Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Complex Social Systems, Agent-Based Modelling, Learning and Games

Project Report

Individual Advantage and Group Identities in Agent-Based Opinion Formation

Han Xi, Daniel Spathelf, Likuan Wang, Suada Mirgadirova

Zurich
December 2021

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

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

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

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

Title of work (in block letters):

Individual advantage and Group Identities in Agent based opinion formation

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Spathelf	Daniel
Wang	Likuan
Xi	Han
Mirgadirova	Suada

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

Place, date

Zürich, 5.12.21

Signature(s)

Likuan Wang

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1 Abstract

In our project, we aim to gain a better understanding of opinion spreading in medical communities and its dependency on relations between distinct agents in respective city clusters. We model physicians as agents and based on the adoption dates of a new drug and adjacency matrices obtained from data concerning relationships between physicians in several cities train a machine learning model consisting of neural network layers with an algorithm that is physically inspired by a process of diffusion. The goal of it is to fit four parameters we have identified to be influential in opinion spreading, namely three factors of relationship strength between physicians (friendship, advise and discussion relations) and profit constant obtained from adopting an innovative medication. Upon obtaining fitted parameters on different clusters of physicians we discuss the applicability of the results in the prediction of opinion percentages in other clusters over time concerning the adoption of a new medication by simulating the algorithm with the obtained parameters on other clusters and analyzing the results.

2 Introduction and Motivations

The goal of this work is to study opinion formation on the case study of the medical communities in different cities. Using the data collected in ([4] Coleman et al. - Diffusion of an Innovation Among Physicians study, 1957), inspections were done how well the opinions of physicians regarding different prescribed drugs can be predicted, from only looking at how doctors interact with each other. The doctors are assumed to form their opinion according to their social interactions with the other doctors via friendship (f), advice (a) and discussion (d) as well as the intrinsic advantage (A) of one drug over the other in form of an individual "profit" they see in this. Therefore the Probability P_j of the doctor j to change his opinion is the following function:

$$P_j = P_j(\mathcal{N}, A, t) = P_j(\mathcal{N}(t, f, a, d), A(t), t)$$

Where $\mathcal{N}$ represents the social Network effect with its time dependency t. The advantage A is in principle also time-dependent, but it is assumed to be not, because the data set has no Information about this. Below the physical inspiration for the model and the concrete micromodel for the probability of change are explained in detail.

2.1 Physical inspiration for the model

Diffusion (from lat. $diffundere$, that means to spread out / scatter) describes the process of an entity spreading out. While the process of diffusion is in most cases treated as a solely physical process of the Diffusion of Mass (Mass Transfer), in this work the basic principles have been applied to a social system where the spreading of a new drug that is prescribed by physicians is examined (Diffusion of an Innovation Among Physicians). The opinion forming mechanism assumed in this study is inspired by the physical problem of diffusion:

$$J = -K \left(\frac{\partial \mu}{\partial x} \right)_{p,T}$$

where J is the flux of Mass per area and time, K is the Proportionality Constant and $\left(\frac{\partial \mu}{\partial x} \right)_{p,T}$ is the Gradient of the driving force (Chemical Potential).

The physical process (the diffusion) that is treated as the mass transfer over the contact area (boundary layer) between control volumes is abstracted in the general setting of opinion formation. It is looked at as the "transfer of opinions" over the "social contact area" which is the social influence of one person on another. The people are treated as discrete agents. In contrast to the physical process, the social influence is not symmetric. Person i may influence person j more than j influences i. Another difference is, that the basic physical laws - conservation of mass and energy, which influence the outcome of physical diffusion, do not apply to social systems. In fact, this is just another way to describe that interactions between agents may be asymmetric. The model set is a 2-state social system with N agents (doctors). The state s of the agent I defines the opinion of the doctor on prescribing one drug or the other (s being 0 = old medicine or 1 = new medicine).

The model describes the probability of the agents to change their opinion in a fraction of time and is, therefore, able to predict a time evolution of opinion dynamic in medical communities.

