Research Group Social Computing
Department of Informatics
Technical University of Munich

Simulation and mining of social networks for quantifying the importance of different features or decisions in the spreading of diseases

Aldi Topalli[1,✉], **Miriam Anschütz**[1,✉*]
[1]Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
✉ {aldi.topalli, m.anschuetz}@tum.de
May 17, 2021

Abstract —
The recent developments in the world, have showed us how important it is to have a deeper knowledge of social interactions, mobility of masses, and the general properties of interactions of people. Coronavirus is one case of infectious disease but it is definitely not the only one. Coronavirus showed to all of us how unprepared we are for a pandemic. Hence more work in this direction is not only important, but also necessary.

1 Introduction

To better understand the workings of the spreading of a virus there are various components needed. First we need a population to observe and to run algorithms on in order for us to understand how the spreading and interactions evolve through a time frame. Then we would need information about how people move in a geographical space in order to complete their daily activities thus exposing themselves to various viruses. Our work consists of a simulation of one component, namely the population and training and tuning machine learning algorithms on the data provided from the aforementioned simulation coupled with simulations from other subgroups in the same project. We will talk about what each of these simulations yields us, and at the same time, the algorithms we have trained with the received data.

2 Related Work

The idea of generating an artificial social network is not a novel one. Nonetheless, it can also be considered novel taking into account the starting point where this generation begins. Most of the work in this domain consists of generation of online networks mainly because of the vast amounts of data available from social media platforms or various online communities. While this is insightful in many fields, it tells little to nothing when it comes to disease spread and properties of the real world interactions of people. It is during these interactions where our interests lays due to the potential of infections and other characteristics of the aforementioned. For this reason we conducted a literature research to try to extrapolate this work on social graphs to offline ones. Talaga and Nowak 2020 talk about properties of real-world social networks such as: high levels of clustering, positive degree assortativity, short average path lengths (small-world property) and right-skewed but rarely power law degree distributions. While they do generate a social network, the size of this network is smaller that would be sufficient to have conclusive observations about various theories, and at the same time the goal of this work is not the network simulation itself, rather to prove the importance of a set of specific properties of social networks. At the same time, the work presented in this paper produces a network that contains only one kind of connection between nodes, which is not fine-grained enough when considering the spread of a virus based on the intensity of human contact.
Bojchevski et al. 2018 work on a similar network generation problem which can be transferred to a social network. They use Generative Adverserial Networks [Goodfellow et al. 2014] to train a neural model so that it produces a similar output to the input data. The main drawback of this approach for our end goal consist of the fact that this approach already requires a social network as input data in order to create another one, which is not available to us.
Li et al. 2018 simulate a social network from scratch by using Deep Learning methods. Their networks' resemblance to real-world counterparts (e.g. molecules) is very realistic. The major problem that

*The work on the generation of the social network was made possible on equal contribution of both authors, the work on the machine learning approach is only from Aldi Topalli

this method has with generating a social network is complexity. The authors use this approach to simulate networks of small to medium size for a significant amount of time. The time needed to generate a network with 100.000 nodes is unreasonable and that is why we decided not to use this method either.
For similar reasons as the ones mentioned above, the work of You et al. 2018 is also unsuitable, specifically for: lack of training data and unfeasable running time for large networks. For the above mentioned reasons we decided to create an original approach to simulating a social network by using classic networks as the ones we describe in the following section.

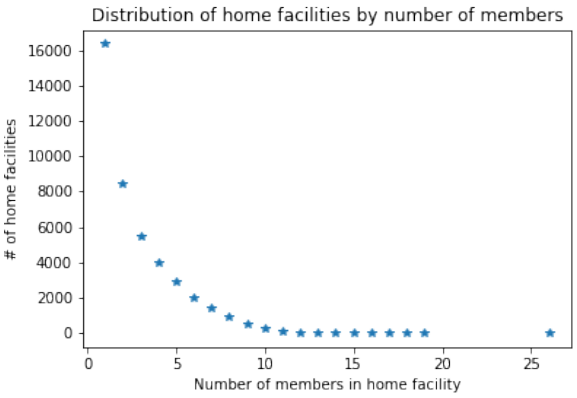

Figure 1 Distribution of home facilities by number of members

3 Generation of social network

In this section, we explain the approach to generate an artificial social network.

