

Risk score learning for COVID-19 contact tracing apps

Kevin Murphy¹, Abhishek Kumar¹, and Stylianos Serghiou¹

¹Google Research

Abstract

Digital contact tracing apps for COVID-19, such as the one developed by Google and Apple, need to estimate the risk that a user was infected during a particular exposure, in order to decide whether to notify the user to take precautions, such as entering into quarantine, or requesting a test. Such risk score models contain numerous parameters that must be set by the public health authority. In this paper, we show how to automatically learn these parameters from data.

Our method needs access to exposure and outcome data. Although this data is already being collected (in an aggregated, privacy-preserving way) by several health authorities, in this paper we limit ourselves to simulated data, so that we can systematically study the different factors that affect the feasibility of the approach. In particular, we show that the parameters become harder to estimate when there is more missing data (e.g., due to infections which were not recorded by the app), and when there is model misspecification. Nevertheless, the learning approach outperforms a strong manually designed baseline. Furthermore, the learning approach can adapt even when the risk factors of the disease change, e.g., due to the evolution of new variants, or the adoption of vaccines.

1. Introduction

Digital contact tracing (DCT) based on mobile phone technology has been proposed as one of many tools to help combat the spread of COVID-19. Such apps have been shown to reduce COVID-19 infections in simulation studies (e.g., (Ferretti et al., 2020b; Abueg et al., 2021; Cencetti et al., 2020)) and real-world deployments (Wymant et al., 2021; Salathe et al., 2020; Ballouz et al., 2020; Masel et al., 2021; Rodríguez et al., 2021; Kendall et al., 2020; Huang et al., 2020). For example, in a 3-month period, Wymant et al. (2021) estimated that the app used in England and Wales led to about 284,000–594,000 averted infections and 4,200–8,700 averted deaths.

DCT apps work by notifying the app user if they have had a "risky encounter" with an index case (someone who has been diagnosed as having COVID-19 at the time of the encounter). Once notified, the user may then be advised by their Public Health Authority (PHA) to enter into quarantine, just as they would if they had been contacted by a manual contact tracer. The key question which we focus on in this paper is: how can we estimate the probability that an exposure resulted in an infection? If the app can estimate this reliably, it can decide who to notify based on a desired false positive / false negative rate.

To quantify the risk of an encounter, we need access to some (anonymized) features which characterize the encounter. In this paper, we focus on the features collected by the Google/ Apple Exposure Notification (GAEN) system (Google-Apple, 2020), although our

techniques could be extended to work with other DCT apps. The GAEN app uses 3 kinds of features to characterize each exposure: the duration of the encounter, the bluetooth signal strength (a proxy for distance between the two people), and (a quantized version of) the infectiousness of the index case (as estimated by the days since their symptoms started, or the date of their positive test).

Let the set of observed features for the n 'th encounter be denoted by \tilde{x}_n , and let $\tilde{\mathcal{X}}_j = \{\tilde{x}_n : j_n = j\}$ be the set of all exposure features for user j . Finally, let $Y_j = 1$ if user j gets infected from one (or more) of these encounters, and $Y_j = 0$ otherwise. The primary goal of the risk score model is to estimate the probability of infection, $p(Y_j = 1 | \tilde{\mathcal{X}}_j; \psi)$. Although PHAs are free to use any model they like to compute this quantity, the vast majority have adopted the risk score model which we describe in Sec. 3, since this is the one implemented in the Google/Apple app.

A key open problem is how to choose the parameters ψ of this risk score model. Although expert guidance for how to set these parameters has been provided (e.g. (LPH)), in this paper we ask if we can do better using a data-driven approach. In particular, we assume the PHA has access to a set of anonymized, labeled features $\{\tilde{\mathcal{X}}_j, Y_j\}$, where $Y_j \in \{0, 1\}$ is the test result for user j , and $\tilde{\mathcal{X}}_j$ is the set of exposure features recorded by their app. Given this data, we can try to optimize the parameters of the risk score model ψ using weakly-supervised machine learning methods, as we explain in Sec. 4.

Since we do not currently have access to such "real world" labeled datasets, in this paper, we create a synthetic dataset using a simple simulator, which we describe in Sec. 2. We then study the ability of the ML algorithm to estimate the risk score parameters from this simulated data, as we vary the amount of missing data and label noise. We compare our learned model to a widely used manually created baseline, and show that the ML method performs better (at least in our simulations). We also show that it is more robust to changes in the true distribution than the manual baseline.¹

Generalizable Insights about Machine Learning in the Context of Healthcare

In this paper, we show that it is possible to learn interpretable risk score models using standard machine learning tools, such as multiple instance learning and stochastic gradient descent, provided we have suitable data. However, some new techniques are also required, such as replacing hard thresholding with soft binning, and using constrained optimization methods to ensure monotonicity of the learned function. We believe these methods could be applied to learn other kinds of risk score models.

2. A simple simulator of COVID-19 transmission

The probability of infection from an exposure event depends on many factors, including the duration of the exposure, the distance between the index case (transmitter) and the user (receiver), the infectiousness of the index case, as well as other unmeasured factors, such as mask wearing, air flow, etc. In this section, we describe a simple probabilistic model of COVID-19 transmission, which only depends on the factors that are recorded by the

1. A Python Jupyter notebook to reproduce the experiments can be found at https://github.com/google-research/agent-based-epidemic-sim/tree/develop/agent_based_epidemic_sim/learning/MLHC_paper_experiments.ipynb.

Google/Apple app. This model forms the foundation of the GAEN risk score model in Sec. 3. We will also use this model to generate our simulated training and test data, as we discuss in Sec. 4.

2.1. Single exposure

Let τ_n be the duration (in minutes) of the n 'th exposure, and let d_n be the distance (in meters) between the two people during this exposure. (For simplicity, we assume the distance is constant during the entire interval; we will relax this assumption later.) Finally, let σ_n be the time since of the onset of symptoms of the index case at the time of exposure, i.e., $\sigma_n = t_{i_n}^{\text{sym}} - t_{j_n}^{\text{exp}}$, where i_n is the index case for this exposure, j_n is the user, $t_{i_n}^{\text{sym}}$ is the time of symptom onset for i_n , and $t_{j_n}^{\text{exp}}$ is the time that j_n gets exposed to i_n . (If the index case did not show symptoms, we use the date that they tested positive, and shift it back in time by 7 days, as a crude approximation.) Note that σ_n can be negative. In (Ferretti et al., 2020a), they show that σ_n can be used to estimate the degree of contagiousness of the index case.

Given the above quantities, we define the "hazard score" for this exposure as follows:

$$s_n = f_{\text{hazard}}(\tau_n, d_n, \sigma_n; \phi) = \tau_n \times f_{\text{dist}}(d_n; \phi) \times f_{\text{inf}}(\sigma_n; \phi) \quad (1)$$

where τ_n is the duration, d_n is the distance, σ_n is the time since symptom onset, $f_{\text{dist}}(d_n; \phi)$ is the simulated risk given distance, $f_{\text{inf}}(\sigma_n; \phi)$ is the simulated risk given time, and ϕ are parameters of the simulator (as opposed to ψ , which are parameters of the risk score model that the phone app needs to learn).