3 Description of the Model

3.1 Time evolution and macro model

The model is evolved in number of time steps according to the time steps of the data set (see more below). We calculate the driving force G of agent j, with the micro model explained below. Two different approaches for the random dataset and the data-driven simulation have been chosen.

-For the random dataset: firstly the driving forces G_j are normalized such that each value in the time step falls in the interval [-2, 2]. Secondly, we apply the cumulative distribution function of the standard normal distribution to obtain the probability of the agent's opinion change:

$$P_j = \mathrm{erf}(G_j) = \frac{2}{\sqrt{\pi}} \int_0^{G_j} e^{-t^2} dt$$

-For the data-driven simulation: in each time step, we use the driving forces of all agents $\vec{G}$ to calculate the probability (P_j) for each agent j to change their opinion:

$$P_j = f(\vec{G})$$

The function above is calculated using a machine learning scheme explained in section 4.2. For the opinion change, we have defined two different functions, one that does not allow our agents to go back to using the older drug once they have adopted the new one in their practice. The reason behind that is that in our used data set neither of the physicians go back to using the older medication after having adopted a new one and to that end we have implemented our opinion function accordingly. The other one does allow the agents to go back to the old drug. These are compared.

3.2 Micro model

The physical mass transfer model is remodeled for the chosen 2 state social system mentioned above which is calculated in finite steps. Namely, this means, that the probability of change of one's opinion is proportional to the social pressure gradient, which is modeled as the difference of one's opinion compared to one's neighbor in the directed graph of agent relations. This way J of the physical model becomes the probability of change of state P. The proportionality factor K is a scalar for most physical systems and the gradient of the chemical potential gradient is anisotropic for the social system, this way K becomes a coupling matrix with different coupling constants k, where k is larger for agents that are socially closer to others. This is modeled as clusters, where agents in the same cluster have a higher coupling constant inside of the group than to people outside of the group. The gradient of chemical potential that reflects exergonic processes is split into an opinion gradient part and a part that is solely based on the system which reflects processes independent of the social system and is called A. This way the following formula for G in the probability distribution above the agent i G_i is derived:

$$G_i = A + \sum_{i=1} K_{i,j} \left(\frac{\partial \text{ opinion}}{\partial x} \right)$$

A driving force (G) that represents the sum of the model parameters that influence the probability of change are composed of the profit A and the social influence that comes from the network $\mathcal{N}$ is expressed by the following formula:

$$\begin{aligned} G_j &= A + \sum_{i=1, i \neq j}^{N} f_{i,j} + a_{i,j} + d_{i,j} \\ &= A + \sum_{i=1, i \neq j}^{N} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}^{\intercal} \begin{bmatrix} \delta_{f_{i,j}} \\ \delta_{a_{i,j}} \\ \delta_{d_{i,j}} \end{bmatrix} \end{aligned}$$

where α, β, γ are global variables used for fitting the data to the model and $\delta_{f_{i,j}}$, $\delta_{a_{i,j}}$,$\delta_{d_{i,j}}$ are binary variables indicating that the agent i is in a friendship , advisory or discussion partner relation to the agent j.

