here
...

Name 1 & Name 2

Zurich
May 2008

Agreement for free-download

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

Name 1

Name 2

Contents

1	Abstract	4
2	Individual contributions	4
3	Introduction and Motivations	4
4	Description of the Model	5
5	Implementation	6
5.1	Generating a Population	6
5.2	Generating the Social Network	7
5.3	Timesteps and Iterations	7
6	Simulation Results and Discussion	7
7	Summary and Outlook	7
8	References	8
	Bibliography	8

1 Abstract

2 Individual contributions

3 Introduction and Motivations

Human behaviour is chiefly governed by societal and environmental influences. Social interactions, nurture and upbringing account for an individual's identity, their choices and way of life, which gives rise to the notion that sociological and behavioural phenomena could be transmitted via social networks in a manner comparable to a viral contagion.

Traditionally, the term "contagion" is associated with strictly epidemiological factors, i.e. infectious diseases or viruses that spread through networks by human contact. In a similar vein, we can extend this connotation by considering a "social contagion" as a behavioural pattern that propagates inter-personally through a *social* network via interaction. Thus, social contagions have the ability to become quintessential determinants for an individual's health and mental well-being. Based on this premise, it enables us to study behavioural phenomena using network analytics and classic epidemiological models, albeit most of those will be subject to slight modifications in order to be compatible with the problems we are concerned with.

One such instance of a recent issue we are facing is the prevalence of obesity in modern society, which may be regarded as an epidemic in its own right. As of 2017, approximately 40% of the Swiss population is overweight and a further 10% is obese. In juxtaposition to figures of 1992, this represents a full 100% increase in the last 25 years [3]. This issue is not limited to Switzerland, by all means – We observe similar trends, if not starker ones, throughout the rest of the globe (in fact, Switzerland holds up comparatively well in that regard). Hitherto, the obesity epidemic cannot be adequately explained by genetic factors (genetics may merely affect one's predisposition towards obesity) [2] and the epidemic permeates through all ages and socioeconomic groups. Hence, as we are lacking congenital or medical explanations to expound this sudden upsurge, this allows the theory of a social contagion to step to the fore.

The hypothesis is that an individual's attitude towards obesity and one's habits change as those around them do. For instance, a person's tolerance for weight-gain may be increased if they converse with multiple contacts that are obese, and group activities like eating-out or smoking may be forced upon an individual by means of

peer pressure. Furthermore, if a child has obese parents, this might reflect in the child's upbringing and diet.

In the following, we implement a model to simulate these aforementioned considerations on the basis of a social network that will be representative of the demographics of Switzerland. We will try interpolate between the figures we observe from 1992 and 2017 and moreover, attempt to make a prediction for the "disease" process in the years to come. Ideally, we will want to confirm the correlation between the spread of obesity and social influence, or in the very least, showcase social contagions and the modelling of those as a viable consideration for a plethora of behavioural phenomena and pseudo-epidemics.

4 Description of the Model

The model we are using is a modification upon the so-called SIS-model as described in REFERENCE which is conventionally used in the mathematical modelling of epidemiology. Similar to a regular infection, our population can generally enter two different states, "susceptible" and "infectious", respectively. The reason for choosing this model as opposed to alternatives like the SIR-model is due to the fact that our individuals cannot develop immunity and enter a "recovered" state upon recuperation - an individual recently recovered from obesity is still as vulnerable to relapse as any other individual who never contracted the disease.

The SIS-model acts upon a network which will be model with a connected graph of agents, where every connection represents a social connection, e.g. "friendship" or "family", with the caveat that for simplicity, we do not differentiate between different types of relations. Contracting obesity from a friend is thus equally as likely as contracting it from a relative which is not the most realistic assumptions - studies do show that transmissions vary depending on the type of relation two individuals have [2]. This may lead to discrepancies and will need to be kept in mind when evaluating the results of the simulation.

The main modification that is made to the traditional SIS-model is that the spread is not strictly limited to transmission [1]. Thus, beside the "transmission" and "recovery" rates, we need to introduce one additional rate, which is the "spontaneous rate". As the name suggests, at any given time, a single agent may develop obesity spontaneously independent of the state of those around them. By introducing this rate, it is ensured that the disease will never die out within our population, which is in line with reality.