The "correct" functional form for the dependence on distance is unknown. In this paper, we follow (Briers et al., 2020), who model both short-range (droplet) effects, as well as longer-range (aerosol) effects, using the following simple truncated quadratic model:

$$f_{\text{dist}}(d; \phi) = \min(1, D_{\text{min}}^2/d^2) \quad (2)$$

They propose to set $D_{\text{min}}^2 = 1$ based on an argument from the physics of COVID-19 droplets.

The functional form for the dependence on symptom onset is better understood. In particular, (Ferretti et al., 2020a) consider a variety of models, and find that the one with the best fit to the empirical data was a scaled skewed logistic distribution:

$$f_{\text{inf}}(\sigma; \phi) \propto \text{SkewedLogistic}(\sigma | \mu = -4, \sigma = 1.85, \alpha = 5.85, \tau = 5.42) \quad (3)$$

In order to convert the hazard score into a probability of infection (so we can generate samples), we use a standard exponential dose response model (Smieszek, 2009; Haas, 2014):

$$p_n = \Pr(y_n = 1 | x_n; \phi) = 1 - e^{-\lambda s_n} \quad (4)$$

where $y_n = 1$ iff exposure n resulted in an infection, and $x_n = (\tau_n, d_n, \sigma_n)$ are the features of this exposure. The parameter λ is a fixed constant with value 3.1×10^{-6} , chosen to match the empirical attack rate reported in (Wilson et al., 2020).

2.2. Multiple exposures

A user may have multiple exposures. Following standard practice, we assume each exposure could independently infect the user, so

$$P_j = \Pr(Y_j = 1 | \mathcal{X}_j; \phi) = 1 - \prod_{n \in E_j} (1 - p_n) \quad (5)$$

$$= 1 - \prod_{n \in E_j} e^{-\lambda s_n} = 1 - e^{-\lambda \sum_n s_n} \quad (6)$$

$$= p_1 + (1 - p_1) \times p_2 + (1 - p_1)(1 - p_2) \times p_3 + \dots \quad (7)$$

where E_j are all of j 's exposure events, and \mathcal{X}_j is the corresponding set of features.

To account for possible "background" exposures that were not recorded by the app, we can add a p_0 term, to reflect the prior probability of an exposure for this user (e.g., based on the prevalence). This model has the form

$$P_j = 1 - (1 - p_0) \prod_{n \in E_j} (1 - p_n) \quad (8)$$

2.3. A bipartite social network

Suppose we have a set of J users. The probability that user j gets infected is given by Eq. (8). However, it remains to specify the set of exposure events E_j for each user. To do this, we create a pool of N randomly generated exposure events, uniformly spaced across the 3d grid of distances (80 points from 0.1m to 5m), durations (20 points from 5 minutes to 60 minutes), and symptom onset times (-10 to 10 days). This gives us a total of $N = 33600$ exposure events.

We assume that each user j is exposed to a random subset of one or more of these events (corresponding to having encounters with different index cases). In particular, let $u_{jn} = 1$ iff user j is exposed to event n . Hence the probability that j is infected is given by $P_j = 1 - \exp[-\lambda S_j]$, where $S_j = \sum_{n=1}^N u_{jn} s_n$ and $s_n = f_{\text{hazard}}(x_n; \phi)$, where $x_n = (\tau_n, d_n, \sigma_n)$. We can write this in a more compact way as follows. Let $\mathbf{U} = [u_{jn}]$ be an $J \times N$ binary assignment matrix, and \mathbf{s} be the vector of event hazard scores. For example, suppose there are $J = 3$ users and $N = 5$ events, and we have the following assignments: user $j = 1$ gets exposed to event $n = 1$; user $j = 2$ gets exposed to events $n = 1, n = 2$ and $n = 3$; and user $j = 3$ gets exposed to events $n = 3$ and $n = 5$. Thus

$$\mathbf{U} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (9)$$

The corresponding vector of true infection probabilities, one per user, is then given by $\mathbf{p} = 1 - \exp[-\lambda \mathbf{U} \mathbf{s}]$. From this, we can sample a vector of infection labels, \mathbf{y} , one per user. If a user gets infected, we assume they are removed from the population, and cannot infect anyone else (i.e., the pool of N index cases is fixed). This is just for simplicity. We leave investigation of more realistic population-based simulations to future work.

2.4. Censoring

In the real world, a user may get exposed to events that are not recorded by their phone. We therefore make a distinction between the events that a user was actually exposed to, encoded by u_{jn} , and the events that are visible to the app, denoted by v_{jn} . We require that $v_{jn} = 1 \implies u_{jn} = 1$, but not vice versa. For example, Eq. (9) might give rise to the following censored assignment matrix:

$$\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 1 & 0 & \mathbf{0} \end{pmatrix} \quad (10)$$

where the censored events are shown in bold. This censored version of the data is what is made available to the user’s app when estimating the exposure risk, as we discuss in Sec. 4. We assume censoring happens uniformly at random. We leave investigation of more realistic censoring simulations to future work.

2.5. Bluetooth simulator

DCT apps do not observe distance directly. Instead, they estimate it based on bluetooth attenuation. In order to simulate the bluetooth signal, which will be fed as input to the DCT risk score model, we use the stochastic forwards model from (Lovett et al., 2020). This models the attenuation as a function of distance using a log-normal distribution, with a mean given by

$$\mathbb{E}[a|d; \phi] = e^{\phi_\alpha + \phi_\beta \log(d)} = e^{\phi_\alpha} d^{\phi_\beta} \quad (11)$$

Using empirical data from MIT Lincoln Labs, Lovett et al. (2020) estimate the offset to be $\phi_\alpha = 3.92$ and the slope to be $\phi_\beta = 0.21$.

In this paper, we assume this mapping is deterministic, even though in reality it is quite noisy, due to multipath reflections and other environmental factors (see e.g., (Leith and Farrell, 2020)). Fortunately, various methods have been developed to try to “denoise” the bluetooth signal. For example, the England/Wales GAEN app uses an approach based on unscented Kalman smoothing (Lovett et al., 2020). We leave the study of the effect of bluetooth noise on our learning methods to future work.

3. The app’s risk score model

In this section, we describe risk score model used by Google/Apple Exposure Notification (GAEN) system. This can be thought of as a simple approximation to the biophysical model we discussed in Sec. 2.

For every exposure recorded by the app, the following risk score is computed:

$$r_n = f_{\text{risk}}(\tau_n, a_n, c_n; \psi) = \tau_n \times f_{\text{ble}}(a_n; \psi) \times f_{\text{con}}(c_n; \psi) \quad (12)$$

where τ_n is the duration, a_n is the bluetooth attenuation, c_n is a quantized version of symptom onset time, and where f_{ble} and f_{con} are functions defined below.