3.3 Adapting the Model to the Dataset

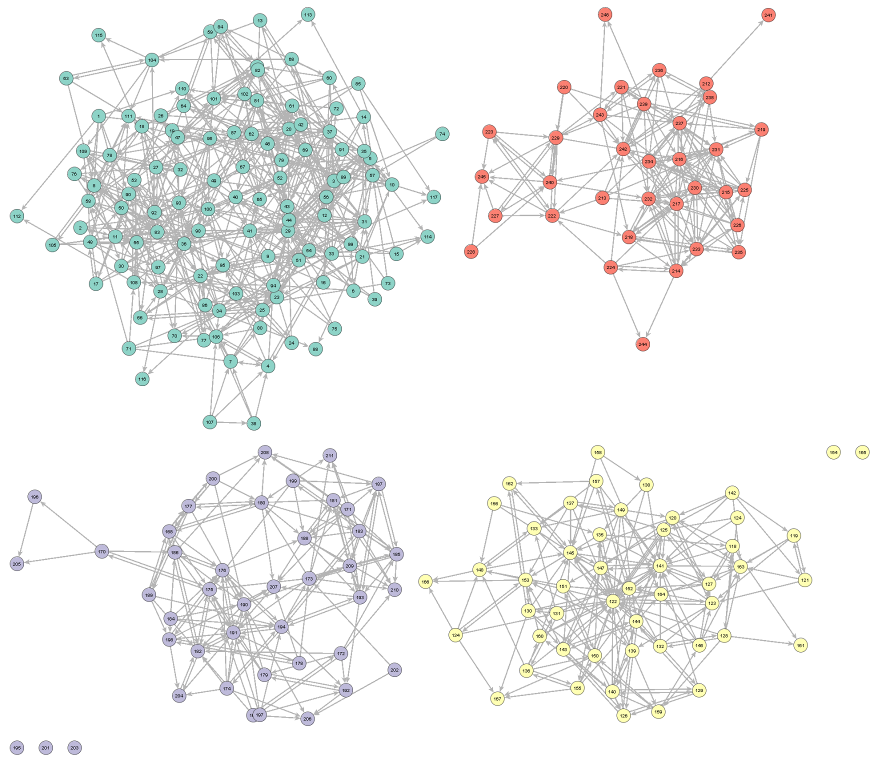

Figure 1: Graph visualizations of physician relations simulated in the Visione software [5] with each cluster representing networks in one of the cities, labeling from top left: Peoria, Galesburg, Quincy, Bloomington

The dataset used to train our model on is taken from Coleman et al. "Diffusion of an Innovation Among Physicians" conducted in 1957 by James Coleman, Elihu Katz, and Herbert Menzel. The study was conducted in four cities: Peoria. Bloomington, Quincy and Galesburg. The relationship data between the examined doctors as agents was collected by surveying physicians in the field relevant to the new drug about their interconnections with the rest of the medical community in their respective cities. The doctors were asked three sociometric questions, namely: to whom they most often turned to for advice and information, with whom they most often would discuss their cases and which colleagues they considered being friends. Each physician could name up to three doctors in each category. It is also important to note that one doctor could stand in multiple types of relations to another. The data was collected on the span of 15 months after the release of the new drug with high potential, which ended up being adopted by everyone. As can be observed in Figure 1 relationship graphs representing each cluster differ in their properties through its connectivity and size. The adoption dates of the physicians were deduced by researchers from pharmacy records of these cities and added to the dataset as well as some personal attributes about the physicians. It is important to note that in this study none of the physicians observed have changed from the adoption of the new drug to using the old medication. Therefore the percentage functions are monotonically increasing.

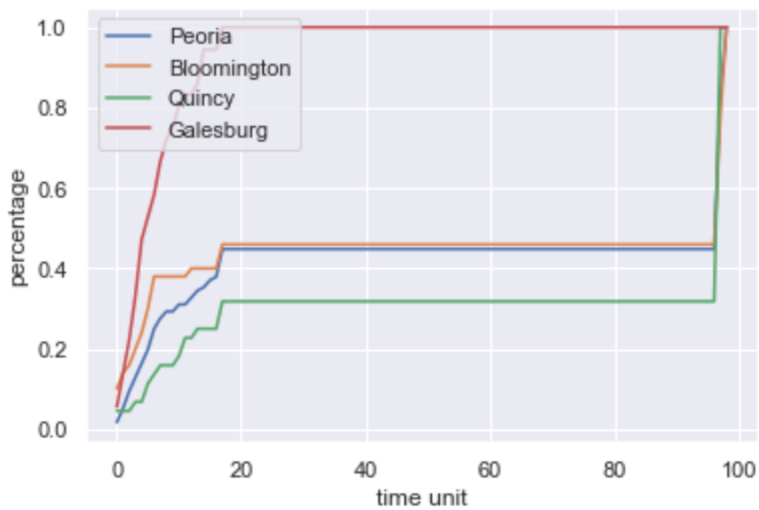

Figure 2: Graph representing time series extracted from the dataset of percentage of physicians who have adopted the new medication per city cluster

In our study, we view the clusters of the physicians and take into account their relations to each other as well as adoption dates to draw our conclusions. We train our model on this data to approximate the importance of each type of relation in opinion formation in a professional field regarding innovation adoption by obtaining the fitted parameters for them from the neural network.

