Meyers Lab Report #1

Advay Vyas

September 2, 2025

Contents

1 Forecast evaluation background

I plan to read [2] in its entirety and take notes to learn how forecast evaluation works.

1.1 Introduction

ML metrics are very different than forecasting metrics because time series data is much messier and the regular ways of determining model success fail to measure accurately. Forecast origin is self-explanatory and forecast horizon is the section of time that we are predicting upon. Fixed origin evaluation uses the same training data each iteration and the forecasts are computed "one-step ahead". On the other hand, rolling origin evaluation incorporates the new data into the testing set first, and then into the training set on the next iteration. In my opinion, rolling origin seems a lot better and I think that is what we use – not sure yet though.

Time series data also often has a series of issues that make predicting and measuring predictions much harder. Time series can have non-stationarities like seasonality, trends, or breaks; non-normality like fat tails and outliers; and series with a very short history which are inherently hard to use as training data.

1.2 Benchmarks and exaggerating success

Next, the paper moves on to the topic of benchmarks. The naive forecast (a.k.a no-change model) uses the last known observation as the forecast and actually has good performance in some scenarios (likely those with little to no change period to period, not for us with the prevalance of vaccine skepticism). While not a viable model for real prediction, it should definitely be used a strong benchmark to check the performance of other models. Rather than an abstract metric, performing better than the most simple prediction should be the bare minimum for an accurate model. For example, finance models sometimes involve heavy computations like neural networks only to fail against the naive forecast (random walk w/out drift). Essentially, ML research fails to account to be better than simple random modeling – benchmarks are useful!

In more complex scenarios like "clear seasonal patterns", a seasonal naive model should be used - makes sense. Even though that seems obvious, researchers often compare to non-seasonal benchmarks and can show great results. To add more to this discussion, papers often use "overkill" ways of forecasting for no reason - the paper here mentions a modeling a 2-dimensional linear relationship with a neural network (page 801).

1.3 Evaluation metrics, plots, and data leakage

Metrics like MSE, RMSE, MAE are often used with many different time series yet it is very important to watch out for each time series having a different scale. Some (like myself last semester) have used the R^2 value to determine how well a model predicts (especially in random walks, while mine was definitely not random). MAPE is also another metric - luckily for us, we are going to be using a weighted interval score (next section!).

Forecasting plots are also another area to watch out for because they can be quite confusing when comparing different models. For example, watching fit by looking at the "horizontal" shift and finding it to be small instead of the "vertical" shift can give the model the illusion of fitting well. This paper makes the suggestion to only use plots of the forecasts for sanity checks and not for real use. I disagree and I think that those horizontal shifts imply that the model can predict spikes and dips after a delay (as long as it's about always the same delay) and that plots serve a very useful purpose of pinpointing far-off predictions without requiring brute force checking.

Lastly, we tackle the topic of data leakage (usage of the test/unseen data during the training process). In forecasting, since it involves time, it is often to hard to keep track of data seperation during rolling origin prediction. While that is unavoidable, indirect forms of data leakage are much more common. For example, in forecasting, smoothing, decomposition, or normalization over the series before prediction can indirectly help the model predict where the missing data will fit into the distribution. With data leakage, models can often easily outperform any existing or new frameworks, causing issues in result accuracy. This problem also arises when using multiple time series at once, where one series could help the model predict the entire outcome. In conclusion, the most important way to avoid data leakage are during preprocessing, feature extraction, and making sure that one series doesn't reveal the future of another series.

1.4 Guidelines and best practices

The paper states that the forecast model construction plus evaluation usually contains these steps:

- Data forecasting
- Forecasting

- Error calculation
- Error measure calculation
- Statistical tests for significance

2 Weighted interval score metric

I plan to read [1], and then summarize the results and its relevance to our current research focus in this section.

2.1 Evaluating epidemic forecasts in intervals

2.2 Relevance

3 Code investigation

I plan to read through the Flusion and Local-Level-Forecasting codebases and write about what I noticed and questions I have.

3.1 Flusion

TO DO

3.2 Local-Level-Forecasting

TO DO
GBM_US_NSSP_public_state_pct.ipynb

4 Miscellaneous

I'll just store ideas I thought were interesting in this section and investigate them.

4.1 Taylor polynomials for forecasting

TO DO

References

[1] Johannes Bracher et al. "Evaluating epidemic forecasts in an interval format". In: *PLOS Computational Biology* 17.2 (Feb. 2021), pp. 1–15. DOI: 10.1371/journal.pcbi.1008618.

[2] Hansika Hewamalage, Klaus Ackermann, and Christoph Bergmeir. "Forecast evaluation for data scientists: common pitfalls and best practices". In: *Data Mining and Knowledge Discovery* 37.2 (2023), pp. 788–832.