We can convert the risk score into an estimated probability of being infected using

$$q_n = \Pr(y_n = 1 | \tilde{x}_n; \psi) = 1 - \exp[-\mu r_n] \quad (13)$$

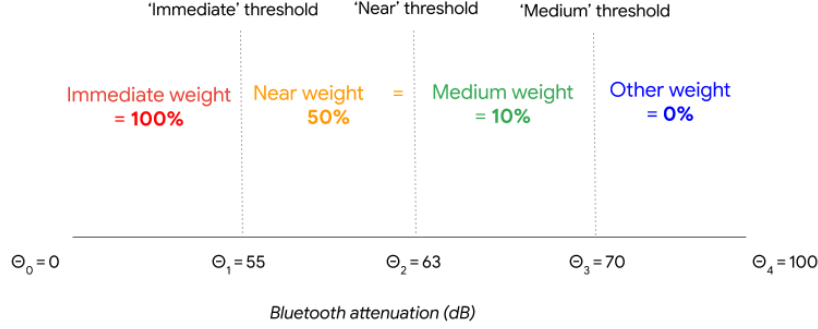


Figure 1: 3 thresholds defines 4 attenuation buckets, each of which can be given a weight. We define $\theta_0^{\text{ble}} = 0$ and $\theta_4^{\text{ble}} = 100$ as boundary conditions. These weights encode the assumption that smaller attenuations (corresponding to closer exposures) have higher risk than larger attenuations.

where $\tilde{x}_n = (\tau_n, a_n, c_n)$ are the observed features recorded by the GAEN app, and μ is a learned scaling parameter, analogous to λ for the simulator.

3.1. Estimated risk vs bluetooth attenuation

The GAEN risk score model makes a piecewise constant approximation to the risk-vs-attenuation function. In particular, it defines 3 attenuation thresholds, $\theta_{1:3}^{\text{ble}}$, which partitions the real line into 4 intervals. Each interval or bucket is assigned a weight, $w_{1:4}^{\text{ble}}$, as shown in Fig. 1.

Overall, we can view this as approximating $f_{\text{dist}}(d_n; \phi)$ by $f_{\text{ble}}(a_n; \psi)$, where

$$f_{\text{ble}}(a_n; \psi) = \begin{cases} w_1^{\text{ble}} & \text{if } a_n \leq \theta_1^{\text{ble}} \\ w_2^{\text{ble}} & \text{if } \theta_1^{\text{ble}} < a_n \leq \theta_2^{\text{ble}} \\ w_3^{\text{ble}} & \text{if } \theta_2^{\text{ble}} < a_n \leq \theta_3^{\text{ble}} \\ w_4^{\text{ble}} & \text{if } \theta_3^{\text{ble}} > a_n \end{cases} \quad (14)$$

3.2. Estimated risk vs days since symptom onset

The health authority can compute the days since symptom onset for each exposure, $\sigma_n = t_{i_n}^{\text{sym}} - t_{j_n}^{\text{exp}}$, by asking the index case when they first showed symptoms. (If they did not show any symptoms, they can use a heuristic, such as the date of their test shifted back by several days.) However, sending the value of σ_n directly was considered to be a security risk by Google/Apple. Therefore, to increase privacy, the symptom onset value is mapped into one of 3 infectiousness or contagiousness levels; this mapping is defined by a lookup table as shown in Fig. 2. The three levels are called "none", "standard" and "high". We denote this mapping by

$$\underbrace{c_{n,l}}_{\text{contagiousness level}} = \mathbb{I} \left(\underbrace{\theta_{l-1}^{\text{con}}}_{\text{time since symptom onset}} < \underbrace{\sigma_n}_{\text{time since symptom onset}} \leq \underbrace{\theta_l^{\text{con}}}_{\text{time since symptom onset}} \right)$$

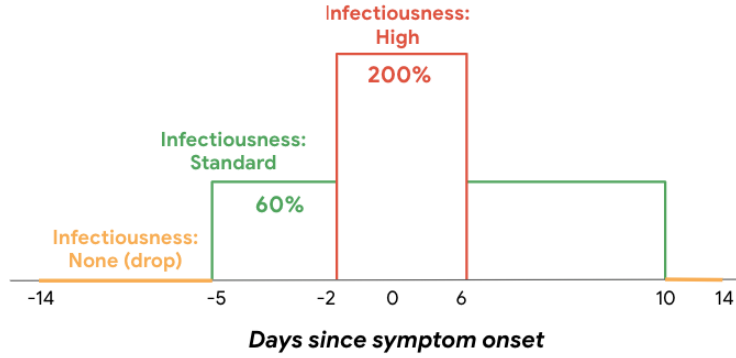


Figure 2: A mapping from symptom onset to 3 infectiousness levels with corresponding weights.

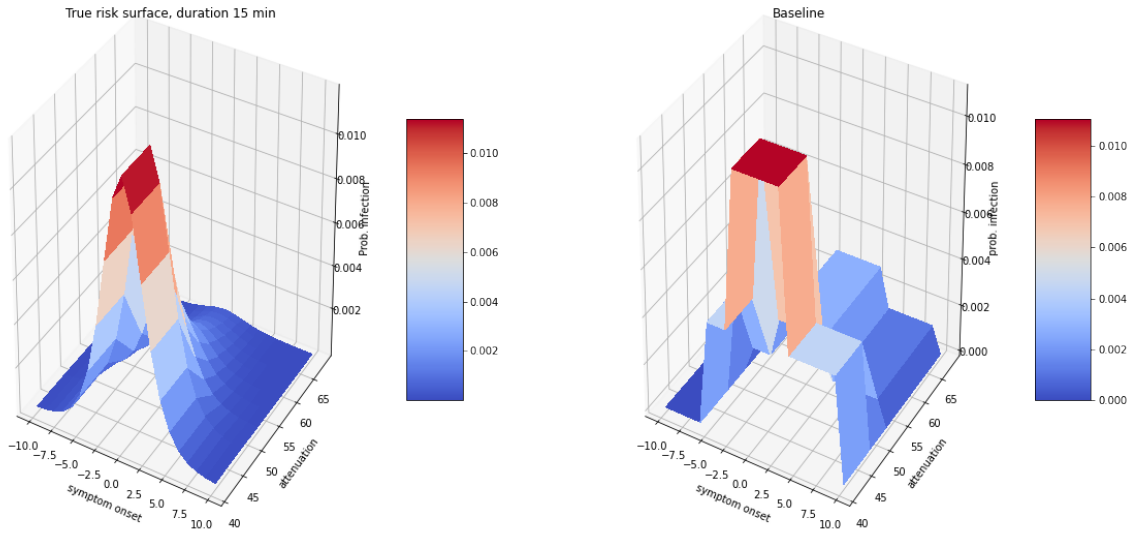


Figure 3: Risk surface used by (left) our simulator and (right) the app. We plot risk as a function of attenuation and symptom onset. We assume the duration of the exposure is $\tau = 15$ minutes.