3.1 Properties of real-world network

Real-world (social) networks have some distinctive properties that distinguish them from random graphs. These properties are a high clustering coefficient, a power-law-like degree distribution, and a small-world phenomenon present.
The clustering coefficient is the proportion of triangles of connected nodes compared to the number of connected triples. It can be interpreted as a kind of community structure thus real-world networks have clear communities, sometimes even only one large one. In contrast, random graphs are mostly loosely connected hence, they have a rather small clustering coefficient Travers and Milgram 1977.
Another very descriptive property of social networks is power-law like distribution of degrees Newman 2010. In many scientific resources this might be found as the Zipf Law. What this entails is a heavy tailed distribution of connections amongst real-world networks. This is illustrated on the pictures shown on Figure 1, Figure 3 and Figure 5. A power law distribution has a general shape as shown here:

$$f(x) = a \cdot x^{-k}$$

while we haven't explicitly calculated the coefficients a and k we have observed on all types of edges the same shape of distribution: heavy-tailed power-law. This let's us assume that this network according to ibid., chapter 3, resembles a real world network.

The small-world phenomenon describes the observation that the average distance between any two nodes in the network grows sublinearly, i.e. in $O(\log n)$, to the number of nodes. Usually, this is measured with the average shortest path length between all nodes that should be smaller or equal to 6 Travers and Milgram 1977. However, the calculation is in $\mathcal{O}(n^2)$ thus it is infeasible to calculate it for our big network.

3.2 Demographic assumptions

To design a population that is similar to a real population, we need to make demographic assumptions for our model. For this, we have a look at the OECD family database [1] that reports information about family structures and household size. As the mobility data is generated from travel plans in France, we also try to mimic a French population. The average household size in France is 2.38 [2]. A household is defined as people that live in the same house or part of the house and share their costs of living. People in the same household do not necessarily have a relationship other than housemates.
In our model, we also want to mimic family structures. For this, we need to know about the average family size. The OECD reports that a French woman has 1.88 children on average [3]. This means that a family consists of four members on average, two parents and two children. Moreover, usually, there are also grandparents. To compensate for the

[1] oecd.org/els/family/database.htm
[2] http://www.oecd.org/els/family/SF_1_1_Family_size_and_composition.pdf
[3] http://www.oecd.org/els/family/SF_2_1_Fertility_rates.pdf

slightly overestimated number of children (2 instead of 1.88), for families that may not have children yet or for families where some of the grandparents have already died, we only add three instead of four grandparents to the average family. This results in an average family size of 7.

3.3 Suggested approach

We generate a network in which every node represents a person and nodes are connected if the two persons have a long-term relationship. These relationships can be:

- friends
- colleagues
- family
- partner
- housemates

An edge in the network can have multiple labels, i.e. two nodes can be for example friends and colleagues at the same time. We do not allow the connection types family and partner or family and friend at the same time. The family relationship only accounts for family members in a direct line, e.g. parents, grandparents, or siblings but not aunts or cousins. Every person in the network has on average 20 long-term relationships.
In addition, every node has a home facility, i.e. the place where the person lives, and a work facility, which is also used to represent schools and other kinds of facilities where a regular activity takes place. The colleague label is only possible if the connected people work at the same work facility and the housemate label is only possible if the connected people share the same home facility. However, if two people share the same facility, it does not necessarily mean that they have a relation/edge connecting them. With this design, we account for large living blocks or big companies.
We generate the network with nodes and edges from scratch and assign attributes and labels afterward. Therefore, we never manually add or delete an edge to preserve the network properties.

3.3.1 Watts Strogatz model

Probably this should go to related work section
A Watts-Strogatz-model Watts and Strogatz 1998 is a very simple model to generate a random graph with a small world phenomenon. The model generates n nodes and aligns them into a circle where every node is connected to its k nearest neighbours. Then, each edge is reconnected to another node with a probability p. When p is set to 0, the graph preserves its ordered structure while $p = 1$ creates a completely disordered graph. It has been shown that this models builds graphs with real-world properties Newman et al. 2002 while scaling for large networks due to the model's simplicity.

3.3.2 Label propagation to distribute node properties

The nodes in our network have attributes that describe their place of home and work. These attributes should be consistent with the social structure, e.g. it is more likely that a node shares a home facility with a family member than with a random person. To be persistent with this interpretation, we make use of the label propagation algorithm Zhu and Ghahramani 2002 that is based on the assumption that similar nodes should have similar labels. It was originally introduced for semi-supervised approaches and propagates node attributes through the network based on only a few labeled nodes. We receive a list of work and home facilities from the population the mobility group uses but it is also possible to generate completely random places. Each of these predefined facilities is assigned to a random node in the network. With this, we generate our small labeled sample. The algorithm generates a label matrix in the shape of (n_{nodes}, n_{labels}) that represents the label of each node in a one-hot encoding. This matrix is multiplied by the adjacency matrix of the network and normalized until convergence. Finally, we retrieve the optimal label for every node. As the place of work is independent of the home facility, we propagate the two attributes independently.

