Epidemic spread on random networks with machine learning methods: How to use neural networks to prevent epidemic waves?

Ágnes Backhausz, András Zempléni (ELTE Eötvös Loránd University, Budapest, Hungary, Faculty of Science, Department of Probability Theory and Statistics)

Spreading processes appear around us all the time: news, opinions are circulating on real-world and online social networks, information (and also malware) travels through info-communication systems, and, as we all learned again during the covid-19 pandemic, infectious diseases reach us through personal contacts. The mathematical analysis of such processes can provide a deeper understanding on how to predict, prevent and control these waves. In our project, by using various statistical and machine-learning methods, we elaborated methods that can help to understand and to quantitatively predict an epidemic wave, and hence that can support decision-makers to make interventions when it is necessary. The basis of our model is a random graph with a special, two-layer structure; although this does not include all features of real-world social networks, it was flexible enough for modelling, and the most important characteristics can be built in. One of our main goals was the quantitative comparison of classical and machine learning methods: we could examine when and how it is worth using neural networks, what is the quantity and quality of information that is necessary for good estimates. It is also crucial to understand which are the best methods in the first period of the epidemic spread, because a quick reaction is often the key point in the prevention of huge damages.

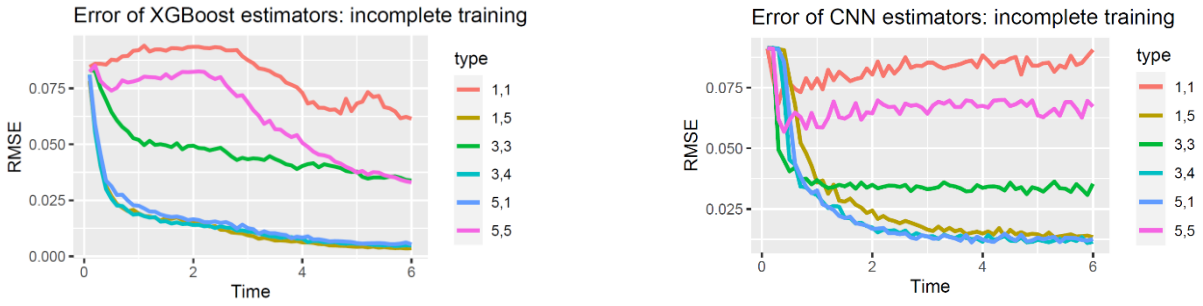
To give a more detailed picture, let us start with a short description of the network (graph) and the spreading process that we used. For simplicity, imagine a city where each household has five members, who meet each other every day. In addition to this, each individual has on average eight friends, who are chosen randomly from the population of the city. Then we use a so-called SIR model, in which vertices can be in susceptible (S), infected (I), or recovered (R) state. At the beginning, only S and I are present. When a vertex gets infected, it will recover after an exponentially distributed random time (with expectation 1), and stay in state R for the rest of the process. If two vertices are connected, meaning that they either live in the same household, or they are friends, and one of them is in state S, the other gets into state I, then an exponentially distributed random variable X with expectation $1/\tau$ is assigned to this connection, and after time X (compared to the time of the infection), the susceptible vertex gets infected – unless it is already infected, or the the other one has already recovered. All randomizations are independent of each other. For more details and information of epidemic spread on networks, see e.g. [4]. To make the model more flexible, randomness is also involved when we choose the underlying networks: when we choose the friends of the individuals, we either use a preferential attachment model (in which individuals with more connections get new friends with larger probability, see e.g. [2]), or generate new groups of equal size uniformly at random. For more details, we refer to [1].