Each infectiousness level is associated with a weight. (However, we require that the weight for level "none" is $w_1^{\text{con}} = 0$, so there are only 2 weights that need to be specified.) Overall, the mapping and the weights define a piecewise constant approximation to the risk vs symptom onset function defined in Eq. (3). We can write this approximation as $f_{\text{inf}}(\sigma_n; \psi) \approx f_{\text{con}}(\text{LUT}(\sigma_n); \psi)$, where

$$f_{\text{con}}(c_n; \psi) = \begin{cases} 0 & \text{if } c_n = 1 \\ w_2^{\text{con}} & \text{if } c_n = 2 \\ w_3^{\text{con}} & \text{if } c_n = 3 \end{cases} \quad (15)$$

3.3. Comparing the simulated and estimated risk score models

In Fig. 3(left), we plot the probability of infection according to our simulator as a function of symptom onset σ_n and distance d_n , for a fixed duration $\tau_n = 15$. We see that this is the convolution of the (slightly asymmetric) bell-shaped curve from the symptom onset function in Eq. (3) along the x-axis with the truncated quadratic curve from the distance function in Eq. (2) along the z-axis. We can approximate this risk surface using the GAEN piecewise constant approximation, as shown in Fig. 3(right).

3.4. Multiple exposures

We combine the risk from multiple exposure windows by adding the risk scores, reflecting the assumption of independence between exposure events. For example, if there are multiple exposures within an exposure window, each with different duration and attenuation, we can combine them as follows. First we compute the total time spent in each bucket:

$$\underbrace{\tau_{nb}}_{\text{duration in bucket } b} = \sum_{k=1}^{K_n} \underbrace{\tau_{nk}}_{\text{duration}} \times \underbrace{\mathbb{I}(\theta_{b-1}^{\text{ble}} < a_{nk} \leq \theta_b^{\text{ble}})}_{\text{attenuation is in bucket } b} \quad (16)$$

where K_n is the number of "micro exposures" within the n 'th exposure window, τ_{nk} is the time spent in the k 'th micro exposure, and a_{nk} is the corresponding attenuation. (This gives a piecewise constant approximation to the distance-vs-time curve for any given interaction between two people.) We then compute the overall risk for this exposure using the following bilinear function:

$$\underbrace{r_n}_{\text{risk score}} = \underbrace{\left[\sum_{b=1}^{N_B} \tau_{nb} w_b^{\text{ble}} \right]}_{\text{weighted exposure minutes}} \times \underbrace{\left[\sum_{\ell=1}^{N_C} \mathbb{I}(c_{n,\ell}) w_\ell^{\text{con}} \right]}_{\text{weighted contagiousness level}} \quad (17)$$

where $N_B = 4$ is the number of attenuation buckets, and $N_C = 3$ is the number of contagiousness levels.

If we have multiple exposures for a user, we sum the risk scores to get $R_j = \sum_{n \in E_j} r_n$, which we can convert to a probability of infection using $Q_j = 1 - e^{-\mu R_j}$.

4. Learning the risk score

In this section, we discuss how to optimize the parameters of the risk score model using machine learning. We assume access to a labeled dataset, $\{(\tilde{\mathcal{X}}_j, Y_j) : j = 1 : J\}$, for a set of J users. This data is generated using the simulator described in Sec. 2.

We assume the lookup table mapping from symptom onset to infectiousness level is fixed, as shown in Fig. 2; however, we assume the corresponding infection weights w_2^{con} and w_3^{con} are unknown. Similarly, we assume the 3 attenuation thresholds $\theta_{1:3}^{\text{ble}}$, as well as the 4 corresponding weights, $w_{1:4}^{\text{ble}}$, are unknown. Finally, we assume the scaling factor μ is unknown. Thus there are 10 parameters in total to learn; we denote these by $\psi = (w_{2:3}^{\text{con}}, \theta_{1:3}^{\text{ble}}, w_{1:4}^{\text{ble}}, \mu)$. It is clear that the parameters are not uniquely identifiable. For example, we could increase

μ and decrease w^{con} and the effects would cancel out. Thus there are many parameter settings that all obtain the same maximum likelihood. Our goal is just to identify one element of this set.

In the sections below, we describe some of the challenges in learning these parameters, and then our experimental results.

4.1. Learning from data with label noise and censoring

We optimize the parameters by maximizing the log-likelihood, or equivalently, minimizing the binary cross entropy:

$$\mathcal{L}(\psi) = - \sum_{j=1}^J Y_j \log Q_j + (1 - Y_j) \log(1 - Q_j) \quad (18)$$

where $Y_j \in \{0, 1\}$ is the infection label for user j coming from the simulator, and

$$Q_j = 1 - \exp \left[-\mu \sum_{n \in E_j} f_{\text{risk}}(\tilde{x}_n; \psi) \right] \quad (19)$$

is the estimated probability of infection, which is computed using the bilinear risk score model in Eq. (17).

A major challenge in optimizing the above objective arises due to the fact that a user may encounter multiple exposure events, but we only see the final outcome from the whole set, not for the individual events. This problem is known as multi-instance learning (see e.g., (Foulds and Frank, 2010)). For example, consider Eq. (9). If we focus on user $j = 2$, we see that they were exposed to a "bag" of 3 exposure events (from index cases 1, 2, and 3). If we observe that this user gets infected (i.e., $Y_j = 1$), we do not know which of these events was the cause. Thus the algorithm does not know which set of features to pay attention to, so the larger the bag each user is exposed to, the more challenging the learning problem. (Note, however, that if $Y_j = 0$, then we know that all the events in the bag must be labeled as negative, since we assume (for simplicity) that the test labels are perfectly reliable.)

In addition, not all exposure events get recorded by the GAEN app. For example, the index case i may not be using the app, or the exposure may be due to environmental transmission (known as fomites). This can be viewed as a form of measurement "censoring", as illustrated in Eq. (10), which is a sparse subset of the true infection matrix in Eq. (9). This can result in a situation in which all the *visible* exposure events in the bag are low risk, but the user is infected anyway, because the true cause of the user's infection is not part of $\tilde{\mathcal{X}}_j$. This kind of false positive is a form of label noise which further complicates learning.

4.2. Optimization

In this section, we discuss some algorithmic issues that arise when trying to optimize the objective in Eq. (18).