3.3.3 Resulting algorithm

Putting the network design, the underlying graph model and the propagation of node attributes together, we propose the algorithm as can be seen in figure 2. All parameters can be set in a config file and can be updated when designing a different population.

```
1. Use a Watts-Strogatz model with
   reconnection probability of 0.5
   and average number of connections
   20 to generate a random graph.
2. Assign the node attribute home
   facility
  2.1 Take the given facilities from
      the population file and assign
      them to random nodes in the
      network
  2.2 Propagate the node attributes
      through the network with the
      label propagation algorithm
3. Assign the attribute work facility
    with the algorithm from 2.
4. Label the edges, i.e. iterate over
    the existing edges:
  4.1 Edges connecting people from
      the same home facility are
      housemates with a probability of
       0.5.
  4.2 Edges connecting people from
      the same work facilities are
      colleagues.
  4.3 Randomly label edges as family
      with a probability of 0.35.
  4.4 Randomly label non-family edges
       as partner with a probability
      of 0.05.
  4.5 Randomly label non-family edges
       as friends with a probability
      of 0.15.
  4.6 All edges without a label are
      friends.
```

Figure 2 Algorithm to generate long-term social network

3.4 Evaluation

The resulting graph has 118,820 nodes with 42,459 home facilities and 27,639 education/work facilities.

3.4.1 Network properties

Figure 3 shows the degree distribution of the nodes in the network. As designed, the majority of nodes have 20 connections. The clustering coefficient for this network is 0.09 what is rather high for networks of this size. The network is fully connected and has one giant component. As the Watts-Strogatz-model has a small-world phenomenon present by design Newman et al. 2002, we assume this property for our graph as well. As can be seen in Figures 4 and 5, some of the edge types and node properties show a power-law degree distribution. We can conclude that the generation of a network with real-world properties was successful.

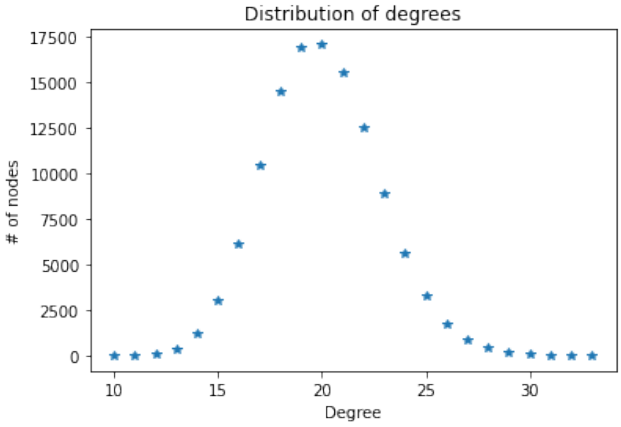

Figure 3 Degree distribution of generated network

3.4.2 Population properties

Figure 4 shows the distribution of number of nodes that are connected with a specific edge label. We can see that most of the nodes have no or only one partner and only 10% of the population have more. This seems to be reasonable as also in real life some people have more than one regular sexual partner. Also the friend and family distribution seems reasonable with friends being the most common relationship to other nodes. The colleague relationship peaks at 1-2 colleagues. It is important to mention that in our design, colleagues are employees at the same facility that the person has a strong relationship with and not only any person that works at the same place. When discussing the colleague relationship, it is also interesting to look at the size of the work facilities. Figure 5 shows the distribution

