PPL Investigation Lessons Learned

Data Science Institute at University of Wisconsin-Madison.
Steve Goldstein, Marlin Lee, Wansoo Cho, Abe Megahed,
May 2022

Probabilistic programming is a paradigm that has begun to receive increased attention in the data science world over the past few years. We began an investigation into PPL in order to gain some experience with the tools, the algorithms, and the processes involved. In the process of this investigation, we applied the tools to a sample dataset from Covid wastewater analysis. The project goals, objectives, and conclusions are discussed below.

What is PPL?

Wikipedia defines probabilistic programming as follows:

Probabilistic programming (PP) is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.
https://en.wikipedia.org/wiki/Probabilistic_programming

The promise of PPL tools in general is that they make it easier to model processes that are inherently stochastic or have a stochastic element to the measurement or data collection aspects.

Using dedicated PPL tools such as Gen, the notions of models, inference, and probability distributions are explicitly defined in the language and the framework, making these concepts easier to understand.

Reasons for this Investigation

1. PPL is relatively new.

To begin with, the notion of probabilistic programming and dedicated probabilistic programming languages are relatively new and not widely understood. One of the oldest PPL languages, Stan was released in 2012. PyMC was released a year later in 2013. The Gen PPL system was released in 2019. These systems have not been around long enough or adopted widely enough to have established a wide base of experience and understanding.

2. Understand how to apply PPL to existing projects.

Another reason to understand probabilistic programming techniques is because we have ongoing projects that could possibly benefit from the application of PPL techniques. For example, our project with the Wisconsin Department of Health Services involves analysis of Covid19 wastewater data that could benefit from improved modeling and prediction techniques.

3. Understand how PPL could be applied to future projects.

Lastly, understanding PPL techniques provides us with another tool in our tool belt that we can bring to bear in implementing or proposing future projects.

Goals

When initiating this project, we had the following high level goals in mind:

- Understand PPL concepts at a high level.
- Understand PPL algorithms.
- Get hands on experience using a PPL language.
- Apply our experience to a sample data set.
- Derive conclusions about PPL from our experience.

Objectives

4. Identify a PPL tool to use

We began the investigation by performing a survey of the available PPL tools and languages. Tools that we examined included the following:

- Gen / Julia
- Turing.jl
- PyMC
- NumPyro
- TensorFlow
- Stan

Based upon examination of the tools and documentation and initial experience trying to install and use these tools, we decided to focus on the Gen / Julia toolset for the following reasons:

- It was relatively easy to install and use.
- It is well documented.
- It has a large and active developer community.
- The Julia language provides a flexible syntax that makes Gen framework concepts easier to manage (macros for models, addresses for traces and distributions etc.)

5. Identify a problem space to explore using PPL

When starting this investigation, we also needed to determine how we would apply PPL tools. The problem spaces that we considered were the following:

1. Covid19 wastewater data analysis
2. Chemical reaction networks

We decided to focus on the Covid wastewater data for the following reasons:

- The problem is simpler and easier to understand

- Covid WW data is available though an ongoing project with Wisconsin DHS (Department of Health Services).
- We have prior experience working with the Covid WW data.

6. Explore problem space using PPL tool

In order to understand the problem space and the data using PPL, we sought to perform the following tasks:

1. Create a probabilistic model that fits the data.
2. Determine how well the probabilistic model matches the data.
3. Evaluate how the probabilistic model handles outliers in the data.
4. Evaluate how the probabilistic model predicts trends in the data.

7. Make conclusions based upon this experience

Finally, the last objective was to draw some conclusions about the general power and applicability of PPL techniques from our experience working with the particular tool and problem space that we identified.

1. Evaluate well the PPL tools performed in or particular problem domain.
2. Try to extrapolate from this experience how well the tools might function in other problem domains.

