Implementing the Brooks et. al. Empirical Bayes Forecasting Model on German Flu Data

Bianca Doone, Joshua Nugent, Kate Hoff Shutta

University of Massachusetts Amherst

Spring 2019

Overview

1. Review of Empirical Bayes Methodology
2. Preliminary Results
3. Implementation Difficulties / Challenges
4. Future Refinements to our Implementation

Review of method: Big idea

Developed by Brooks, Farrow, Hyun, Tibshirani, Rosenfeld (2015)

- Generate a set of possible trajectories (the "prior") via transformations of smoothed previous-year curves
- Weight the possible trajectories by their resemblance to season's data so far
- Use weights to create point and interval predictions for the future

Our work:

- Implement method in R6, using ForecastFramework for interoperability
- Original paper applies method to CDC influenza data; we apply it to German flu data from public SurvStat database (https://survstat.rki.de/)

Review of method: Components of Prior

Collect the following from training seasons:

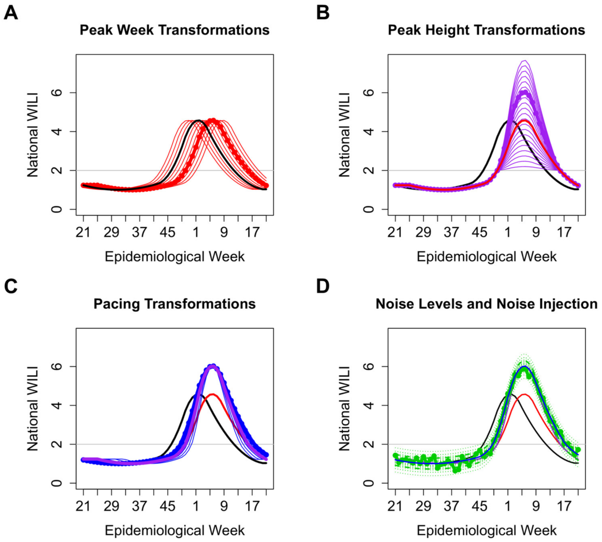
- Smoothed shape using piecewise quadratic filter
- Peak height
- Peak week
- Pace
- Noise

From these values, build candidate trajectories:

- Select a curve at random from previous seasons by drawing from the shapes
- Transform it by drawing randomly from set of peak heights, peak weeks, and pacing
- Associate a noise parameter for weighting during the prediction phase

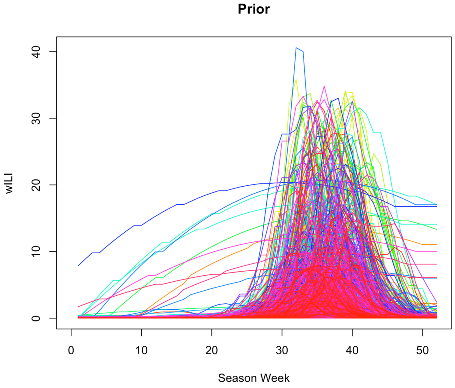
The ‘prior’ is made up of a large set of these possible trajectories.

