

Corona Contact Tracing

TODO

March 21, 2020

Abstract

This technical report accompanies the implementation of the `corona_contact_tracing` package found at https://github.com/PellelNitram/corona_contact_tracing.

It extends an existing stochastic SIR simulation to generate data sets suitable to simulating the current corona pandemic. This simulation has full access to all agents' health status. To mirror a real world scenario the health status of agents is only partially observed through tests.

Next we develop a graph convolutional approach to predict the health status of all agents in this simulation. It uses past contacts as well as observed health information to derive this prediction. It is able to deal with partial data such as missing locations and missing health status, as it is a probabilistic approach.

Lastly, this prediction can be used to derive measures to reduce the diseases' spread. We propose to find a optimal trade of between removing the least amount of edges in said graph (e.g. through quarantine, social distancing, etc.) and limiting the spread of the disease. Opposed to classical non-pharmaceutical intervention methods such as contact tracing, our approach directly identifies the nodes with the greatest potential to accelerate the diseases' spread in the network.

This technical report was created within the #WirVsVirus Hackathon of the German government and is Work in Progress!

1 Algorithm notes

1.1 SIR Model

1.2 Probabilistic Health Prediction

We think of partially infected population as a graph, where each individual (or agent) is a node. Edges of this graph model contacts of two agents. As the health state of the entire population is unknown, we use graph convolution, as explained below, to update our assumptions on the current health state of all individuals.

The definition of a graph convolution for here is

$$h_{v_i}^{(l+1)} = \sum_{j \in A(i)} h_{v_j}^{(l)} \quad (1)$$

with feature vector h of node i from iteration (l) to $(l+1)$. This formulation is equivalent to $h^{(l+1)} = Ah^{(l)}$ with A as adjacency matrix without diagonal elements as shown in section 2.1.

Our main propagation rule is based on the definition of a graph convolution and reads component wise as follows

$$h_{v_i, m}^{(l+1)} = \underbrace{\sum_k \frac{\hat{A}_{v_i, k}^{(l)}}{\sum_j \hat{A}_{v_i, j}^{(l)}} h_{k, m}^{(l)}}_{\text{Graph}} \delta_{m, e_I} + \underbrace{(h_{v_i}^{(l)} \cdot T)_m}_{\text{Temporal}} \quad (2)$$

This propagation rule is based on the following aspects:

- give setting about graph and temporal parts.
- only I-component to be included in graph part as shown in blue.
- \hat{A} is constructed from A and I which are the regular continuous adjacency matrix and the infection matrix, respectively.
 - The adjacency matrix A is time dependent, $A^{(l)}$, and inferred from data. In our use case, $A_{ij} \propto \text{dist}(v_i, v_j)$, hence A_{ij} is large when persons i and j have been in contact.
 - The infection matrix is constructed as

Think about that!

$$I = \begin{pmatrix} 0 & 0 & 0 \\ \beta & 0 & \alpha \\ 0 & 0 & 0 \end{pmatrix} \quad (3)$$

with I_{ij} and i is the index of the host state and j is the index of the contact person state. The states that we consider here are ordered as follows: susceptible, infected, recovered. β denotes the probability of infection after contact (also known as attack rate). $\alpha := 0$ models the probability of being reinfected, which we assume to be zero based upon medical research.

- \hat{A} , with $\hat{A}_{ij} \in [0, 1]$, is the adjacency matrix that takes the infection interactions into account and is computed as follows

$$\hat{A}_{ij} = A_{ij} \cdot \frac{v_1^T I v_2 + v_2^T I v_1}{\beta}. \quad (4)$$

The sum comes from the fact that both, agent i and j , can act as host during a contact. The division by β normalises the factor to one to ensure $\hat{A}_{ij} \in [0, 1]$. Since I is not symmetric, p_a is a proper normalization because the sum is in $\{0, p_a\}$. Note that the fraction has the desired properties for pure S -, I - and R -persons.

- The feature matrix, $h^{(l)}$, consists of all agents' features at time l and is thereby of dimension $N \times D$ where there are N agents in the population and each agent is described by D features. A three dimensional feature space is used, $D = 3$, modeling the three possible health states. The unit vectors of this space are interpreted as following:

- \vec{e}_0 : susceptible state
- \vec{e}_1 : infected state
- \vec{e}_2 : recovered state

A uniform distribution over these possible states expresses complete uncertainty of the health state of an agent.

- The transition of a persons' health state $h^{(l)}$ is determined by the following assumptions:
 - A susceptible person always stays susceptible (unless there is a vaccine, which we do not model)
 - A infected person has a probability γ called recovery rate to recover. The remaining probability $1 - \gamma$ means that the person stays sick.
 - A recovered person could have a probability to be re-infected, but we assume this to be zero. Thus a recovered person always stays recovered.