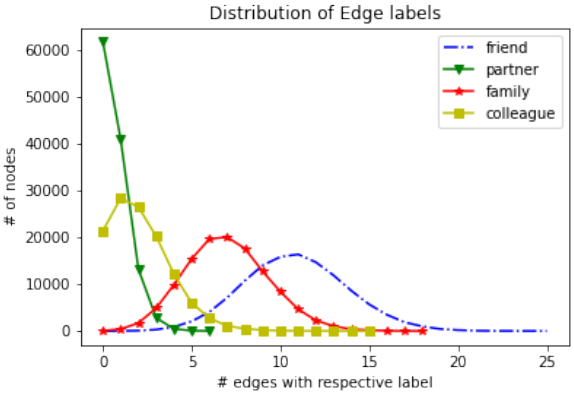

Figure 4 Degree distribution of different edge labels

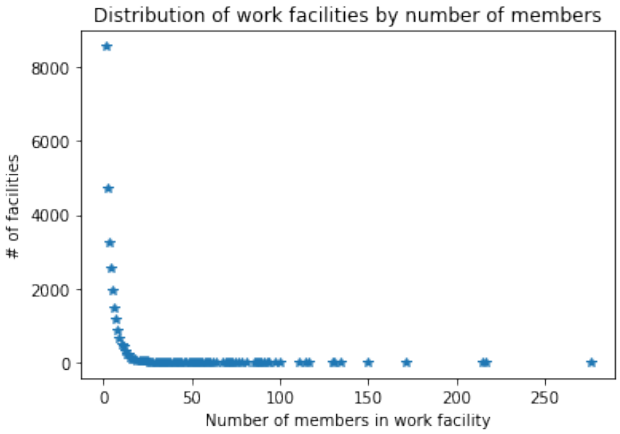

Figure 5 Distribution of work facilities by number of members

of work facilities by their number of employees. We can see that there are many facilities with less than 10 employees and a few facilities with more than 200 workers. This seems to mimic work in real life where most of the people work in small or medium sized companies and there are some very big global players.

4 Other Parts of the Simulation

Generation of the social network was only one part of a larger simulation. and Lopez 2021 et al. worked on simulation of mobility based on the graph that was generated by us, and Yee 2021 developed their work further to simulate disease spreading using a disease model. It is important for us to clarify the whole simulation process because the data from all stages will be used for the next section, namely Machine Learning. To summarise the simulation consisted of three main parts: 1. Generation of an offline social network, 2. Simulation of Mobility of this social network, 3. Simulation of infection spreading among this population during their daily activities.

5 Action Based Risk Prediction

The Machine Learning approach for risk prediction is based on actions. Everyday we do a set of actions in order to complete tasks or for other objectives. This actions usually look like this:

$$Home- > Bus- > Work- > Bus- > Supermarket- > Home \quad (1)$$

In this representation of daily actions each arrow represents an action which has attributes like time, number of people around, facility, the id of the person doing this action and so on. All these attributes are pulled from the simulation which is mentioned on the previous stage. The first step towards classifying actions and/or their temporal and spatial properties in terms of risk of infection is to define a method with which to quantify this risk by observing infections on a 'god-like' mode that the simulation gives us.

5.1 Word2vec

5.1.1 Resemblence to word2vec

Word2vec Mikolov et al. 2013 is an algorithm which given text data produces dense vectors for tokens extracted from this text. Tokens can be words or even sub-word chunks for which, by using word2vec, we get a representation of that token on a high-dimensional semantic space on which we can do various machine learning approaches for many applications. The intuition behind this method is that it captures a semantic relation between words, and eventually among larger blocks of text such as sentences paragraphs and so on, e.g. probably the most popular illustration of this idea is that in this vector space if we perform the following operation:

$$w_{queen} \approx w_{king} - w_{man} + w_{woman}$$

where w represents a dense word vector.
Enumerating actions of a day again it is clear to see the resemblance with a text sentence:

$$Home- > Bus- > Work- > Bus- > Supermarket- > Home \quad (2)$$

the question we try to answer here is how to represent an action in order to get a vector representation from it. By getting dense vectors for this concepts we should be able to cluster risky locations and actions so that decision makers can take more meaningful and effective measures.

5.1.2 Word2Vec details

To get to these word vectors the basic idea is simple: train a neural network to predict a word, given it's context. The two main approaches used for training word2vec are: *Continous bag of words model* and *Continous skip-gram model*. For our approach we will use skip gram as Meyer et al. 2018 did. The objective of the skip-gram training algorithm is to learn word embeddings that can be used to predict a word's surrounding words in a sentence. In

this case two vector representations, an input embedding and an output embedding, are learned for each word. Input is the embedding that is generated when a word is used to predict the context and output is the embedding that is generated when the context is used to predict the word. The final embeddings that are used for the words are the input embeddings. To obtain this embeddings we need to maximise the following function:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-k \leq j \leq k, j \neq 0} \log p(w_{t+j}|w_t;\theta)$$

where w_1, w_2 ... is the sequence of words in the training corpus, T is the length of the sequence, k specifies the training window size, i.e. the context. and the probability $p(...)$ represents the probability of word w_{t+j} occurring in the context of word w_t. Parameters θ are then used as word embeddings. Maximizing the above function means minimizing the following function:

$$C(\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-k \leq j \leq k, j \neq 0} L(w_t, w_{t+j};\theta)$$

where $L(w_t, w_{t+j};\theta)$ is the loss when trying to predict the sourrounding word w_t.
The skip-gram model trades complexity for computational efficiency by not using non-linear layers, usually very common among other neural language models. The probability of word w_{t+j} given word w_t and parameters θ is computed using the following formula:

$$p(w_{t+j}|w_t) = \frac{\exp(v\prime_{w_{t+j}}^{\intercal} v_{wt})}{\sum_{i=1}^{T} \exp(v_i\prime^{\intercal} v_{wt})}$$

where v_w and v'_w are the input and output vector representations for w, $v^{\intercal}$ is the transpose of vector v and V is the number of words in the vocabulary.