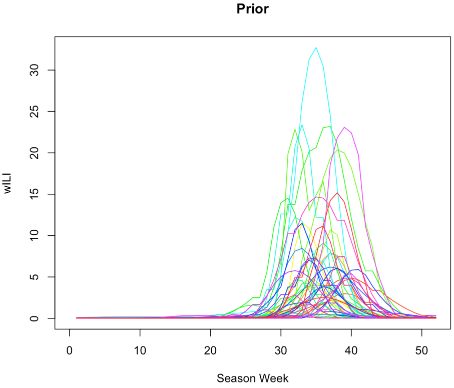
Review of method: Transformations



Brooks et al. (2015), doi.org/10.1371/journal.pcbi.1004382

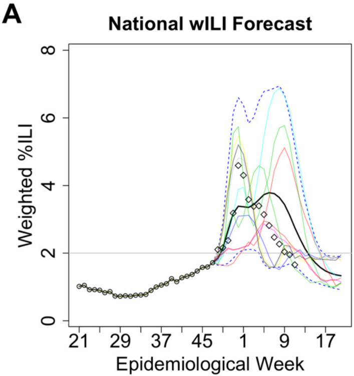
Visualizing the Prior





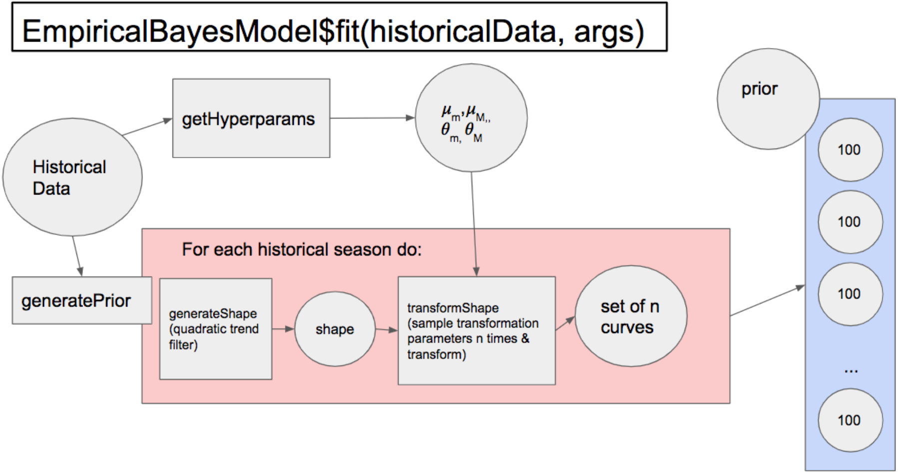
Review of method: Weight curves as data comes in

The weight calculation assumes a normal distribution around each curve and compares the predicted values to the truth, given its associated σ value. Generate a forecast from the weighted future values of the curves.

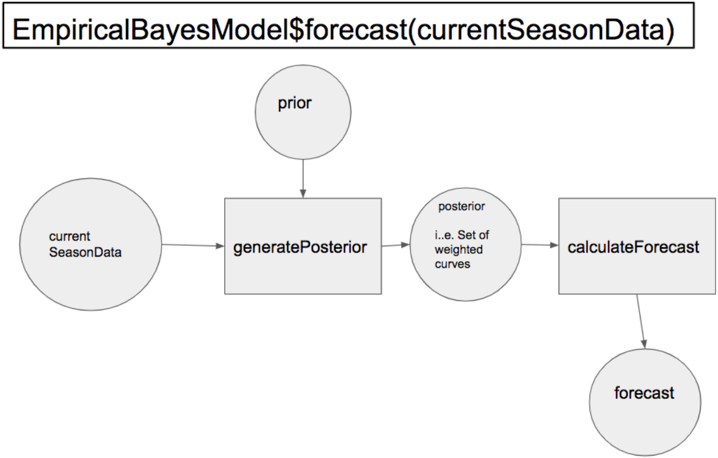


Brooks et al. (2015), doi.org/10.1371/journal.pcbi.1004382

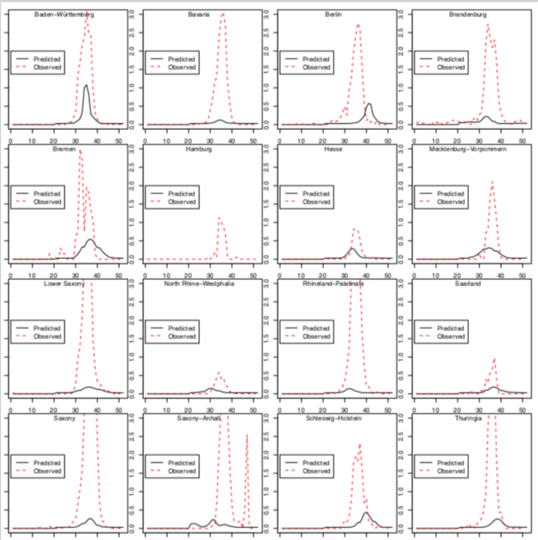
R6 Class Design: Model Fitting



R6 Class Design: Model Forecasting



Forecasts: Initial Results: Not good!