4 Implementation

The git repository of the project can be accessed here: `https://github.com/HX-Tfd/CSIS`

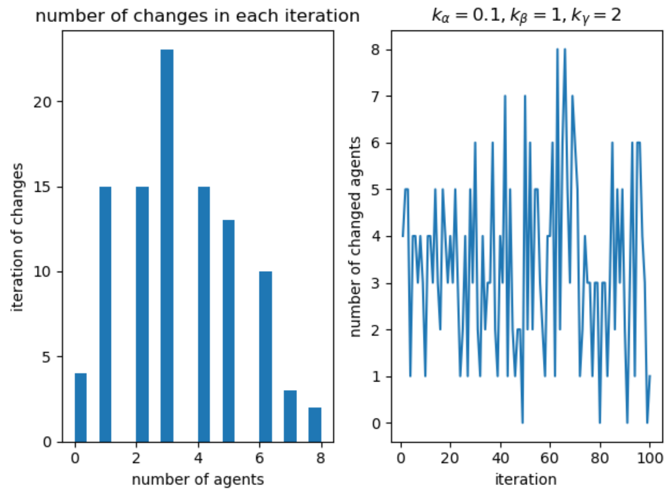

Figure 3: 100 steps of simulation with random data using a Gaussian distribution for parameterising the probability of change

4.1 Random Data

Before assessing the model with real data, we implement the model with the numerical setup described in section 3.2. Specifically, the clusters are created by uniformly partitioning the agents. The driving forces are randomly initialized in the interval $[0, 1]$. The opinions of the agents are randomly initialized as well. We provide the opportunity to experiment with different initialization schemes in our code (see *experiment.py*) and to visualize the simulation dynamically using the graph-tool [3] module (see *visualization.py*). An example plot of the statistics of a random experiment is shown in Figure 3.

To test the simulation with custom parameters, it is possible to modify the default settings in the two scripts mentioned above. Afterward, one can adjust the global configurations and simulation settings in *main.py* and simply run the main script after having installed the dependencies.

4.2 Data-driven Simulation

For the real-world data set, we construct a machine learning model that is supposed to aid us in the prediction of the new opinions adopted by the agents in a cluster throughout the viewed period of time. For that purpose, we need to fit the aforementioned parameters representing the three relational coefficients and the profit variable.

The model is a simple fully-connected neural network with one hidden layer that is twice the size of the input length. The input is a vector that consists of the parameters $(\alpha, \beta, \gamma, A)$ and the driving forces. We use the Sigmoid function defined as $\sigma(x) = \frac{1}{1+e^{-x}}$

for all activations and apply dropout with $p = 0.8$. We use PyTorch [1] for implementing the neural network.

The model is trained with Adam ([2] Kingma et.al - Adaptive Moment Estimation, 2015) with a learning rate of 0.01 throughout the experiments. In each time step t of the simulation, we perform 50 training steps. The first part of the training step involves predicting the probabilities of change with the model with the driving force fixed and using them to predict the new opinions of the agents. Afterward, the loss is calculated as the mean squared error of the opinion differences:

$$\frac{1}{n}\sum_{i=1}^{n}(\tilde{o}_i - o_i)^2$$

where n is is the number of agents in the cluster, since we fit the same model and the global parameters for each cluster individually. Moreover, o_i is the real opinion of the agent i at time step t, and $\tilde{o}_i$ is the corresponding predicted opinion. After the training step, we update the driving forces according to the new opinions.

A potential concern with this training scheme is that it greedily updates the global parameters in the order of the time steps, making use of only the global parameters and the driving forces, thus being exposed to the danger of getting sub optimal results. In the hope to alleviate the issue, we implemented an additional training procedure that first runs the simulation, assembles all the opinions and the driving forces, and then randomly shuffles them for training the model. The input of the new model is additionally conditioned on the opinions. For a fair comparison, we keep the training configurations of the old training scheme. However, we have not observed significant improvements when using the modified scheme. Therefore, we kept the old training scheme for all of our experiments.