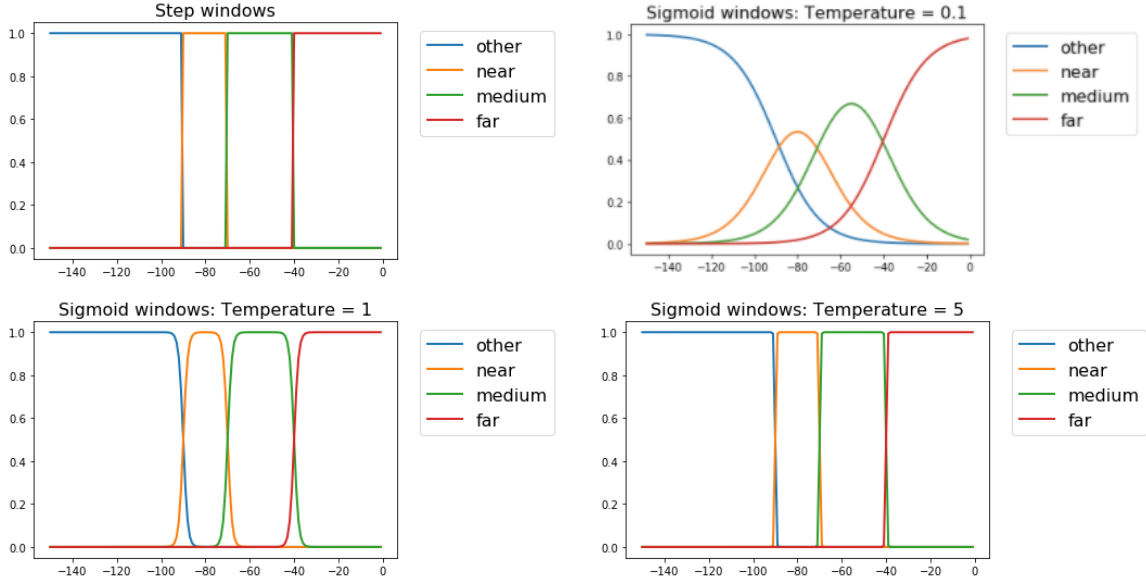


Figure 4: (Top left) Hard threshold. (Other). Sigmoidal approximation at increasing temperature.

4.2.1. MONOTONOCITY

We want to ensure that the risk score is monotonically increasing in attenuation. To do this, we can order the attenuation buckets from low risk to high, so τ_{n1} is time spent in lowest risk bin (largest attenuation), and τ_{n4} is the time spent in the highest risk bin (smallest attenuation). Next we reparameterize w^{ble} as follows:

$$w^{\text{ble}} = [w_1^{\text{ble}}, w_2^{\text{ble}} = w_1^{\text{ble}} + \Delta_2^{\text{ble}}, w_3^{\text{ble}} = w_2^{\text{ble}} + \Delta_3^{\text{ble}}, w_4^{\text{ble}} = w_3^{\text{ble}} + \Delta_4^{\text{ble}}] \quad (20)$$

Then we optimize over $(w_1^{\text{ble}}, \Delta_2^{\text{ble}}, \Delta_3^{\text{ble}}, \Delta_4^{\text{ble}})$, where we use projected gradient descent to ensure $\Delta_b^{\text{ble}} > 0$. We can use a similar trick to ensure the risk is monotonically increasing with the infectiousness level.

4.2.2. SOFT THRESHOLDING

The loss function is not differentiable wrt the attenuation thresholds θ_b^{ble} . We consider two solutions to this. In the first approach, we use a gradient-free optimizer for θ_b^{ble} in the outer loop (e.g., grid search), and a gradient-based optimizer for the weights w_b^{ble} in the inner loop.

In the second approach, we replace the hard binning in Eq. (16) with soft binning, as follows:

$$\tau_{nb} = \sum_k \tau_{nk} \mathbb{I}(\theta_{b-1}^{\text{ble}} < a_{nk} \leq \theta_b^{\text{ble}}) \quad (21)$$

$$\approx \sum_k \tau_{nk} \sigma_\tau(a_{nk} - \theta_{b-1}^{\text{ble}}) \sigma_\tau(\theta_b^{\text{ble}} - a_{nk}) \quad (22)$$

where $\sigma_t(x) = \frac{1}{1+e^{-tx}}$ is the sigmoid (logistic) function, and t is a temperature parameter. The effect of t is illustrated in Fig. 4. When learning, we start with a small temperature,

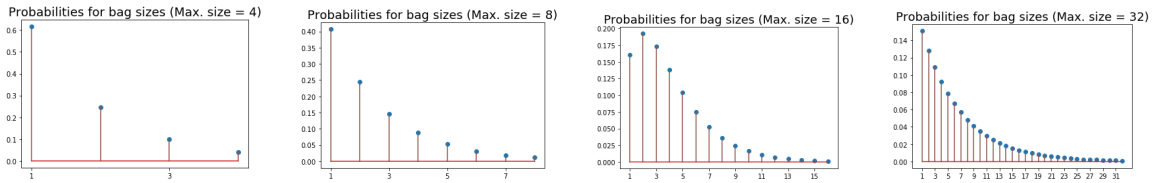


Figure 5: Multi-instance bag simulation: probabilities for sampling bag sizes for a given maximum bag size (from left to right: 4, 8, 16, 32).

and then gradually increase it until we approximate the hard-threshold form, as required by the GAEN app.

The advantage of the soft binning approach is that we can optimize the thresholds, as well as the weights, using gradient descent. This is much faster and simpler than the nested optimization approach, in which we use grid search in the outer loop. Furthermore, preliminary experiments suggested that the two approaches yield similar results. We will therefore focus on soft binning for the rest of this paper.

4.3. Experiments

In this section, we describe our experimental results.

4.3.1. SETUP

We generated a set of simulated exposures from a fine uniform quantization of the three dimensional grid, corresponding to duration \times distance \times symptom onset.² For each point in this 3d grid, we sampled a label from the Bernoulli distribution parameterized by the probability of infection using the infection model described in Sec. 2.

After creating this "pool" of exposure events, we next assigned a random bag of size k of these events to each user. The value k is sampled from a truncated negative binomial distribution with parameters (p, r) , where the truncation parameter is the maximum bag size b . Figure 5 shows the probabilities of bag sizes for different settings of maximum bag size b . Note that larger bag sizes make the multi-instance learning problem harder, because of the difficulty of "credit assignment".

Negative bags are composed of all negative exposures. We consider two scenarios for constructing positive bags: (i) each positive bag contains exactly one positive exposure and rest are negative exposures, (ii) each positive bag contains N positive exposures, where N is sampled uniformly from $\{1, 2, 3\}$. When there is only one positive in a bag, the learning problem is harder, since it is like finding a "needle in a haystack". (This corresponds to not knowing which of the people that you encountered actually infected you.) When there are more positives in a bag, the learning problem is a little easier, since there are "multiple needles", any of which are sufficient to explain the overall positive infection.

2. The empirical distribution of duration, distance and onset can be estimated using the Exposure Notifications Privacy Analytics tool (see <https://implementers.lfph.io/analytics/>). Some summary statistics shared with us by the MITRE organization, that runs this service, confirms that the marginal distributions of each variable are close to uniform.

