Decision Analytics for Business and Policy

Project: Analytic Modeling of Epidemic Spread

Due Dates: Final Report due May 2 at 11:59pm **Submission:** Canvas

Project Scope.

You've learned to solve structured, clean problems in Python with Gurobi in assignments. Now, we focus on a more general optimization framework, where the objective function and constraints do not have explicit functional forms. Instead, you can only query them via blackbox evaluations. In this case, the blackbox is an SIR simulation model.

Part 1. How did the basic reproduction number, R_0, change over time through the pandemic? Answer this question for 5 regions (5 counties, or 5 states, or 5 countries).

The goal of this part is to use the SIR model with time-varying R_0^τ to fit case count as closely as possible. If you work at the state or county level, we recommend the New York Times COVID data repository (https://github.com/nytimes/covid-19-data). You can complement your study with other data sources as you see fit. The following breakdown can help you organize this part.

1. Pick one region, and download the case count data over time (e.g., daily case count from January 2020 to March 2022). Set each time period to be either a day or a week. Let T be the overall time horizon. Let I_t be the number of cases for each time period $t \in T$. Let's call $\{I_t\}_{t\in T}$ the *true* outcome.

2. Design a dynamic SIR simulation model. Here you will have to decide the number of different R_0^τ's to include over the entire epidemic process. Let's call each τ an *epoch*. The duration of each epoch can vary. In addition, you should experiment with different numbers of epochs to see what works best for this project: the fewer epochs you use, the larger the error, but it becomes more computationally tractable, and also the results are more "generalizable". (This is similar to the concept of regularization in machine learning, which trades off in-sample accuracy and out-of-sample generalizability by adjusting model complexity.) Let's call the simulated case count our *simulated* outcome, $\hat{I}_t$, for $t \in T$.

3. Define two formulae to calculate the *error* of your simulation model. The first one is the root mean squared error between the true case counts and the simulated case counts: $e_{\text{rms}}(\{I_t\}, \{\hat{I}_t\}) := \sqrt{\frac{1}{|T|}\sum_{t\in T}(I_t - \hat{I}_t)^2}$. The second one is the mean absolute error: $e_{\text{abs}}(\{I_t\}, \{\hat{I}_t\}) := \sum_{t\in T}\frac{1}{|T|}|I_t - \hat{I}_t|$.

4. Find R_0^τ values to minimize e_{rms}. Find another set of R_0^τ values to minimize e_{abs}. This step is going to be the most involved step in Part 1. You can do so manually by trial and error, or use more sophisticated methods. Successfully applying a sophisticated method (such as either one mentioned here) will get you 5 bonus points, partial credit is possible (this project is worth 30 points in total). You can pick at most one method from below. If you choose to do this bonus, you can set up meetings with me for additional guidance.

   - **Simulation → Machine Learning → Optimization**. Train a machine learning model (e.g., polynomial regression with degree 1 or 2) between the features (i.e., R_0^τ values) and the label (i.e., the error). Then use the trained model to find the values of R_0^τ that minimize error. If you choose polynomial regression with degree 2, then this minimization step corresponds to a quadratic optimization model, which you may be able to solve in Python and Gurobi. Make sure to specify reasonable upper and lower bounds for each R_0^τ in the optimization step. You may also need to manually perturb the resulting R_0^τ values to improve results, since the trained regression model only gives an approximate relationship between $\{R_0^\tau\}$ and error.

- **Gradient Descent with Estimated Gradients**. Directly perform gradient descent on R_0^τ to find the optimal values to minimize error. Again, define reasonable upper and lower bounds for R_0^τ. In each step of the descent algorithm, if any value of R_0^τ is out of bounds, "project" it back to the closest feasible value. Since gradient can not be calculated analytically, you'll have to resort to approximation by running the simulation many times each time you need a gradient.

5. Once you are done with this region, repeat for all 5 regions. Report the final $\{R_0^\tau\}$ values, and show the actual case counts versus simulated case counts over time for each region.

So overall, Part 1 asks you to solve these two optimization problems

$$\min_{\{R_0^\tau\}} e_{\text{rms}}(\{I_t\}, \{\hat{I}_t\}), \text{ where } \underline{R}_0^\tau \le R_0^\tau \le \overline{R}_0^\tau \text{ for all } \tau, \text{ and } \{\hat{I}_t\} \text{ are determined by } \{R_0^\tau\},$$

$$\min_{\{R_0^\tau\}} e_{\text{abs}}(\{I_t\}, \{\hat{I}_t\}), \text{ where } \underline{R}_0^\tau \le R_0^\tau \le \overline{R}_0^\tau \text{ for all } \tau, \text{ and } \{\hat{I}_t\} \text{ are determined by } \{R_0^\tau\}.$$

Part 2. Propose your own question related to the pandemic. Answer it with the analysis from Part 1, and potentially other information and data sources.

Sample Question. How did the infectiousness and fatality rate of the virus change over time? To understand the infectiousness change, you would have to tease out the following time-varying effects from R_0: mobility level, mask and social distancing behavior, vaccination percentage (and vaccine effectiveness), and so on. To calculate the virus fatality rate change, you can compare the case count and death count over time, and also incorporate information about the quality of care over time. Note that the New York Times COVID data includes information about masks and death counts. You may find data about mobility level from SafeGraph (https://www.safegraph.com, free academic access). Some vaccination and vaccine effectiveness data are available online, for example from the CDC website: https://data.cdc.gov/browse?category=Vaccinations. To visualize the final results, you can use a scatter plot with "infectiousness" on the horizontal axis, and "fatality rate" on the vertical axis, and plot how the virus drifted in this space over time.

The question you propose should be approximately at this level of complexity, otherwise we will ask you to revise (see timeline below). You can choose to use this sample question directly too.

Timeline.

- March 28, 2022 – Project release. Form your own teams, 4 to 6 people each. If you have trouble finding a team, let us know immediately.
- April 6, 2022 - Email your team's question for Part 2 to the TA and instructor.
- April 11, 2022 - Feedback on the Part 2 question (if no feedback, you can directly proceed).
- April 25 & 27, 2022 - Presentations.
- May 2, 2022 - Final report due.

Deliverables. Details for the final deliverable (per team):

1. Final report:
   - The report should be no more than 10 pages.
   - The report should include problem statement, analytical formulations, data summary, implementation details, and analysis.
   - The report should be written for intelligent readers who are not necessarily familiar with the tools we learned in this course.
   - NOTE: Your final report should include a cover page (in addition to the 10 pages) that outlines all resources you used and contribution from each team member. It is expected that every team member contributes equally (in time and energy). Any potential imbalance should be discussed with the instructor as soon as possible.
2. Code
   - If you have multiple files, include a "ReadMe" file to explain which files should be executed and in what order, and/or the structure of the source/data file directories.
   - Your files should be self-standing and executable on other computers (e.g., do not hard-copy your local address in your code, use relative addresses).
3. Data
   - Include a "DataDictionary" file to describe where and how you found the data. You can include web links in this file if you used online resources.
4. Presentation
   - 10-20 min for each team, depending on how many teams we have eventually.
   - Organize your presentation flow in the same way you organize the final report. Beware of the time constraints, and put emphasis on clearly describing what problem you are trying to solve, what assumptions you are making, data sources, and the results.

Grading. The project accounts for 30% of course grade. Report (along with code and data description): 25 points. Presentation: 5 points.

For the report, Part 1 analysis accounts for 10 points, Part 2 analysis accounts for 10 points, and the overall report quality accounts for 5 points. (Potential bonus: +5 points.)