Alternative architectures of the neural network are discussed in the future work section.

5 Simulation Results and Discussion

5.1 Experiments

The following eight experiments are conducted. The first four assume that there is no way for the doctors to switch back to the old drug. Therefore the probability of change for a given social system is the following function: P(t, A, α, β, γ). In the above function, t stands for the timestep, I for the profit gained by adopting a new drug, α, β, and γ stand respectively for parameters representing the strength of friendship, advice, and discussion relational ties. These are initialised as $\alpha = 0.1, \beta = 1, \gamma = 2, I = 15$. For the first timestep we do not initialize the driving force but we do so for the second timestep, by using the initial opinions from the given dataset for each cluster and applying our function as described in section 3.2. We run the fitting of the searched parameters separately on each cluster and then having obtained the parameters apply them to the remaining three cities and run our simulation. We hypothesize that the cities that are similarly densely connected would have close values for the fitted parameters and therefore have similar opinion spreading behavior.

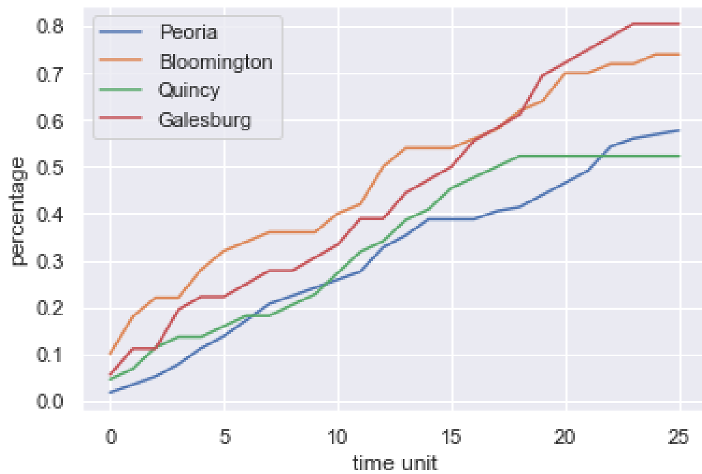

Figure 4: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Peoria

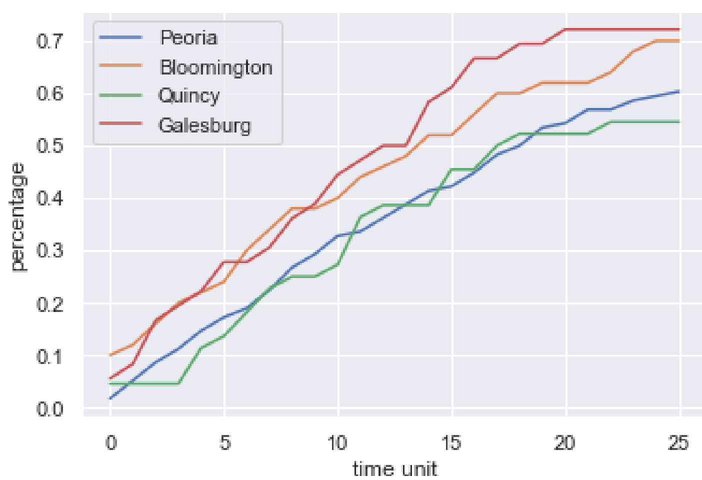

Figure 5: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Bloomington

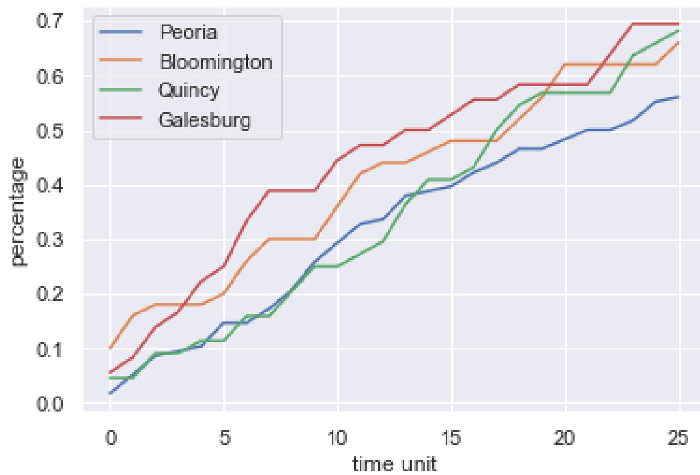

Figure 6: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Quincy

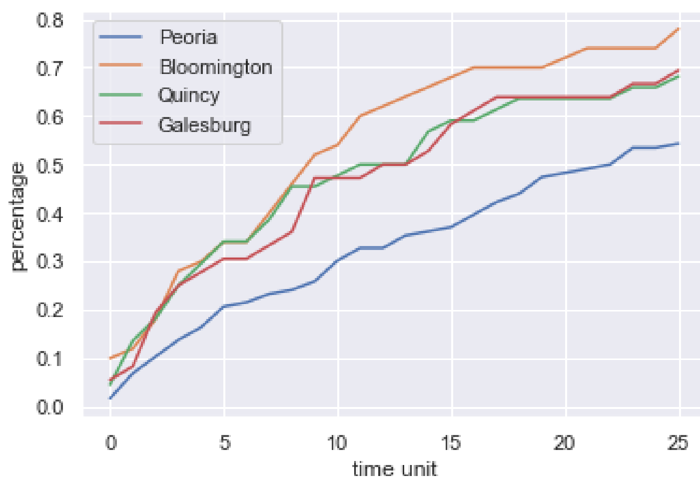

Figure 7: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Galesburg

The next four experiments assume that it would be in principle possible to switch the opinion from the new drug adoption to the older one. Therefore there we use a different opinion function that allows for this functionality. We apply the same procedure as in the first four experiments: calculating the coefficients by application of the machine learning training process on each cluster separately and then simulating the behavior of the agent-based models for the rest of the city clusters with the obtained coefficients.

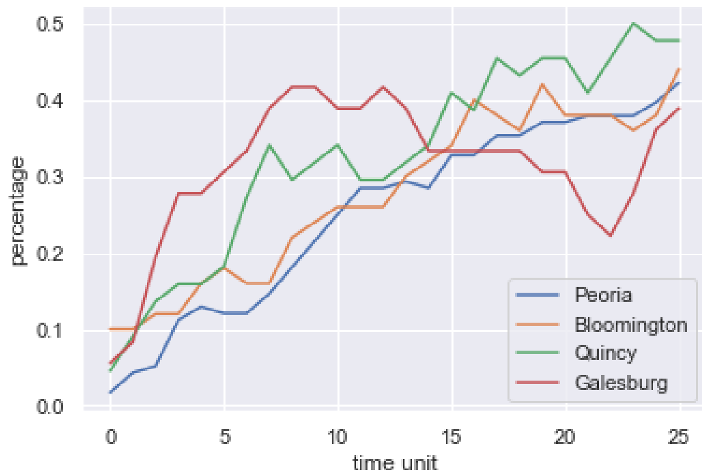

Figure 8: Graph shows the simulated percentage change of physician who adopted a new medication among doctors in its city, when we train the model on Peoria