RISK SCORE LEARNING FOR COVID

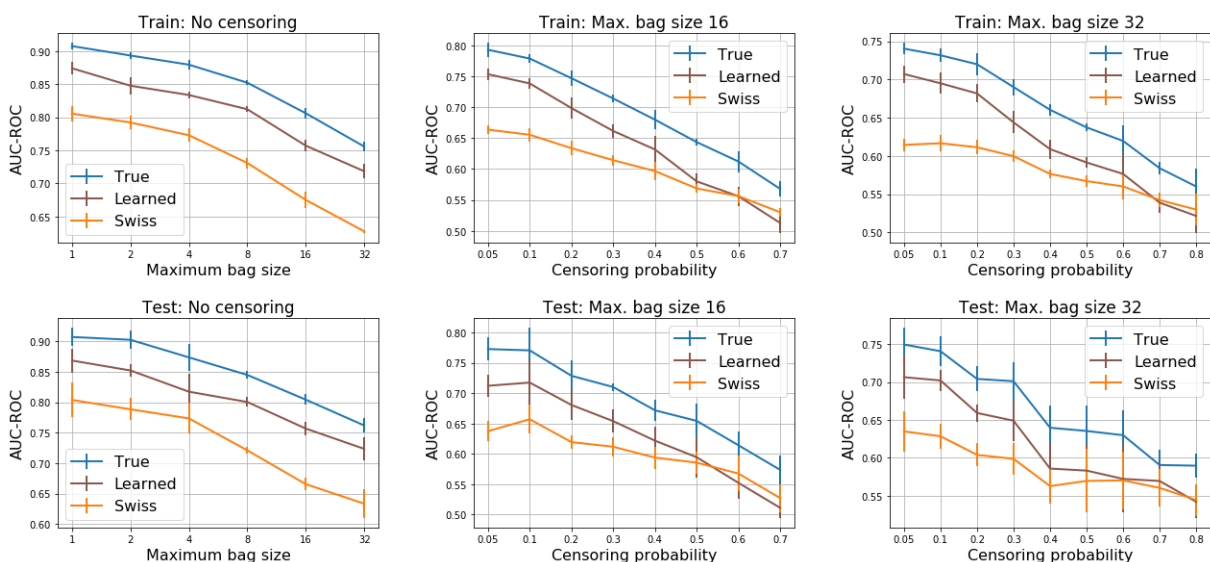


Figure 6: Learning performance (AUC) with increasing problem difficulty. Top row: training set. Bottom row: test set. Left column: No censoring. Performance vs bag size. Middle column: Performance vs censoring probability, bag size 16. Right column: Performance vs censoring probability, bag size 32. Each positive bag contains exactly one positive exposure.

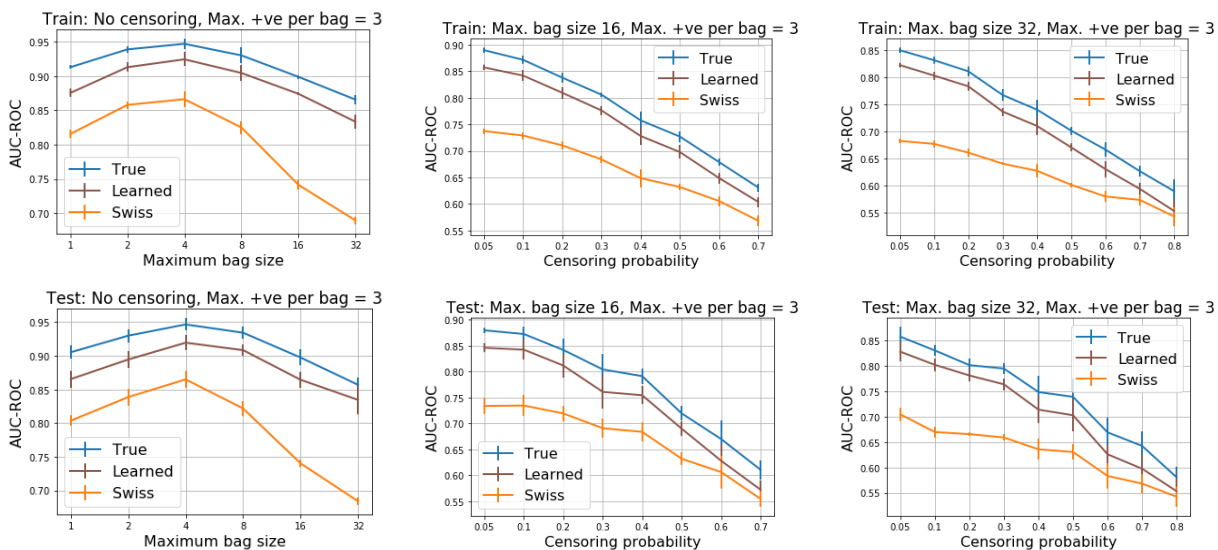


Figure 7: Same as Fig. 6 except now each positive bag may contain up to three positive exposures (number sampled uniformly from $\{1, 2, 3\}$).

To simulate censored exposures, we censor the positive exposures in each positive bag independently and identically with probability varying in the range $\{0.05, 0.1, 0.2, \dots, 0.8\}$. We do not censor the negative exposures to prevent the bag sizes from changing too much from the control value.

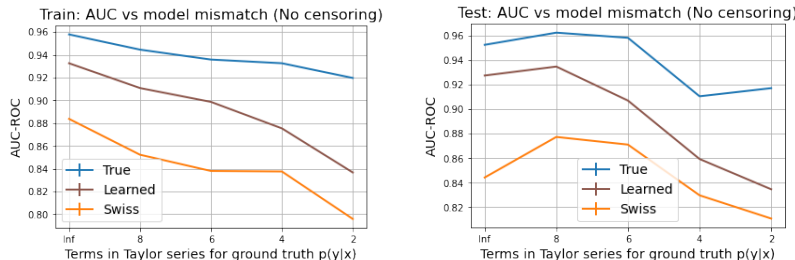


Figure 8: Learning performance (AUC) with increasing model mismatch for train and test set. We do not censor the exposures and the bag size is fixed to 4. Each positive bag may contain up to three positive exposures (same setup as in Fig. 7). The AUC worsens with increasing model mismatch but the learned scores still outperform the ‘Swiss’ manual configuration.

We use 80% of the data for training, and 20% for testing. We fit the model using 1000 iterations of (projected) stochastic gradient descent, with a batch size of 100. We then study the performance (as measured by the area under the ROC curve, or AUC) as a function of the problem difficulty along two dimensions: (i) multi-instance learning (*i.e.*, bag-level labels), and (ii) label noise (*i.e.*, censored exposures). We compare the performance of the learned parameters with that of two baselines. The first corresponds to the oracle performance using the true probabilities coming from the simulator. The second is a more realistic comparison, namely the risk score configuration used by the Swiss version of the GAEN app (Salathe et al., 2020), whose parameters have been widely copied by many other health authorities. This parameter configuration uses 2 attenuation thresholds of $\theta_1^{\text{ble}} = 53$ and $\theta_2^{\text{ble}} = 60$. The corresponding weights for the 3 bins are $w_1^{\text{ble}} = 1.0$, $w_2^{\text{ble}} = 0.5$ and $w_3^{\text{ble}} = 0.0$ (so exposures in the third bin are effectively ignored). The infectiousness weight is kept constant for all symptom onset values, so is effectively ignored.