As for the statistical problem in this setup, we assumed that the expected time for recovery is known, and focused on estimating the expected time of infection from an individual in state I to its contact (household member or friend) in state S (without recovery). This can be described with the infection rate τ (defined above), which is the reciprocal of this expected time; the larger τ is, the quicker is the spread of the epidemic, and the quicker we need an intervention to prevent a huge epidemic wave. In order to estimate the infection parameter, the input data were the following, all as functions of time:

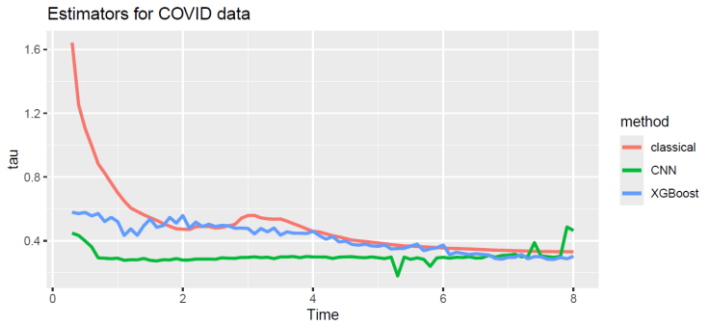
- number of vertices in state S and state I
- the number of S-I edges (in certain cases)
- the average degree of the infected vertices (also only in certain cases)

Given these data, with the classical methods, we used an explicit formula, derived from well-known statistical tools (maximum likelihood estimator). With the machine learning methods, our strategy was the following. We simulated several hundreds of epidemic spreading scenarios with different values of τ, this was our training set. These were given to an XGBoost algorithm (a machine learning method based on decision trees, see [3]), and to an appropriately calibrated convolutional neural network. Then, again, epidemic scenarios were generated, but these were given to the algorithm without the corresponding infection parameter; we got back an estimated τ based on the previous training. Finally, we could compare the real value with the estimate.

Our main results can be summarized as follows. All the three methods (classical, XGBoost, neural network) can lead to the best results in certain periods of the epidemic wave, under certain conditions. For example, if only the number of infected and recovered vertices is known, then the xgboost method is the best (the neural network is just slightly worse). If additional information is known (as listed above), xgboost and neural network perform the best during the increasing period of the epidemic wave, but the classical method becomes better in the decreasing period. That is, although the machine learning methods use more computational capacity, for a successful prevention of a huge wave, they seem to be the best choices. We also showed that it is very important that the structure of the graphs in the training set is similar to the graphs in the test set; otherwise the machine learning methods do not perform well. To examine this, we focused on the size of the households, we trained the machine learning methods on all but one option, by using only the number of S and I vertices, and tested them on one selected household size. The figure below shows the root mean square error (RMSE) of these estimates, as a function of time (the peak was around time 2). The curves are labeled as follows: the first number represents the index of the value missing from the potential clique sizes (7, 8, 9, 10, 11) in the training set, while the second one refers to the index of the clique size in the test set. (E.g. 1,1 refers to the case when the training set contained clique sizes of 8, 9, 10, 11, while in the test set we had only graphs of household size 7.) Where the two numbers are identical, then the graphs with a similar structure to the ones in the training set were missing from the test set, and we get significantly worse results. We can also see that both machine learning methods lead to quite good estimates already in the first, growing period of the wave (the peak was around time 2).



Furthermore, we also compared our methods for real data from the Covid-19 pandemic, from the Isle of Man with 80000 inhabitants (source of data, WHO). In this case, the real infection parameter τ is not known, so we cannot calculate the error of the estimate precisely; still, we can see that the convolutional neural network is the most stable, but the XGBoost estimate is also not far from the final value, already in the first period.



Given these results, our future plans include the application of further neural networks (such as transformers), and to examine opinion spread, where interactions of more than two individuals play a crucial role in the behavior of the process. Altogether, we aim to understand the advantages and disadvantages of simpler and more complex statistical and machine learning methods, in order to achieve efficient tools to predict spreading processes.

References

[1] Backhausz, Á., Bognár, E., Csiszár, V., Tárkányi, D., & Zempléni, A. (2025). Comparison of classical, XGBoost and neural network methods for parameter estimation in epidemic processes on random graphs. *Franklin Open*, 100271.

[2] Barabási, A.-L. (2002). Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life

[3] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).

[4] Kiss, I. Z., Miller, J. C., & Simon, P. L. (2017). Mathematics of epidemics on networks. *Cham: Springer*, *598*(2017), 31.