Results

1. Basic Import and Display of Data

The first step in the process was simply to work out how to read in and display our dataset. Thankfully, the Julia language provides an ample collection of utilities for accomplishing this task. To perform this task, we used the following Julia modules:

- CSV
- Dataframe
- Plot

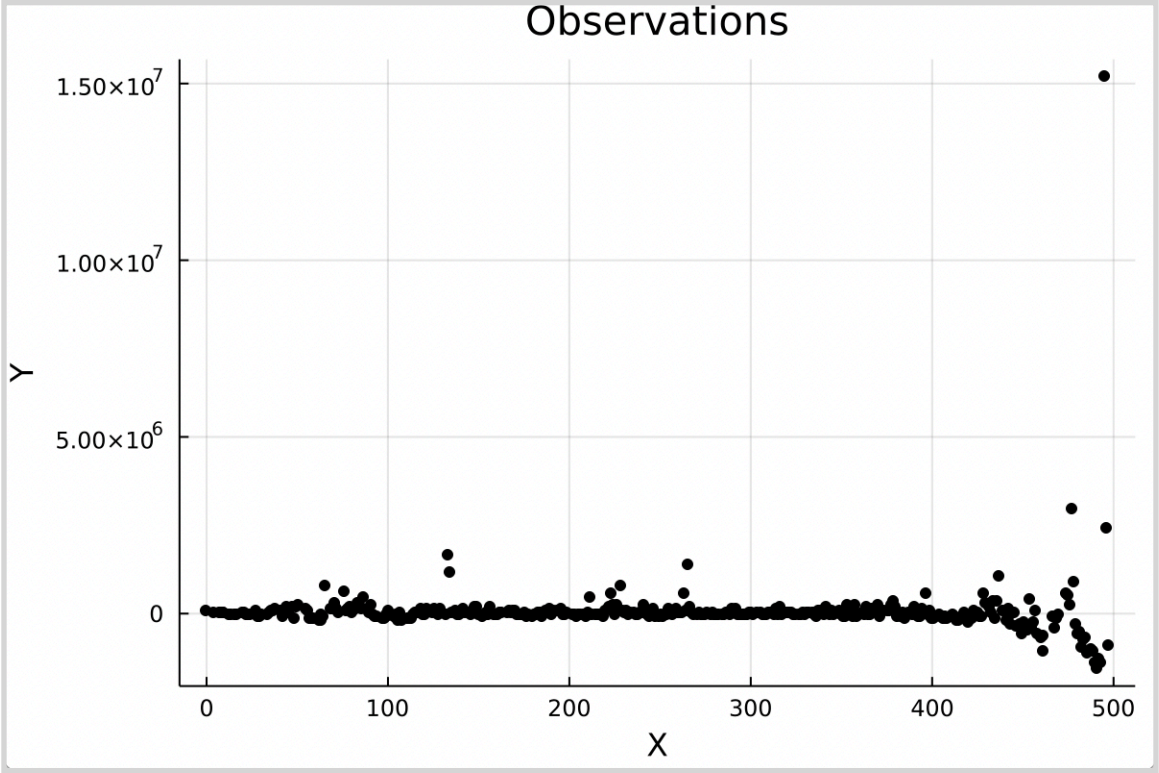

Figure 1: Data Plot

2. Linear Model

Next, we applied the basic linear model described in the Gen tutorial documentation to our data set:

Gen tutorial linear model:
https://www.gen.dev/tutorials/intro-to-modeling/tutorial#writing-model

Using this approach, we were able to fit a line to the data, however, given the fact that the data contains obvious non-linear long term trends as well as short term variations, the fit is not particularly good.

As shown in Figure 2 below, the left hand plot displays a single line through the data points. The range of the graph doesn't fit the data very well due to a few extreme outliers. The graph to the right is the data plotted using a logarithmic scale.

