E414 Model Iteration One

Predicting U.S. Tornado Counts

Author: 118777

Q2 2024

Table of Contents

1 Motivation

As with (likely) most of the other models in this repository, their raison d'être is to facilitate forecasting on Metaculus question 22307, which is part of the Bridgewater Forecasting Contest. The resolution criteria for question 22307 reads:

> This question will resolve as the number of tornadoes in the United States in April of 2024, according to the Storm Prediction Center's preliminary tornado summary.
>
> If the resolution value is below the lower bound or above the upper bound, the question resolves outside the respective bound.
>
> The question will resolve as the number of tornadoes shown for the month of April 2024 when accessed by Metaculus on May 3, 2024. If there is a discrepancy between the "Map" and "Tables" views, the question resolves to the figure displayed on the "Map" view.

By the time you are reading this, the question has very likely closed. Also, given that the open upper bound on the question begins at 300 and that the National Oceanic and Atmospheric Administration's (NOAA) Storm Prediction Center reports 310 as the number of preliminary tornadoes across the USA in April 2024, you might be saying to yourself, how can this brief report help me? Well dear forecaster, should a similar question come up on Metaculus, even if it is not necessarily about tornadoes, then the following discussion—on multilevel modelling—might enable you to produce better, more statistically informed Metaculus forecasts. There is additionally an argument to be made that

$$P(\text{Finds The Below Interesting} \mid \text{Participates On Metaculus})$$

is high…but you can always stop reading at any time.

Before we begin, though, the author wants to note that NOAA does indeed permit data downloading; however, this process is quite tedious and annoying! Please commend the author on their patience manually copying the data into text files. The data available for use in modelling includes tornado, wind, and hail counts for each state, by month and year (1950 to 2024). Find the data here. The author downloaded the *preliminary* report values (given that this is the forecasting target for question 22307) for January 2019 to March 2024. Final reports were not downloaded since for many months across many states in the distant past, these report values are still not available.

The author is still maturing mathematically, so (please!) gaze upon the statistical sections with skeptical eyes. Note as well that the author operates by Crocker's Rules, so criticism is welcome. To critique, please make a new issue here. The following was informed in part from Richard McElreath's 13th chapter of Statistical Rethinking [1]

2 Model(s)

Tornado counts vary across state, month, and year. One might imagine that climatic factors change over time (across years), that the formation of tornadoes alters within a year, over different months, and that the rate of tornado formation is also affected by local geographical features (state location).

To adequately incorporate information across these categories, we can create a multilevel model. For each relevant category (`State`, `Month`, `Year`) available to us, we want to estimate its intercept and internal variation (e.g., across each `State`, how does tornado count vary?).

Since we are estimating tornado *counts*, a Poisson distribution seems ap-

propriate.

Proceeding, let α_i denote the random effect of the location of state i on expected tornado count. Further, let γ_j and δ_k denote the random of effects of month j on and year k, respectively, on expected tornado count. Let the expected tornado count for state i, month j, and year k be called Y_{ijk}.

The tornado count T_{ijk} for state i, month j, and year k can be modelled via $T_{ijk} \sim \text{Poisson}(Y_{ijk})$, where $\log(Y_{ijk}) = \alpha_{\text{STATE}[i]} + \gamma_{\text{MONTH}[j]} + \delta_{\text{YEAR}[i]}$, i.e. Y_{ijk} is log-linear.

Note that the author is "uncertain" regarding what constitute "appropriate" priors distributions and parameter values to use for a model on tornadoes. Note also that the author explores these decisions somewhat via a sensitivity analysis. The prior values for the hyperparameters $\overline{\alpha}$, $\overline{\gamma}$, and $\overline{\delta}$ represent the author's best guess that state effects vary more than year effects vary more than month effects, with respect to tornado count.

Adaptive priors:

$$\begin{aligned}
\alpha_l &\sim \text{Normal}(\overline{\alpha}, \sigma_\alpha) \quad \text{for} \quad l = 1, 2, \ldots, 50 \\
\gamma_l &\sim \text{Normal}(\overline{\alpha}, \sigma_\gamma) \quad \text{for} \quad l = 1, 2, \ldots, 12 \\
\delta_l &\sim \text{Normal}(\overline{\alpha}, \sigma_\delta) \quad \text{for} \quad l = 1, 2, \ldots, 5
\end{aligned}$$

where

$$\begin{aligned}
\overline{\alpha} &\sim \text{Normal}(0, 3.0) \\
\overline{\gamma} &\sim \text{Normal}(0, 1.0) \\
\overline{\delta} &\sim \text{Normal}(0, 2.0)
\end{aligned}$$