4.3.2. RESULTS

We do 5 random trials, and report the mean and standard error of the results. Figure 6 and Fig. 7 show the results for two ways of constructing positive bags: (i) each positive bag containing exactly one positive exposure, and (ii) each positive bag containing up to three positive exposures. Not surprisingly, the oracle performance is highest. However, we also see that the learned parameters outperform the Swiss baseline in all settings, often by a large margin.

In terms of the difficulty of the learning problem, the general trend is that the AUC decreases with increasing bag size as expected. The AUC also decreases as the censoring probability is increased. The only exception is the case of fully observed exposures, when each positive bag can have up to three positive exposures, where the AUC increases in the beginning as the bag size is increased (up to bag size of 4) and then starts decreasing again. This is expected since the learning algorithm gets more signal to identify positive bags. This signal starts fading again as the bag size increases beyond 4.

4.3.3. ROBUSTNESS TO MODEL MISMATCH

For a given risk score r_n , we take the functional form of the learned risk score model to be $p_l(y_n|x_n) = 1 - \exp(-\mu r_n)$ in all earlier experiments. This functional form matches the exponential dose response model used in our simulator where the probability of infection is given by $p_s(y_n|x_n) = 1 - \exp(-\lambda s_n)$ for hazard s_n (note that r_n and s_n still have different functional forms and are based on different inputs). Here we simulate model mismatch in $p(y_n|x_n)$ and do preliminary experiments on how it might impact the performance of the learned risk score model.

In more detail, we use $p_s(y_n|x_n) = 1 - f_t(-\mu s_n)$ for the dose response model in the simulator, where f_t is the truncated Taylor series approximation of the exponential with t terms. This reflects the fact that our simulator may not be a very accurate of how COVID-19 is actually transmitted. The functional form of learned model $p_l(y_n|x_n)$ is kept unchanged, since that is required by the app. We vary $t \in \{2, 4, 6, 8\}$ and plot the AUC for the learned model in Figure 8. We fix the bag size to 4 with each bag containing up to three positive exposures randomly sampled from $\{1, 2, 3\}$. As expected, the AUC gets worse as the model mismatch increases, however, the learned scores still outperform the manual ‘Swiss’ configuration. We hypothesize that increasing the model capacity for computing the risk score r_n (e.g., using a multilayer neural network or Gaussian Processes) can impart some robustness to model mismatch, but we leave as a direction for future work.

5. Related work

Although there have been several papers which simulate the benefits of using digital contact tracing apps such as GAEN (e.g., (Ferretti et al., 2020b; Abueg et al., 2021)), all of these papers assume the risk score model is known. This is also true for papers that focus on inferring the individual risk using graphical models of various forms (e.g., (Cencetti et al., 2020; Herbrich et al., 2020)).

The only paper we are aware of that tries to learn the risk score parameters from data is (Sattler et al., 2020). They collect a small ground truth dataset of distance-attenuation pairs from 50 pairs of soldiers walking along a prespecified grid, whereas we use simulated data. However, they only focus on distance and duration, and ignore infectiousness of the index case. By contrast, our risk score model matches the form used in the GAEN app, so the resulting learned parameters could (in principle) be deployed “in the wild”.

In addition, (Sattler et al., 2020) uses standard supervised learning, whereas we consider the case of weakly supervised learning, in which the true infection outcome from each individual exposure event is not directly observed; instead users only get to see the label for their entire “bag” of exposure events (which may also have false negatives due to censoring). Such weakly supervised learning is significantly harder from an ML point of view, but is also much more realistic of the kinds of techniques that would be needed in practice.

6. Discussion

We have shown that it is possible to optimize the parameters of the risk score model for the Google/Apple COVID-19 app using machine learning methods, even in the presence

of considerable missing data and model mismatch. The resulting risk score configuration resulted in much higher AUC scores than those produced by a widely used baseline.

Limitations The most important limitation of this paper is that this is a simulation study, and does not use real data (which was not available to the authors).

The second limitation is that we have assumed a centralized, batch setting, in which a single dataset is collected, based on an existing set of users, and then the parameters are learned, and broadcast back to new app users. However, in a real deployment, the model would have to be learned online, as the data streams in, and the app parameters would need to be periodically updated. (This can also help if the data distribution is non-stationary, due to new variants, or the adoption of vaccines.)

The third limitation is that we have not considered privacy issues. The GAEN system was designed from the ground-up to be privacy preserving, and follows similar principles to the PACT (Private Automated Contact Tracing) protocol (<https://pact.mit.edu/>). In particular, all the features are collected anonymously. However, joining these exposure features x with epidemiological outcomes y in a central server might require additional protections, such as the use of differential privacy (see e.g., (Abadi et al., 2016)), which could reduce statistical performance. (The privacy / performance tradeoff in DCT is discussed in more detail in (Bengio et al., 2021).) We could avoid the use of a central server if we used federated learning (see e.g., (Kairouz and McMahan, 2021)), but this raises many additional challenges from a practical and statistical standpoint.

The fourth limitation is more fundamental, and stems from our finding that learning performance degrades when the recorded data does not contain all the relevant information. This could be a problem in practical settings, where app usage may be rare, so that the cause of most positive test results remain "unexplained". Without enough data to learn from, any ML method is limited in its utility.

References

- Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *23rd ACM Conference on Computer and Communications Security (CCS)*, 2016. URL <http://arxiv.org/abs/1607.00133>.
- Matthew Abueg, Robert Hinch, Neo Wu, Luyang Liu, William J M Probert, Austin Wu, Paul Eastham, Yusef Shafi, Matt Rosencrantz, Michael Dikovsky, Zhao Cheng, Anel Nur-tay, Lucie Abeler-Dörner, David G Bonsall, Michael V McConnell, Shawn O’Banion, and Christophe Fraser. Modeling the combined effect of digital exposure notification and non-pharmaceutical interventions on the COVID-19 epidemic in washington state. *NPJ Digital Medicine*, (medrxiv:2020.08.29.20184135v1), 2021. URL <https://www.medrxiv.org/content/10.1101/2020.08.29.20184135v1.abstract>.
- Tala Ballouz, Dominik Menges, Helene E Aschmann, Anja Domenghino, Jan S Fehr, Milo A Puhan, and Viktor von Wyl. Digital proximity tracing app notifications lead to faster quarantine in non-household contacts: results from the zurich SARS-CoV-2 cohort study. December 2020. URL <http://medrxiv.org/lookup/doi/10.1101/2020.12.21.20248619>.
- Yoshua Bengio, Daphne Ippolito, Richard Janda, Max Jarvie, Benjamin Prud’homme, Jean-François Rousseau, Abhinav Sharma, and Yun William Yu. Inherent privacy limitations of decentralized