5.2 Generation of data

5.2.1 From xml to tabular format

The simulation from the previous stage has given us data about 3 parts: the social graph, the mobility and the disease evolution among this population. The data we will be using for this machine learning approach is mainly from the mobility part Lopez 2021. This simulation has only 6 types of events, which means 2 types of actions, and unfortunately are too general for any comprehensive result:

- actstart
- actend
- PersonEntersVehicle
- PersonLeavesVehicle
- episimContact
- episimAlmostContact

actstart represents the beggining of an action while *actend* represents end of it. Equally the *PersonEntersVehicle* represents the beginning of commute from one place to another, and *PersonLeavesVehicle* marks end of this commute. The aforementioned do not intersect with eachother, which means that *PersonEntersVehicle* cannot occur between an *actstart* and an *actend*, and the same applies for *actstart* and *actend*, they cannot occur between *PersonEntersVehicle* and *PersonLeavesVehicle*. At the same time, the order is also start before end, and of course, you cannot leave a vehicle without first entering it. The mobility simulation simulates only one day, for 108246 people, and the number of actions per day per person can vary from 1 to 38, meaning that at max a person does 38 actions per day. From this data, aggregating an action, i.e. *act* or *commute*, into one atomic action, means to read everything between start and end, i.e. *actstart* and *actend*, and add the information as attributes of one action. In the end the generated data has the following attributes per action:

- id (int)[person id]
- datetime (string) [timestamp]
- concept (string) [location]
- duration (int)[seconds]
- groupSize (int)
- contact (bool)

In total there are 558860 actions. As a starting step, as concept we have decided to keep the location, this gives us a vocabulary of rougly 211.000 concepts.

5.2.2 From 2 actions to 211.000

As also shown above 2 actions are not sufficient to do an analysis of action risk, or location and action based risk prediction. For this reason we need to explore different ways to get more fine-grained actions from these very general ones. To do this, as a concept we use the name of the location, and for the computation of similarity, which is used on the training of the embeddings phase, we use action data such as: start of the action, duration and group size.

5.2.3 Preparing the train file

The difference from *word2vec* algorithm is that instead of using the same weight for all words in the context, here for each concept in the context we use a weight which is computed based on certain attributes extracted from the simulation. First for each concept in the datafile we compute the following attributes:

- time difference
- duration difference
- group size difference

where each one represents the difference between the respective attributes between two actions, e.g. time difference is the difference in seconds of the starting time. After this, the numbers are scaled with weights from 0 to 1 for all three using an exponential decay function:

$$f(x) = r \cdot e^{-k \cdot x}$$

This formula has the properties we desire because for a large difference between the attributes, i.e. value of x in the equation, the value of $f(x)$ is 0, and for a small value of difference, they get a high weight, r. The values of r and k are:

	r	k
Time difference	1	0.0009
Duration difference	1	0.0009
Group size difference	1	0.09

5.3 Training embeddings using Time2Vec

Time2Vec is an algorithm, an extension to word2vec, which Meyer et al. 2018 use to train embeddings on medical temporal data. The data consists of diagnosis of patients with various diseases through time, and the initial idea is to get vector representations for these diseases based on their temporal relation.
Using the scaled weights mentioned in the previous sub section, by taking a simple average we arrive to a single weight for the similarity between two concepts. This weight is then used for a modified cost function:

$$C(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{c_j \in S_t} w_{tj} \cdot L(c_t, c_j)$$

where w is the scaled weight. The concept embeddings are then trained using this cost function.

5.4 Embedding space

The embeddings are a matrix $M \in \mathbb{R}^{n \times 50}$ on which each row is a vector representing a location that is compared with the other ones based on actions that happen there. In order to get some insights into these embeddings, we do a down-smapling and later visualization of the multi dimensional space by using t-distributed stochastic neighbor embedding (t-SNE).