and

$$\sigma_\alpha, \sigma_\gamma, \sigma_\delta \sim \text{Exponential}(1.0)$$

2.1 Implementation

Listing 1: Model Implementation In Numpyro

```
def model_01(states=None, months=None, years=None, tornadoes=None):

    # states, months, years, and tornadoes represented
    num_states = len(np.unique(states))
    num_months = len(np.unique(months))
    num_years = len(np.unique(years))
    num_tornados = len(tornadoes)

    # state effect hyperparameters
    alpha_mu = npro.sample("alpha_mu", dist.Normal(0, 3.0))
    alpha_sigma = npro.sample("alpha_sigma", dist.Exponential(1.0))

    # state effect
    with npro.plate("states", num_states):
        alpha = npro.sample("alpha", dist.Normal(alpha_mu,
            alpha_sigma))

    # month effect hyperparameters
    gamma_mu = npro.sample("gamma_mu", dist.Normal(0, 1.0))
    gamma_sigma = npro.sample("gamma_sigma", dist.Exponential(1.0))

    # month effect
    with npro.plate("months", num_months):
        gamma = npro.sample("gamma", dist.Normal(gamma_mu,
            gamma_sigma))

    # year effect hyperparameters
    delta_mu = npro.sample("delta_mu", dist.Normal(0, 2.0))
    delta_sigma = npro.sample("delta_sigma", dist.Exponential(1.0))

    # year effect
    with npro.plate("years", num_years):
        delta = npro.sample("delta", dist.Normal(delta_mu,
            delta_sigma))
```

```
    # expected tornadoes
    Y = jnp.exp(alpha[states] + gamma[months] + delta[years])

    # likelihood
    with npro.plate("data", size=num_tornados):
        npro.sample("obs", dist.Poisson(Y), obs=tornadoes)
```

2.2 Inference

2.2.1 Prior

2.2.2 Prior Predictive Check

2.2.3 Posterior Predictive Check

2.2.4 Posterior Samples

2.2.5 Other Results

3 Appendix

3.1 Random Intercepts And Slopes

From the Wikipedia page on multilevel models (linked earlier), accessed 2024-04-20.

> Before conducting a multilevel model analysis, a researcher must decide on several aspects, including which predictors are to be in-

cluded in the analysis, if any. Second, the researcher must decide whether parameter values (i.e., the elements that will be estimated) will be fixed or random.[2][5][4] Fixed parameters are composed of a constant over all the groups, whereas a random parameter has a different value for each of the groups.[4] Additionally, the researcher must decide whether to employ a maximum likelihood estimation or a restricted maximum likelihood estimation type.[2]

Random intercepts model

A random intercepts model is a model in which intercepts are allowed to vary, and therefore, the scores on the dependent variable for each individual observation are predicted by the intercept that varies across groups.[5][8][4] This model assumes that slopes are fixed (the same across different contexts). In addition, this model provides information about intraclass correlations, which are helpful in determining whether multilevel models are required in the first place.[2]

Random slopes model

A random slopes model is a model in which slopes are allowed to vary according to a correlation matrix, and therefore, the slopes are different across grouping variable such as time or individuals. This model assumes that intercepts are fixed (the same across different contexts).[5]

Random intercepts and slopes model

A model that includes both random intercepts and random slopes is likely the most realistic type of model, although it is also the most complex. In this model, both intercepts and slopes are allowed to vary across groups, meaning that they are different in different contexts.[5]

3.2 McElreath Multilevel Modelling

Description from pp. 400 of the 2nd edition of Statistical Rethinking:

These models remember features of each cluster in the data as they learn about all of the clusters. Depending upon the variation among clusters, which is learned from the data as well, the model pools information across clusters. This pooling tends to improve estimates

about each cluster. This improved estimation leads to several, more prag- matic sounding, benefits of the multilevel approach. I mentioned them in Chapter 1. They are worth repeating.

1. Improved estimates for repeat sampling. When more than one observation arises from the same individual, location, or time, then traditional, single-level models either maximally underfit or overfit the data.
2. Improved estimates for imbalance in sampling. When some individuals, locations, or times are sampled more than others, multilevel models automatically cope with differing uncertainty across these clusters. This prevents over-sampled clusters from unfairly dominating inference.
3. Estimates of variation. If our research questions include variation among individu- also or other groups within the data, then multilevel models are a big help, because they model variation explicitly.
4. Avoid averaging, retain variation. Frequently, scholars pre-average some data to construct variables. This can be dangerous, because averaging removes variation, and there are also typically several different ways to perform the averaging. Averaging therefore both manufactures false confidence and introduces arbitrary datatransformations. Multilevel models allow us to preserve the uncertainty and avoid data transformations

Description from pp. 401 of the 2nd edition of Statistical Rethinking:

Rethinking: A model by any other name. Multilevel models go by many different names, and some statisticians use the same names for different specialized variants, while others use them all inter- changeably. The most common synonyms for "multilevel" are hierarchical and mixed effects. The type of parameters that appear in multilevel models are most commonly known as random ef- fects, which itself can mean very different things to different analysts and in different contexts.193 And even the innocent term "level" can mean different things to different people. There's really no cure for this swamp of vocabulary aside from demanding a mathematical or algorithmic definition of the model. Otherwise, there will always be ambiguity

References

[1] R. McElreath, *Statistical rethinking: A bayesian course with examples in r and stan*. Chapman; Hall/CRC, 2018.