Thus the transition matrix T is:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 - \gamma & \gamma \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

The temporal update rule based on the health status thus becomes:

$$H^{(t+1)} = H^{(t)}T \quad (6)$$

1.3 Disease Spread Reduction with Minimal Intervention

This section presents approaches to optimally reduce the spread of the disease with minimal intervention.

1.3.1 Edge Cutting Analysis

All non pharmaceutical interventions (NPI) can be understood as some kind of edge removal in our graph-based approach:

- Isolation of an infected individual removes all of its edges with very high probability.
- Quarantine of a contact person removes all of its edges with high probability.

- Social distancing removes some edges of many individuals.
- Cancellation of large events remove many edges of many individuals.

As a first step analyze the effect of these measures in R_0 . R_0 models how many people are infected by a single infected person. It is crucial to constrain R_0 below zero to avoid exponential growth in the number of infected individuals.

As future work we would like to explore ways to calculate the minimal set of edge removals which reduces the R_0 value of the pandemic below 1.

In our approach R_0 can be derived from the ratio of infected people of $H^{(t+1)}$ and of $H^{(t)}$:

$$R_0 = \frac{\|H^{(t+1)}\|_0}{\|H^{(t)}\|_0} \quad (7)$$

The square matrix C with dimensions $N \times N$ models desirable edge cancellations. Note, that this matrix does not have to know the edges of a future time step, it only expresses which edges must not exist. It is multiplied element-wise onto the adjacency matrix A , thus $\bar{A} = A \odot C$ describes a adjacency matrix with applied cancellations.

To optimally limit the spread of the disease, we seek to minimize the number of cancellations $\|C\|_0$, given that R_0 is below zero:

$$\max C \|C\|_0, \text{ s.t. } R_0 < 1 \quad (8)$$

TODO: Derive an expanded statement from the above formulation.

1.3.2 Test Prioritization

When tests are limited, they should be used to discover as much as possible about the health state of the overall population. This in turn allows to reduce the R_0 value in further time steps as edge cutting is more efficient.

Lets assume there are t_{\max} tests per time step. A test reveals the true health state of an individual (ignoring false negatives and false positives)

$$h^{(t)} \xrightarrow{\text{test}} h^{(t+1)} \in \{\vec{e}_0, \vec{e}_1, \vec{e}_2\} \quad (9)$$

The test assignment T with dimension N is a binary variable describing which individuals should be tested.

2 Appendix

2.1 Consistency of notations

Using the notation from the blog and the paper, including $H' \in \mathbb{R}^{N \times D}$, $A \in \mathbb{R}^{N \times N}$, $H \in \mathbb{R}^{N \times D}$, the propagation rule is:

$$H' = AH \quad (10)$$

is equivalent to

$$\begin{aligned} H'_{v_i,m} &= \sum_l \overbrace{A_{v_i,l}}^{\in \{0,1\}} H_{l,m} \\ &= \sum_{l'} H_{l',m} \end{aligned} \tag{11}$$

where l' takes all $A_{v_i,l} = 1$, i.e. neighbours, into account.
Hence, I do understand that the notations are equal.

2.2 Formulate normalisation

Use from ICLR 2017 paper the normalisation $D_{ii} = \sum_j \hat{A}_{ij}$.

Use the knowledge from the blog post to write:

$$\begin{aligned} \underbrace{\in \mathbb{R}^{N \times D}}_{H'} &= \underbrace{\in \mathbb{R}^{N \times N}}_{D^{-1}} \underbrace{\in \mathbb{R}^{N \times N}}_{\hat{A}} \underbrace{\in \mathbb{R}^{N \times D}}_{H} \Leftrightarrow H'_{v_i,m} = \sum_k \overbrace{D_{v_i,k}^{-1}}^{=D_{v_i,v_i}, \delta_{v_i,k}} (\hat{A}H)_{km} \\ &= D_{v_i,v_i}^{-1} (\hat{A}H)_{v_i,m} = D_{v_i,v_i}^{-1} \sum_k \hat{A}_{v_i,k} H_{k,m} = \sum_k \frac{\hat{A}_{v_i,k}}{\underbrace{\sum_j \hat{A}_{v_i,j}}_{\text{normalised}}} H_{k,m}. \end{aligned} \tag{12}$$

The weighting by the adjacency matrix is, indeed, normalised to its column sums.

Note that this part is also normalised:

$$\begin{aligned} \sum_m H'_{v_i,m} &= \sum_m \sum_k \frac{\hat{A}_{v_i,k}}{\sum_j \hat{A}_{v_i,j}} H_{k,m} \\ &= \sum_k \frac{\hat{A}_{v_i,k}}{\sum_j \hat{A}_{v_i,j}} \underbrace{\sum_m H_{k,m}}_{=1, \text{ per construction}} = 1 \end{aligned} \tag{13}$$