- contact tracing apps. *J. Am. Med. Inform. Assoc.*, 28(1):193–195, January 2021. URL <http://dx.doi.org/10.1093/jamia/ocaa153>.
- Mark Briers, Marcos Charalambides, and Chris Holmes. Risk scoring calculation for the current NHSx contact tracing app. May 2020. URL <http://arxiv.org/abs/2005.11057>.
- Giulia Cencetti, Gabriele Santin, Antonio Longa, Emanuele Pigani, Alain Barrat, Ciro Cattuto, Sune Lehmann, and Bruno Lepri. Using real-world contact networks to quantify the effectiveness of digital contact tracing and isolation strategies for covid-19 pandemic. *medRxiv*, page 2020.05.29.20115915, May 2020. URL <https://www.medrxiv.org/content/10.1101/2020.05.29.20115915v1.abstract>.
- Luca Ferretti, Alice Ledda, Chris Wymant, Lele Zhao, Virginia Ledda, Lucie Abeler-Dörner, Michelle Kendall, Anel Nurtay, Hao-Yuan Cheng, Ta-Chou Ng, Hsien-Ho Lin, Rob Hinch, Joanna Masel, A Marm Kilpatrick, and Christophe Fraser. The timing of COVID-19 transmission. September 2020a. URL <https://www.medrxiv.org/content/10.1101/2020.09.04.20188516v1.abstract>.
- Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491), May 2020b. URL <http://dx.doi.org/10.1126/science.abb6936>.
- James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *Knowl. Eng. Rev.*, 25(1):1–25, March 2010.
- Google-Apple. Exposure notifications: Using technology to help public health authorities fight covid-19, 2020. URL <https://www.google.com/covid19/exposurenotifications/>.
- Charles N Haas. *Quantitative Microbial Risk Assessment (2nd edn)*. Wiley, July 2014. URL https://openlibrary.org/books/OL27557844M/Quantitative_Microbial_Risk_Assessment.
- Ralf Herbrich, Rajeev Rastogi, and Roland Vollgraf. CRISP: A probabilistic model for Individual-Level COVID-19 infection risk estimation based on contact data. June 2020. URL <http://arxiv.org/abs/2006.04942>.
- Zhilian Huang, Huiling Guo, Yee-Mun Lee, Eu Chin Ho, Hou Ang, and Angela Chow. Performance of Digital Contact Tracing Tools for COVID-19 Response in Singapore: Cross-Sectional Study. *JMIR Mhealth Uhealth*, 8(10):e23148, October 2020. URL <http://dx.doi.org/10.2196/23148>.
- Peter Kairouz and H Brendan McMahan, editors. *Advances and Open Problems in Federated Learning*, volume 14. Now Publishers, 2021. URL <https://arxiv.org/abs/1912.04977>.
- Michelle Kendall, Luke Milsom, Lucie Abeler-Dörner, Chris Wymant, Luca Ferretti, Mark Briers, Chris Holmes, David Bonsall, Johannes Abeler, and Christophe Fraser. Epidemiological changes on the isle of wight after the launch of the NHS test and trace programme: a preliminary analysis. *The Lancet Digital Health*, October 2020. URL [https://doi.org/10.1016/S2589-7500\(20\)30241-7](https://doi.org/10.1016/S2589-7500(20)30241-7).
- Douglas J Leith and Stephen Farrell. Measurement-based evaluation of Google/Apple exposure notification API for proximity detection in a light-rail tram. *PLoS One*, 15(9):e0239943, September 2020. URL <http://dx.doi.org/10.1371/journal.pone.0239943>.
- LFPH. Configuring Exposure Notification Risk Scores for COVID-19 (Linux Foundation Public Health). URL <https://github.com/lfph/gaen-risk-scoring/blob/main/risk-scoring.md>.

- Tom Lovett, Mark Briers, Marcos Charalambides, Radka Jersakova, James Lomax, and Chris Holmes. Inferring proximity from bluetooth low energy RSSI with unscented kalman smoothers. *arXiv*, July 2020. URL <http://arxiv.org/abs/2007.05057>.
- Joanna Masel, Alexandra Nicole Shilen, Bruce H Helming, Jenna Doucett Rutschman, Gary D Windham, Michael Judd, Kristen Pogreba Brown, and Kacey Ernst. Quantifying meaningful adoption of a SARS-CoV-2 exposure notification app on the campus of the university of arizona. February 2021. URL <http://medrxiv.org/lookup/doi/10.1101/2021.02.02.21251022>.
- Pablo Rodríguez, Santiago Graña, Eva Elisa Alvarez-León, Manuela Battaglini, Francisco Javier Darias, Miguel A Hernán, Raquel López, Paloma Llana, Maria Cristina Martín, RadarCovid-Pilot Group, Oriana Ramirez-Rubio, Adriana Romaní, Berta Suárez-Rodríguez, Javier Sánchez-Monedero, Alex Arenas, and Lucas Lacasa. A population-based controlled experiment assessing the epidemiological impact of digital contact tracing. *Nat. Commun.*, 12(1):587, January 2021. URL <http://dx.doi.org/10.1038/s41467-020-20817-6>.
- Marcel Salathe, Christian Althaus, Nanina Anderegg, Daniele Antonioli, Tala Ballouz, Edouard Bugnon, Srdjan Čapkun, Dennis Jackson, Sang-Il Kim, Jim Larus, Nicola Low, Wouter Lueks, Dominik Menges, Cédric Moullet, Mathias Payer, Julien Riou, Theresa Stadler, Carmela Troncoso, Effy Vayena, and Viktor von Wyl. Early evidence of effectiveness of digital contact tracing for SARS-CoV-2 in switzerland. *Swiss Med. Wkly*, 150:w20457, December 2020. URL <http://dx.doi.org/10.4414/smw.2020.20457>.
- Felix Sattler, Jackie Ma, Patrick Wagner, David Neumann, Markus Wenzel, Ralf Schäfer, Wojciech Samek, Klaus-Robert Müller, and Thomas Wiegand. Risk estimation of SARS-CoV-2 transmission from bluetooth low energy measurements. *npj Digital Medicine*, 3(1):129, October 2020. URL <https://doi.org/10.1038/s41746-020-00340-0>.
- Timo Smieszek. A mechanistic model of infection: why duration and intensity of contacts should be included in models of disease spread. *Theor. Biol. Med. Model.*, 6:25, November 2009. URL <http://dx.doi.org/10.1186/1742-4682-6-25>.
- Amanda M Wilson, Nathan Aviles, Paloma I Beamer, Zsombor Szabo, Kacey C Ernst, and Joanna Masel. Quantifying SARS-CoV-2 infection risk within the Apple/Google exposure notification framework to inform quarantine recommendations. *medRxiv*, (medrxiv:2020.07.17.20156539v1), July 2020. URL <https://www.medrxiv.org/content/10.1101/2020.07.17.20156539v1.abstract>.
- Chris Wymant, Luca Ferretti, Daphne Tsallis, Marcos Charalambides, Lucie Abeler-Dörner, David Bonsall, Robert Hinch, Michelle Kendall, Luke Milsom, Matthew Ayres, Chris Holmes, Mark Briers, and Christophe Fraser. The epidemiological impact of the NHS COVID-19 App. 2021. URL https://github.com/BDI-pathogens/covid-19_instant_tracing/blob/master/Epidemiological_Impact_of_the_NHS_COVID_19_App_Public_Release_V1.pdf.