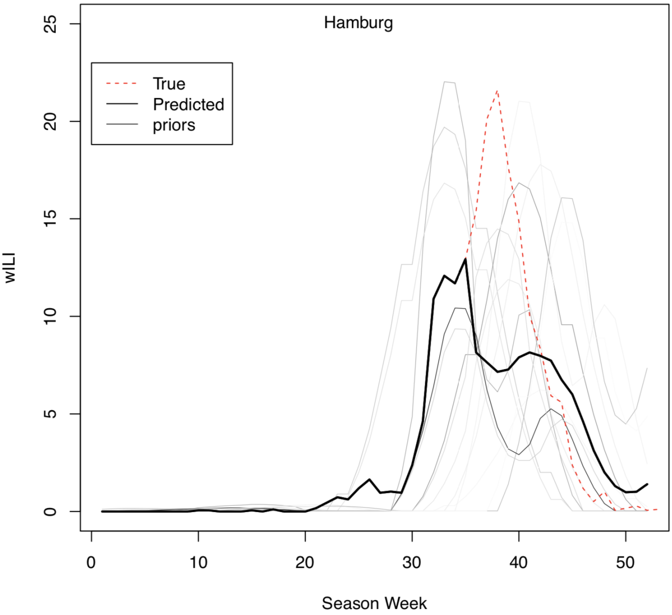
Forecasts: Overfitting, underfitting, time-weighted

Tuning the weighting function was a key step in improving forecasts.

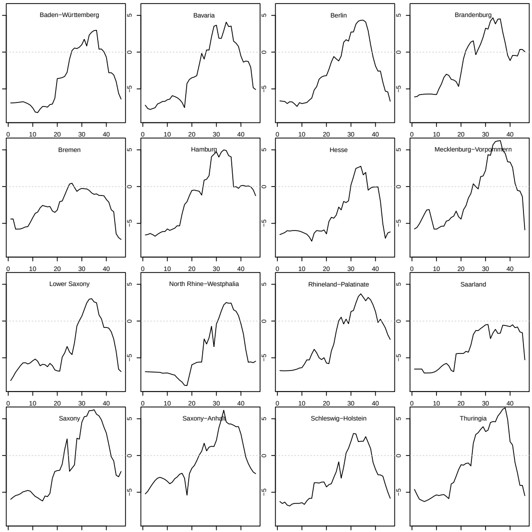
- σ values extracted from training data were too small; caused NaN errors in forecasting process
- Ad hoc solution: Multiply σ by a constant $C \in (4, 10)$. If C too large, forecasts converge to mean of all curves in prior
- Additional experiment: Does down-weighting older data via an exponential decay function improve forecasts?

Choice of these parameters should be informed by a validation process.