Figure 9: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Bloomington

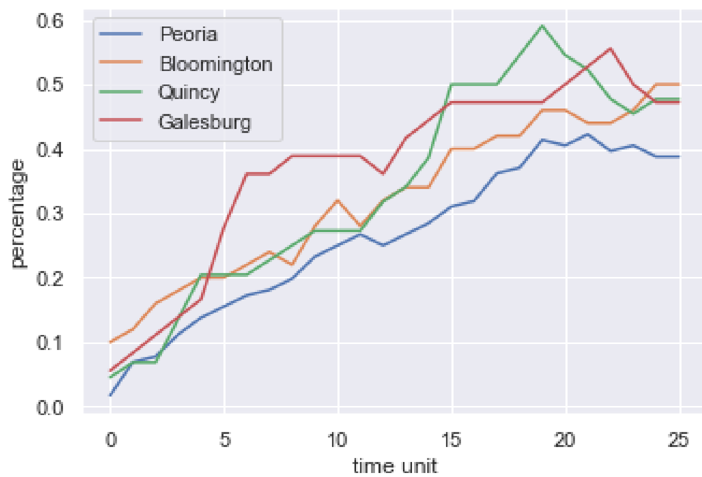

Figure 10: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Quincy

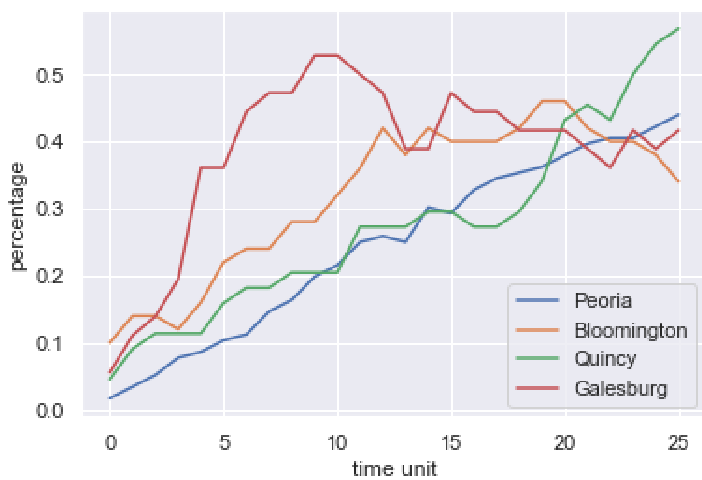

Figure 11: The graph shows the simulated percentage change of physicians who adopted a new medication among doctors in its city when we train the model on Galesburg

5.2 Discussion

First of all, we would like to give the reason why we end up our simulation process after only 25 time units. The reason is that we think that the data set we use has some flaws. As one could easily recognize from Figure 2, the percentage of physicians who have adopted the new medication per city cluster stopped growing after about 20 time units, and rise suddenly at the 98th time unit up to one hundred percent. One could only guess about the reason: May after some time in the city, different opinions about the adoption of new drugs have reached a balanced status, this is similar to the dissemination of new technologies when it has hit a bottleneck. The unusual rise at the end of the time units could be caused by the passage of a new act, which made the use of new drugs mandatory. Another possibility is that influence from outside the social interaction in the city-cluster of the doctors has affected them. Namely, it could be that at the end of the timeframe the doctors that stayed with the old drug until then, are affected by a marketing campaign of a pharmaceutical company advertising and/or incentivizing the doctors to switch their medication scheme. It could also be possible, that the doctors are affected by a study that came out suggesting switching to the drug. All in all the model and dataset with the used methods can not explain the resulting jump at the end of the time period.

We noticed that the experiment with the assumption which doctors could switch back to the old drug simulates more close to true value results. Perhaps this is due to the fact that such a setting is closer to a real-life situation. People's opinions are often wavering, they may try both and then make a choice after. This is an interesting insight because it shows, that including a case in the model that is not even present in the dataset may improve the accuracy of a model.

The uniformity of the distribution of relationships varies from city to city. In Galesburg, the new drug was adopted the fastest. We also noticed that in Galesburg the relationships are more evenly distributed, each agent has a similar number of relationships, whereas other cities have "focus" in the map of the relationship (see Figure 1), i.e. some agents have many connections with others, whereas some agents have few.

A reasonable guess is that in these cities with small groups, opinions spread quickly among the small groups, but because there are fewer ties between each small group or between people on the fringe, opinions spread more slowly throughout the city.

We associate this with a common-sense fact that nepotism is more prevalent in smaller cities than in larger ones, where people have more anonymity and relationships are more evenly distributed.