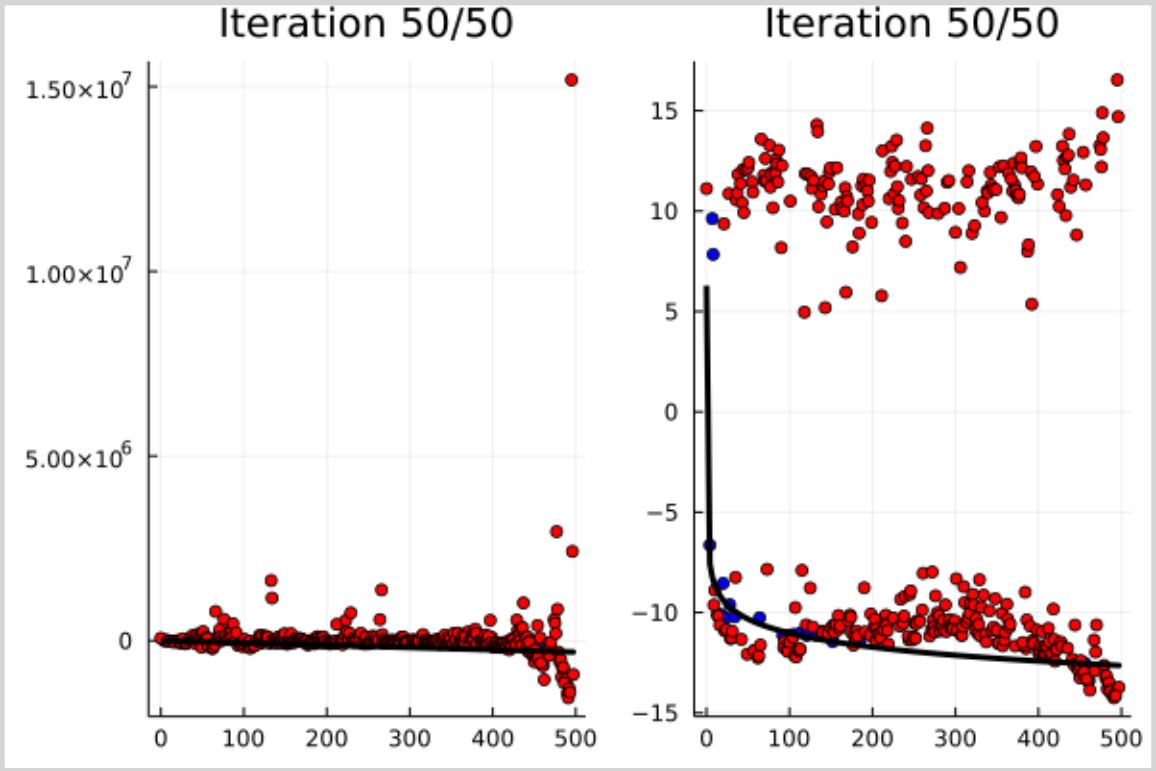

Figure 2: Linear Model

3. Linear Spline Model

To better account for the variations in the data, our next step was to switch from using a single line to using a sequence of line segments. As you would expect, this does a better job of modeling the variations in the data.

