Pre Work - Paper Summary & Exploration Thoughts
Team: Zixi Chen, Marcel Torne Villasevil, Zihe Zhang
Topic: Risk score learning for COVID-19 contact tracing apps
(https://static1.squarespace.com/static/59d5ac1780bd5ef9c396eda6/t/60fb3ae8379556598ce0aab4/1627077353303/mlhc_risk_score_learn_camera.pdf)

Problem Statement

The paper aims to use ML algorithms to automatically learn risk score models' parameters to estimate the risk that a user of digital contact tracing apps was infected during a particular exposure, and further extend it to the case with multiple exposures.

Summary

Model setup:
They quantify and approximate three features

- Duration of the exposure <- bluetooth attenuation
- Distance between the index case and the user
- Infectiousness of the index case <- days since symptom onset

and set up how their piecewise constant approximations together describe the risk of an encounter.
The parameters they need to estimate are

- Attenuation thresholds and the weights for each interval for the piecewise constant approximation of bluetooth attenuation
- Attenuation thresholds and the weights for each interval for the piecewise constant approximation of days since symptom onset

They also set up censoring simulations with randomness, as it's possible that a user can get exposed to events that are not recorded.

Optimization:
They try to find the parameters that maximize the log-likelihood of the observed data in the following manner:

- Monotonicity: They ensure that the risk score and the infectiousness level are monotonically increasing in attenuation by ordering the attenuation buckets from low risk to high and using projected gradient descent
- Soft thresholding: Since the loss function is not differentiable with respect to the attenuation thresholds, they either
    - Use a gradient-free optimizer such as grid search for the thresholds in the outer loop and a gradient-based optimizer for the weights in the inner loop
    - Or soft thresholding with logistic sigmoid function, with t being the temperature parameter that would gradually increase, and use a gradient-based optimizer on both thresholds and weights

Experiments:

They simulate a pool of exposure events and assign random bag of events to each user, and then consider two scenarios for “positive bags”

- Every positive bag has only one positive exposure (harder learning)
- Every positive bag has multiple positive exposures

and then perform a train-test split and 1000 iterations of mini-batch gradient descent.
They compare the performance of the learned parameters from ML to the performance using true simulator probabilities and using the Swiss version of the exposure notification app.

Bonus Steps

[failure mode/s] Can we identify scenarios in which the learning methods lose/maintain power in estimation or robustness to model mismatch, for instance when using more realistic censoring simulations / having bluetooth noise / increasing model capacity for risk score computing?

[online learning] How well does the model perform if we do online learning? We could try to split the training set in multiple batches and feed it one by one to the model simulating that time is passing and we are collecting new datasets. We could also try to have batches from datasets generated by different parameters and verify again how well our model adapts, this would simulate new COVID-19 variants for example.

[new optimization methods] Since we can verify how well our models performed, can we train some RL algorithm / GA algorithm and see how well it generalizes and how well it performs? Just to see the comparison.