6 Summary and Outlook

In our paper, we have defined a physical diffusion-inspired model to interpret opinion spreading and have simulated it with random data generated by a Gaussian distribution of opinions over a set of agents. Thereafter we have constructed a neural network architecture and tried through a machine learning process to fit parameters in order to predict the behavior of the system over time. We have run our simulations two-fold, one assuming the agents would only change their opinion once (as has been observed in the real-world dataset) and one with an assumption that agents can change their opinion back to the older drug. We have observed that despite the first assumption being closer to the observations in the provided dataset, the simulation has been performing better with the second assumption. With the first assumption, the tendency has been that the simulated results (no matter what the system has been trained on) show a higher overall percentage of physicians adopting the drug per cluster in comparison to the real data. With the second batch of experiments, the results are much closer in the first 25 time steps to the real obtained data. We also noticed that in the second batch of experiments clusters with similar structures can predict each other's behaviors well. For example, this is the case for Bloomington and Peoria. They both have a big amount of nodes and the graphs are more densely connected than the ones of Quincy and Galesburg. Therefore we see that in Figure 8 at 25th timestep the prediction for Bloomington and in Figure 9 The prediction of Peoria is computed much more accurately than for other cities in these two graphs. When trained on the other two cities the predictions for Bloomington and Peoria have not been as accurate. Therefore we assume that we need a much bigger dataset with more types of clusters to be able to adapt our model to different types of city structures. We have also noted the imperfections of the dataset we have concerning the lack of data collection between the timesteps 25 and 98 since this type of sudden acceptance can only be explained by outer factors as mentioned in Section 5.2 or inconsistencies in data collection.

The current model uses a fully-connected neural network for fitting the parameters and predicting the probabilities of change resp. the new opinions. We found that empirically, setting a higher learning rate will slightly increase the goodness of fit. On the other hand, increasing the number of hidden layers of the network or using other activation functions does not affect the performance much. One possible explanation for this is the expressiveness of the vanilla fully-connected neural network, whose ability to learn useful features from time series is rather limited. A more natural choice will be the use of recurrent neural networks (e.g. GRU, LSTM) or autoregressive models, which have been successful in non-i.i.d data/time series forecasting such as stock markets and weather prediction. Another methodology in the machine learning community that is closely related to the interaction of agents is Reinforcement Learning, which is powerful in modeling the actions (in our case, whether to change the opinion or not) of the agents by learning from their environment (e.g. driving forces) based on some reward function. As our concept is rather new, we have decided to start simple for the design of the network and will leave the exploration of different architectures and methodologies for future works. In addition, we would also like to try different loss functions, that are not merely dependent on the opinions.

References

[1] PyTorch: An open source machine learning framework that accelerates the path from research prototyping to production deployment. *https://pytorch.org/*

[2] Adam: A Method for Stochastic Optimization *https://arxiv.org/abs/1412.6980*

[3] Graph-tool: an efficient Python module for manipulation and statistical analysis of graphs *https://graph-tool.skewed.de/*

[4] "Diffusion of an Innovation among Physicians" , James Coleman, Elihu Katz and Herbert Menzel (1957)

[5] Visione Software: software for the visual creation, transformation, exploration, analysis, and representation of network data *https://visone.ethz.ch/*

Individual Contribution

Han Xi
Implementation of machine learning model, monitoring/managing the codebase (refactoring and adding project description), implementation of the code skeleton for random experiments, implementation of the visualization of experiments on random data using graph-tool; section 4 and latter half of section 6 of the report

Suada Mirgadirova
securing the dataset, extracting relevant data from the graphml file, implementing the refactoring of the data to use it for learning and simulation processes, coding the cluster specific driving force update function, adaptation of probability functions for the real dataset, written sections 1, 3.3, start of section 6.

Daniel Spathelf
Working out Macro and Micro model, Implementing the general model for the given dataset, Testing and Debugging of the Implementation. Introduction and Modelling section of the report.

Likuan Wang
Implementation of the code skeleton for the general process of the project (process.ipynb). Pre-processing of the data set. Adopting the machine learning model and implementation of the simulation process and prediction. Implementation of the visualization with seaborn. Section 5 of the report.

Han Xi

Suada Mirgadirova

Daniel Spathelf

Likuan Wang