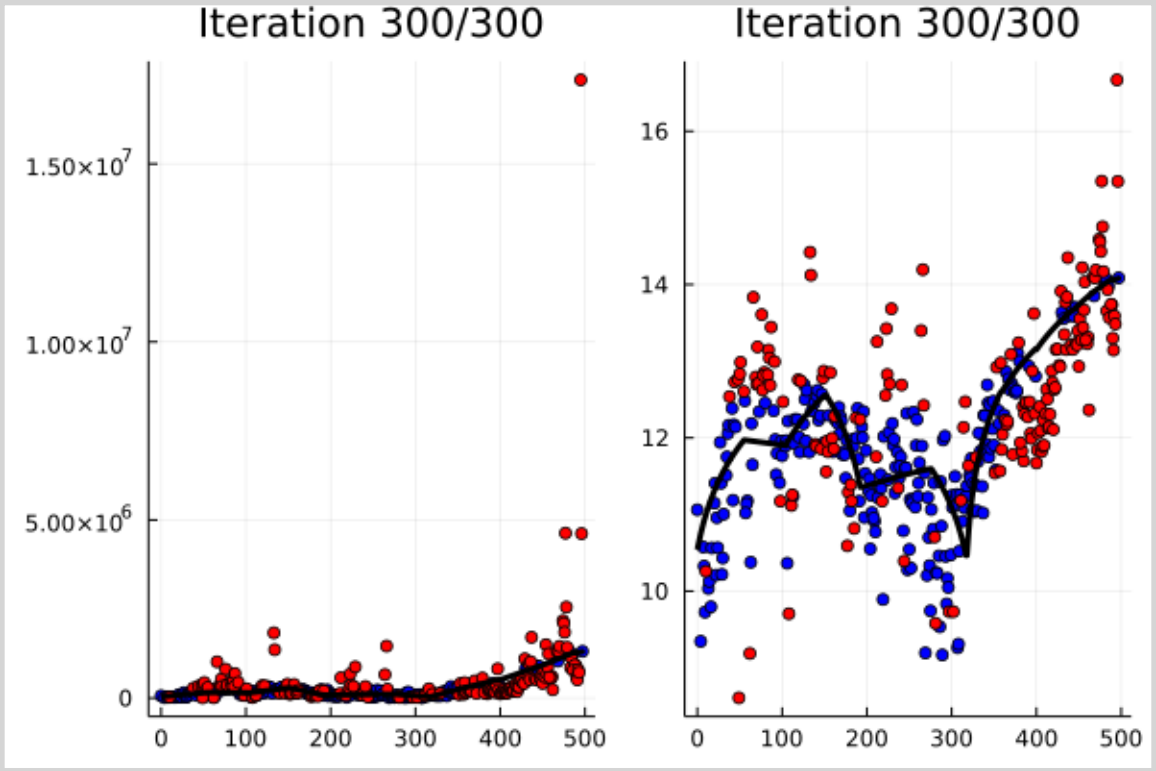

Figure 3: Linear Spline Model

4. Linear Log Spline Model

The next step to modeling the data was based on the observation that epidemiological trends tend to follow an exponential curve. When plotted in a logarithmic graph, these exponential trends become linear trends. Therefore, since our model is linear, it makes sense to covert the data to a logarithmic space first, before applying the linear model. When we do this, we arrive at a linear spline in logarithmic space which maps to a piecewise series of exponential curves in linear space.

5. Quadratic Spline Model

Lastly, we sought to overcome the limitations of the linear model by substituting a quadratic model which can better account for non-linear trends in the data.

Conclusions

1. Benefits of PPL tools

During our investigation, we found the following benefits of the PPL technique:

1. **Focus on the model**
   One significant advantage of the PPL technique is that it allows the investigator to focus solely on the model rather than on the mechanics of fitting the model to the data. By having a clear separation between the model and the inference code, it allows the investigator to think about the model in isolation and to ignore the details of how the inference process works.

2. **Expressiveness in programming**
   Another advantage of PPL programming is that the model code tends to be quite succinct and expressive. This makes the model easier to understand.

3. **Good at expressing inherently stochastic processes**
   One last possible advantage of the probabilistic approach is that some inherently stochastic elements such as the existence of outliers are handled in a probabilistic way. If we were to take the simple accept / reject approach to modeling outliers, then it would always throw away results that deviate from the norm. If we know that there is a non-zero probability of data with a large variation, then a more probabilistic approach having a non-zero probability of including these results may be more appropriate.

4. Limitations of PPL tools

Unfortunately, no tool is perfect and we also found the following limitations associated with the PPL technique:

1. **Model sometimes requires arbitrary parameters / heuristics**.
   When constructing probabilistic models, we need to sometimes include heuristics that are determined by somewhat arbitrary parameters. For example, when including a probability for outliers, it was necessary to guess at a reasonable approximation

2. **Processor intensive / Long run times.**
   When working with even small datasets and relatively simple models, we encountered run times of several minutes or tens of minutes. We investigated the use of more powerful hardware and GPU acceleration.

3. **Tools can be hard to work with.**
   The fact that the probabilistic models frequently need some tuning to apply to different data sets limits the usefulness of this approach. PPL parameters require the developer to specify values in probabilistic terms which is not always obvious or intuitive. This requires users of the tool to have some understanding of PPL in order to effectively apply the tools.