5.4.1 Interpretation

As can be seen from the figure, there are no clear clusters in the embeddings. We believe that the main cause for this is the concepts themselves. The ones we use as actions are locations, and as described above, the measures that are used to weight their co-occurrence is mainly: group size, start time and duration. Which for example in one location could possibly take all values for various people, which than leads to in-difference in the exact value of these properties, and hence, no patterns to learn from it. As it will be discussed on the future work section, this highlights the fact that a good fine-grained action simulation, coupled with the mobility simulation is necessary. A facility alone is not sufficient to determine risk, based only on these three properties provided so for a better prediction, we arrive to the conclusion that the simulation should, from the beginning include many types of actions that could be done on these facilities.

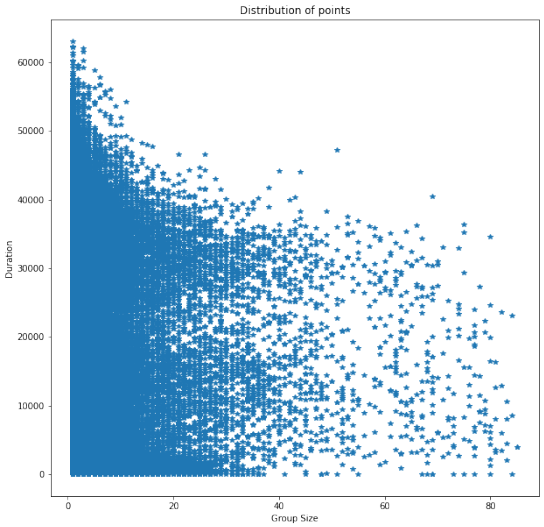

Figure 6 Distribution of (group size, duration) points

5.5 Alternative approach towards risk prediction

Since embeddings from the previous stage did not give any predictive capabilities in terms of infections based on actions done, we decided to take an additional approach to give some kind of closure to the action based risk prediction. The approach we take here is based on a simple heuristic which we use to distribute risk based on the aforementioned properties of each action. Namely, we start by first scaling the time difference, duration difference and group size difference. On Figure 6 you can see the distribution of the points we will be working with, the x-axis represents the group size, y-axis represents the duration and each point is an action.

5.5.1 Distribution of risk

The distribution begins by first normalizing both measures to the same scale. Since at a subsequent step we will be normalizing for each agent to a probability distribution, we chose to only scale the duration to the same scale as the group size range. For this we use a standard formula:

$$d_i^{scaled} = \frac{d_i - \min_j(d_j)}{\max_j(d_j) - \min_j(d_j)} * (\max_j(gs_j) - \min_j(gs_j)) + \min_j(gs_j) \quad (3)$$

where d stands for *duration*, and gs stands for *group size*. The equation 3 will scale duration to the same scale as group size. After this scale, we construct the timelines for all people involved in the simulation. By timeline here we mean all actions that we can collect for that person for the past 14 days, since the simulation on which we are running this experiment is repeating the same day multiple times, we safely consider only the past day. The reason for this safely consideration of only the last 24h is because since it is the same day, for all risks on all actions we would be simply multiplying by the number of the days, e.g. 14 and then dividing by the same number 14, thus giving us the same risk for that action for one day. For each person i we construct a $timeline$, which is a set containing all the actions person i did in the past 24h, in our case. To compute the risk for each of these actions per this person, we simply add the duration and group size for that action in that specific timeline:

$$action_{ij} = duration_{ij} + group\_size_{ij}$$

where i is the person and j is the j-th action that person does. So the duration and group size are specific to this combination. After this stage, we make a probability distribution from this distribution within each timeline:

$$action^{'}_{ij} = softmax^{i}_{j} action_{ij}$$

where we transform each action relative to it's timeline tied to one person, i in the above formula. Once it is normalized in the range 0-1 we now distribute the risks to the concept for all the people. For example for concept j the risk would be computed across all people that did that action, let's call this set X:

$$risk_j = \sum_{i \in X} action^{'}_{ij}$$

The final step would be to go back to the timeline, and pick the riskiest action for all the people, according to the distribution described above. This method yields roughly 40% accuracy, meaning that from the picked actions, 40% of them actually infected someone. We stop at the accuracy since the f1-score and other metrics would give a much grimmer view since the dataset is largely unbalanced, and we back this decision with two real world facts:

- False positives are cheap (corona-tests are free in many countries now)
- Unsupervised methods as the one above, can rely on accuracy as a metric