After tuning: Better; weights change over time



Forecasts: Test MSE by Region



- log(MSE) for six weeks ahead vs. season-week

Forecasts: Share across regions? Initial experiments

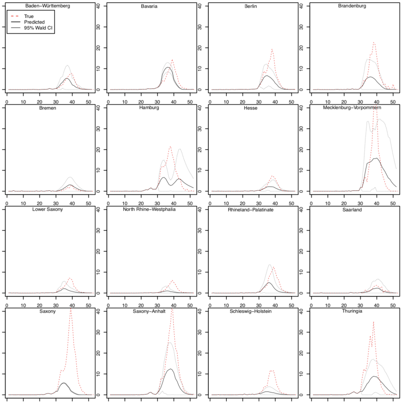


Figure: No sharing across regions

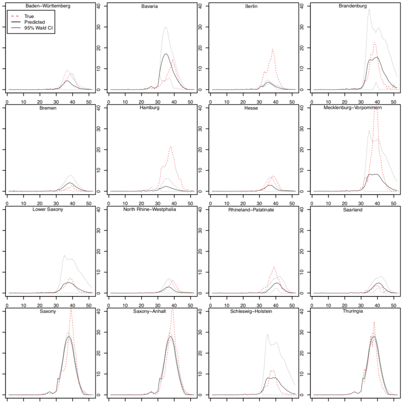
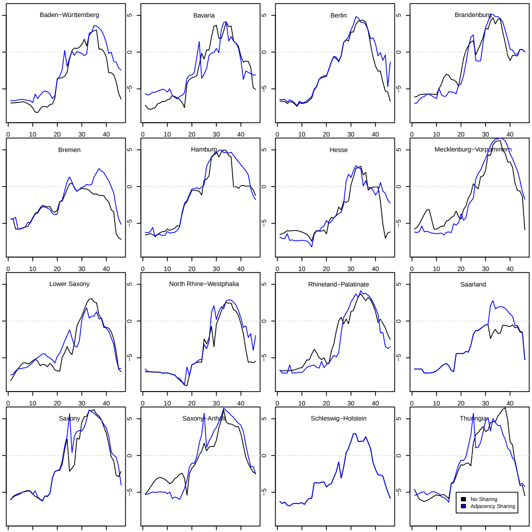


Figure: Sharing across regions

Forecasts: Share across regions? Comparing MSE: Plots



- log(MSE) for six weeks ahead vs. season-week

Forecasts: Share across regions? Comparing MSE: Table

	sig5	sig5_reg	sig100
Baden-Württemberg	1.87	3.67	1.24
Bavaria	5.51	3.77	2.15
Berlin	8.55	11.07	9.22
Brandenburg	10.91	19.55	22.14
Bremen	0.26	1.33	1.74
Hamburg	18.32	25.18	25.25
Hesse	1.37	1.54	1.63
Mecklenburg-Vorpommern	48.63	98.57	56.15
Lower Saxony	1.98	0.53	2.78
North Rhine-Westphalia	1.36	2.36	1.68
Rhineland-Palatinate	4.22	7.26	5.13
Saarland	0.17	1.50	0.49
Saxony	57.19	53.07	40.86
Saxony-Anhalt	24.03	62.54	25.77
Schleswig-Holstein	2.03	2.03	3.56
Thuringia	51.13	23.25	23.31
MMMSE	14.85	19.83	13.94

Implementation Difficulties / Challenges

- How closely to hew to the model used by the original research group vs. make our own modifications
- How to choose the noise parameter? QTF generated very small σ values
- Unclear from paper how prediction intervals were generated; we made our own decisions
- How much should regions be segregated vs. sharing information, in the prior-generating process and/or the forecasting?
- How to handle early-season predictions when so many values are close to zero and weights just latch on to noise?
- QTF did only very mild smoothing. Would other trend-extraction methods work better?
- Optimal value of candidate trajectories to include in the prior?
- Difficult to compartmentalize parts of model and test without re-compiling and re-fitting; time-consuming process

What would we do with more time?

Build an automated CV-based testing system to:

- Try different QTF filtering levels to figure out optimal smoothness parameter and σ value generated
- Experiment with the weighting function to give more recent weeks higher weights (we did this to some extent, but it could be developed further - what time-weighting function is optimal?)
- Test pooling and/or partial pooling of prior values across regions
- Develop a more rigorous (and practically relevant) way of measuring uncertainty

References

Brooks LC, Farrow DC, Hyun S, Tibshirani RJ, Rosenfeld R (2015) Flexible Modeling of Epidemics with an Empirical Bayes Framework. PLOS Computational Biology 11(8): e1004382. https://doi.org/10.1371/journal.pcbi.1004382

Joshua Kaminsky, Justin Lessler and Nicholas Reich (2019). ForecastFramework: A Basis for Modular Model Creation. R package version 0.10.2. https://CRAN.R-project.org/package=ForecastFramework

German flu data collected from SurvStat@RKI 2.0, Robert Koch Institute, https://survstat.rki.de/