In the altered model, the spontaneous rate will serve as a base rate for every agent upon initialisation. Depending on the transmission rate and number of infectious contacts an individual has, the overall likelihood of contracting obesity will then increase, hence the probability is cumulative over the number of infected contacts an agent possesses. The more connections an individual has, the more the impact of one single obese contact will be diluted. This mean for instance that an agent with two friends which are both obese is more likely to become obese then one with 100 friends out of which two are obese. The rationale behind that decision is that the fewer social contacts one agent has, the more time is spent with each one. The recovery, spontaneous and transmission rates will remain static in our model and will only be set at the beginning of the simulation.

In our simulations, we will use the rates presented in [1] as a starting point and simulate 25 years (1992 - 2017) of development. One must bear in mind that these rates were found in population studies pertaining the US, thus are not likely to give us the figures we observed in Switzerland in 2017. Hence, we will use the spontaneous and recovery rates as parameters in order to find the numbers via which we obtain the Swiss obesity figures we are looking for. Note that the transmission rate will not serve as a parameter in our simulations, as we assume that Switzerland and the US do not differ in that regard as strongly as in the other two rates.

5 Implementation

5.1 Generating a Population

Individual agents are implemented in the Python class *Agent*, containing multiple attributes that describe an individual in the population. The main attributes are *age*, *sex*, and *obese*. The age of an agent is stored as a number, the sex is either 'f' or 'm', and the obesity of an agent is a Boolean value. Additionally, there is a variable called *next_state*, that is used in each step of the iteration, as described in chapter 5.3.

To create a population of agents, the methods *createSwissAgents1992* and *createSwissAgents2017* are used. Given a number n , these methods return a list of n agents whose ages and obesity are distributed according to the age structure¹ and obesity rates² for females and males by age group in Switzerland in 1992/2017.

5.2 Generating the Social Network

Next, we want to generate a random graph whose nodes and edges represent agents in a social network and friendship connections, respectively. To generate networks we use the Python library *NetworkX* which lets us create and manipulate graph structures.

Given a list of agents, the method *createNetwork* adds nodes containing the individual agents to an empty graph. Next, connections between the agents are generated using...

5.3 Timesteps and Iterations

After initialising a population and generating the friendship-network, the function *simulate* takes the network and the global rates of transmission β , recovery γ and spontaneous infection α , as described in chapter 4. Then, it simulates the spreading-dynamics of obesity in the network. It does this by calling the *step* function. This function iterates over all agents in the network and calls each agent's *interact* function. Then it iterates over all agents again, and calls each agent's *update* function.

In the *interact* function, if the agent is currently obese, he will recover (set *next_state* to *False*) with probability $\mu \cdot \gamma$. Otherwise, if the agent is not currently obese, his initial probability to become obese is $p = \mu \cdot \alpha$. For each obese friend, p increases by $\mu \cdot \frac{\beta}{k}$. Finally, he becomes obese (sets *next_state* to *True*) with probability p .

In the *update* function, the agent updates its obesity state to *next_state*. The *interact* function is shown more accurately in *Algorithm 1*.

6 Simulation Results and Discussion

7 Summary and Outlook

Algorithm 1: *interact* function of an agent.

Input: Agent *self*, Graph *G*, rates of transmission β , recovery γ , spontaneous infection α

Output: agents next_state is set

```
1  $\mu \sim N(1, 0.25)$ ;
2  $\epsilon \sim U(0, 1)$ ;
3 if self is obese then
4   if  $\epsilon < \mu * \gamma$  then
5     next_state = False;
6 else
7    $p = \mu * \alpha$ ;
8    $k = \text{size}(\text{contacts}(\text{self}))$ ;
9   for contact in contacts(self) do
10    if contact is obese then
11       $\delta \sim N(1, 0.25)$ ;
12       $p += \delta * (\frac{\beta}{k})$ ;
13    if  $\epsilon < p$  then
14      next_state = True;
```

8 References

Bibliography

- [1] Age structure of switzerland 2017.
- [2] Bundesamt für Statistik. Body mass index (bmi) nach geschlecht, alter, bildungsniveau, sprachgebiet - 1992, 1997, 2002, 2007, 2012, 2017: Tabelle.