6 Ethical Considerations

Nothing vast enters the life of mortals without a curse.
Sophocles

Machine learning has disrupted many fields of humanity by introducing new and better ways of doing things, but this did not come only with benefits. As everything in the world, this technology is also a tradeoff. The fuel on which Machine Learning runs is data, and the collection and use of data more times than often stands on the borderline of ethical and legal considerations. It might be legal to scrape a website and get public data from it, but is it ethical to use the aforementioned data to profile users? This question and many other ones similar to it, are what should guide a machine learning practitioner when she or he decide to build a model that is used for sensitive purposes.
The prediction of infection risk includes a lot of these considerations since it involves data such as a social network, which contains information about people's social interactions and relationship, mobility data, which tells us how people move in a geographical area, and the disease data which tells us how the infection moves from one person to the next. Discussing all of these aspects of the data is not a small endeavour, for this reason we will focus on this section on the ethical considerations that come with the action and location based risk prediction. Predicting the risk of infection for people based on their temporal and spatial data, first of all, is not an easy technical problem, nonetheless, it is even harder to stay on the ethical side, especially when the times are so challenging as the recent time of COVID-19 pandemic. The urgency and the desperation, both of the masses and the decision makers, might push people to ignore these consideration for an efficient and fast solution. On this paragraph we aim to show the main risks posed by our approach, and possible ways this might cross ethical borders.

6.1 Marginalization and risk of discrimination

Training the embeddings based on action and location data, gives us a way to determine which locations and possibly at which times, are dangerous for the spread of a virus. As expected, most probably these places usually consist of locations that involve a lot of people in a small area, which means that most likely it is a low income area. The same way marginalization based on actions would mean the the actions with a high risk of spreading the virus are the ones who exchange with a lot of people in their daily routine, such as: supermarket staff, waiters etc.. Taking measures on this data would mean punishing one segment of the society much more than the rest, as it was shown to be the case during the latest pandemic in many places in the world. This sort of marginalization would contribute to an increasing inequality gap in the society by making it hard to nearly impossible for so many people to get on with their daily lives, while at the same time affecting very little to not at all the higher end of the society. This could lead to problems that have come with large divisions in the society, time and time again throughout history, such as violent conflicts, rise of crime with severe consequences. A very clear example of this through the COVID-19 pandemic was the city of New York in the United States, where poor burrows of the city, e.g. Bronx, had a much harder time with the pandemic, with authorities reporting a higher number of unemployed individuals, and also a spike in armed clashes between different groups. By taking measures on an absolute level, only taking into consideration the following results from the data, these are risks to any society. Thus the measures should be accompanied with additional ones to counteract this extending gap between the social groups, such as making sure that the financial burden is the same for all. If people can continue to work and be profitable during this time, they should contribute slightly more than the segment of the society that is forced to stop their financial activity. At the same time, relief efforts should be distributed with this inequality in mind. Instead of equally distributing these supplies, health and financial urgency should be the decider on who gets access to these supplies. Of course this might raise a series of problems, and it needs to be planned carefully, but implementation of measures should always take into account the fact that not everyone is going through the same pandemic.

6.2 Extremism risks

Besides the risks from marginalization, taking measures based on action and location data, can also lead to a rise in political extremism. The risk stems from the fact that, same as mentioned above, most likely the groups which will result in a high risk one

from the algorithm will be low-income, high contact one. It is very common that these roles are done from immigrants which form minorities in most of the developed cities throughout the world. This would mean that pointing a finger at them would lead to various interpretation from far-right activists who are against immigration policies and sometimes manifest their views in a violent way. Again an example can be drawn in United States of America, where due to the against Asian rhetoric during the 2019 pandemic, many Asian-Americans experienced a series of violent attacks which unfortunately sometimes resulted in casualties. In a world where the far right is becoming ever more powerful, these are problems that the same way as the marginalization ones, should be addressed in the measures that are to be taken to fight the pandemic.

6.3 Political risks

Another risk that needs to be considered is that these tools might serve as a way for countries and democracies to slip into authoritarianism. It should be taken into account that the tools will be used by a party which is currently in power where ever they are to be deployed, and this would mean that this political party, or leader, would have unrestricted access into voter data which could lead to them more easily manipulating people into their own world-view and thus leading to a weaker opposition, crucial for a healthy democracy. Again this has been viewed in many countries in the world during the pandemic, developed ones with strong democratic institutions and also developing ones with quasi-democratic regimes. In both cases the situation tilted more towards a more centralised regime, where the party in power had in many cases absolute control.

6.4 Ethical conclusion

The above mentioned risks are all very important for fighting the pandemic, and also to get back to a "normal" life, once a way to fight "the current virus" is found. If we still want to live in an ever developing world when the pandemic is gone, the above mentioned issues need to be taken into account so that problems and other crises, possibly more dangerous, do not emerge from this extraordinary situation. The measures should ideally be implemented with a clear reverse scenario, and transparent anti-measures for when the threat is dealt with.

7 Conclusion and Future work

During the last months we have worked on developing and testing approaches to assign risk to a given action with regard to disease spreading. The first approach, concept embeddings, even though it performed poorly in our experiments, ought to be a very promising one, given the right data. I believe that the reason why this did not perform as expected lays in the objective function, we try to discriminate by taking into account two features, group size and duration, but the problem is that these features take the whole range of values for all of our concepts, thus rendering a discrimination on them useless. That is exactly what we have observed. The simple observation would lead us to believe that the simulation does not yield data good enough for this kind of approach, and that the same approach with more fine-grained information about the actions, such as contact intensity, physical interaction etc., would most likely deliver the wished for results. The second approach consisted of a simple heuristic where we simply try to distribute the risk across concepts. This gave us a modest accuracy into prediction of infections, but still not sufficient for the end goal of predicting infections and accurate risk numbers.

7.1 Future Work

In order for better results on all approaches, a fundamental change is needed on our approach: the data. As also mentioned above, to understand the results we need to only look at the objective function of the methods we trained. The minimization of the error of prediction during training, tries to discriminate on features where there is little to nothing to discriminate on: group size and duration. In order to tackle this problem, and to have a better prediction model, a new mobility simulation is needed, one which simulates not only high level movements of people, but also gives details into the actions so that the algorithm can learn to differentiate between close and intense, hence risky, actions, and distant and cold, non-risky, ones.

7.2 Acknowledgments

This is a personal paragraph unrelated to the technical details. I wish to thank all those involved in the project, for their assistance and patience. In particular: Professor Georg Groh, M.Sc. Edoardo Mosca,

M.Sc. Tobias Eder and Miriam Anschütz(order arbitrary) for assisting us in their work. It would not have been possible to go on without your help.

References

Bojchevski, Aleksandar, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann (2018). *NetGAN: Generating Graphs via Random Walks*. arXiv: 1803.00816 [stat.ML].

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].

Li, Yujia, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia (2018). *Learning Deep Generative Models of Graphs*. arXiv: 1803.03324 [cs.LG].

Lopez, Maria (2021). "Ethical AI for Pandemic Control". In:

Meyer, Francois, Brink van der Merwe, and Dirko Coetsee (2018). "Learning Concept Embeddings from Temporal Data". In: 24.10. |http://www.jucs.org/jucs$_{24}1_0/learning_{c}oncept_{e}mbeddings_{f}rom|$, pp. 1378–1402.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Newman, M. E. J. (2010). *Networks: an introduction*. Oxford; New York: Oxford University Press. ISBN: 9780199206650 0199206651. URL: http://www.amazon.com/Networks-An-Introduction-Mark-Newman/dp/0199206651/ref=sr_1_5?ie=UTF8&qid=1352896678&sr=8-5&keywords=complex+networks.

Newman, M. E. J., D. J. Watts, and S. H. Strogatz (2002). "Random graph models of social networks". In: *Proceedings of the National Academy of Sciences* 99.suppl 1, pp. 2566–2572. ISSN: 0027-8424. DOI: 10.1073/pnas.012582999. eprint: https://www.pnas.org/content/99/suppl_1/2566.full.pdf. URL: https://www.pnas.org/content/99/suppl_1/2566.

Talaga, Szymon and Andrzej Nowak (2020). "Homophily as a Process Generating Social Networks: Insights from Social Distance Attachment Model". In: *Journal of Artificial Societies and Social Simulation* 23.2. ISSN: 1460-7425. DOI: 10.18564/jasss.4252. URL: http://dx.doi.org/10.18564/jasss.4252.

Travers, Jeffrey and Stanley Milgram (1977). "An experimental study of the small world problem". In: *Social Networks*. Elsevier, pp. 179–197.

Watts, Duncan J and Steven H Strogatz (1998). "Collective dynamics of 'small-world'networks". In: *nature* 393.6684, pp. 440–442.

Yee, Soh (2021). "Ethical AI for Pandemic Management". In:

You, Jiaxuan, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec (2018). *GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models*. arXiv: 1802.08773 [cs.LG].

Zhu, Xiaojin and Zoubin Ghahramani (2002). "Learning from labeled and unlabeled data with label propagation". In: