UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT

May 2022

UNCTAD Research Paper No. 83
UNCTAD/SER.RP/2022/3

Benchmarking Econometric and Machine Learning Methodologies in Nowcasting

Daniel Hopp
Associate Statistician
Division on Globalisation and Development Strategies, UNCTAD
daniel.hopp@unctad.org

Abstract

Nowcasting can play a key role in giving policymakers timelier insight to data published with a significant time lag, such as final GDP figures. Currently, there are a plethora of methodologies and approaches for practitioners to choose from. However, there lacks a comprehensive comparison of these disparate approaches in terms of predictive performance and characteristics. This paper addresses that deficiency by examining the performance of 12 different methodologies in nowcasting US quarterly GDP growth, including all the methods most commonly employed in nowcasting, as well as some of the most popular traditional machine learning approaches. Performance was assessed on three different tumultuous periods in US economic history: the early 1980s recession, the 2008 financial crisis, and the COVID crisis. The two best performing methodologies in the analysis were long short-term memory artificial neural networks (LSTM) and Bayesian vector autoregression (BVAR). To facilitate further application and testing of each of the examined methodologies, an open-source repository containing boilerplate code that can be applied to different datasets is published alongside the paper, available at: github.com/dhopp1/nowcasting_benchmark.

Key words: Economic forecast, Machine learning, GDP, LSTM, Bayesian VAR

Table of Contents

Acknowledgements

I would like to thank Anu Peltola for her valuable comments and feedback.

1. Introduction

Gross domestic product's (GDP) importance in quantifying the size and performance of the economy cannot be understated. It is the go-to metric for government officials, policymakers, and indeed even the general public for insight to economic health, and by extension a myriad of related measures, such as social well-being (Dynan et al., 2018; IMF, 2020; Kapoor and Debroy, 2019). However, as much as we depend on this figure, the reality is that it is often published with a significant lag. This lag depends on the country, but in the United States, advanced estimates for an elapsed quarter are not released for at least one month afterwards, with final estimates not appearing until three months afterwards (Federal Reserve Bank of San Francisco, 2005). This delay is related to the complexity of calculating GDP relative to other indicators, with its myriad of sources and adjustments. Policymakers can turn to other, faster-publishing indicators for a quicker look into the state of the economy, such as prices or industrial production, but GDP's comprehensive nature, that which simultaneously delays its publication, is precisely what makes it desirable as an indicator.

Given these characteristics, the utility of applying nowcasting to the case of GDP becomes clear. Nowcasting "is the estimation of the current, or near to it either forwards or backwards in time, state of a target variable using information that is available in a timelier manner" (Hopp, 2021a). In essence, series that are available in a timelier manner than GDP can be used to estimate a model using historical data, when data for both GDP and these independent series are available. This model can then be used to obtain estimates for GDP well before advanced estimates are available, even while the quarter for prediction is ongoing. GDP's publication lag and salience as an indicator have rendered it perhaps the most common target variable for nowcasting applications. This makes it an ideal choice for a benchmark analysis comparing different nowcasting approaches.

The need for such an analysis stems from the existence of several disparate methodologies specifically employed for nowcasting, in addition to other commonly used econometric and machine learning techniques. A new practitioner, or an experienced practitioner looking to improve their models or expand to different datasets, is currently hard-pressed to find a starting point or an overview of nowcasting approaches when making their modelling decisions. The need to refer to several papers to get a sense of different methodologies' performance and characteristics, which may be applied to different datasets, can obfuscate conclusions. It is thus the goal of this paper to consolidate results of the most common statistical, econometric, and machine learning methodologies in nowcasting using perhaps the closest thing to a benchmark dataset available in the field; nowcasting quarterly US GDP growth using explanatory variables from the Federal Reserve of Economic Data (FRED) as specified in Bok et al. (2018). This dataset has the additional benefit of a long publication history, dating back to 1947. This allows performance to be tested on three separate periods in American history with exceptional economic circumstances: the early 1980s recession, the 2008 financial crisis, and the COVID crisis. These three periods are used to test the performance of 12 different methodologies in nowcasting GDP growth. Detailed information on each is provided in section 2.2.

In alphabetical order, they are:

- autoregressive-moving-average (ARMA)
- Bayesian mixed-frequency vector autoregression (Bayesian VAR)
- decision trees
- dynamic factor models (DFM)
- gradient boosted trees
- long short-term memory artificial neural networks (LSTM)
- mixed data sampling regression (MIDAS)
- mixed-frequency vector autoregression (MF-VAR)
- multilayer perceptron feedforward artificial neural networks (MLP)
- ordinary least squares regression (OLS)
- random forest
- ridge regression

The primary goal of this paper is not only to shed light on these methods' relative performance and characteristics in nowcasting, but also to enable practitioners to take these findings and apply them to their own data. Consequently, an accompanying open-source repository has been created where boilerplate code in Python or R for each methodology can be found and easily adapted to different datasets (Hopp, 2022). With this tool, the barrier to trying out nowcasting on different applications may be lowered, as well as the barrier to trying out multiple methodologies to validate results and increase the chances of obtaining a well-performing model.

The rest of the paper will proceed in the following manner: section two will provide further background on nowcasting and the methodologies employed; section three will detail the data used; section four will explain the modelling approach and how results were obtained; section five will display and discuss results; section six will conclude.

2. Background

2.1 Nowcasting

The term nowcasting was first coined and applied in the meteorological domain in the early 1980s to describe weather forecasting of the near future with information on current meteorological conditions (WMO, 2017). It did not begin appearing in economic literature until the mid-2000s, where the term became popularized after the publication of Giannone et al. (2005). The concept of obtaining real-time, data-based estimates of the macroeconomic situation predates 2005, however, with Mariano and Murasawa (2003) developing a coincident business cycle index based on monthly and quarterly series. This application was already very similar to what is considered economic nowcasting today, but did not directly nowcast GDP, rather equating its synthesized business index to "the smoothed estimate of latent monthly real GDP" (Mariano and Murasawa, 2003). Post-2005, a wealth of papers began to be published examining nowcasting different combinations of indicators, most commonly GDP, and geographies. Examples include Portuguese GDP (Morgado et al., 2007), European GDP (Giannone et al., 2009), global trade (Cantú, 2018; Guichard and Rusticelli, 2011), and German GDP (Marcellino and Schumacher, 2010).

A further differentiating axis in the nowcasting literature, and that of primary concern in this paper, is the methodological approach employed. The most commonly-used approach in nowcasting is perhaps the DFM, which is employed in a wealth of papers, including in Giannone et al. (2005). Other examples include Antolin-Diaz et al. (2020), Cantú (2018), and Guichard and Rusticelli (2011), among many others. Other commonly employed approaches include MIDAS (Kuzin et al., 2009; Marcellino and Schumacher, 2010), MF-VAR (Kuzin et al., 2009), Bayesian VAR (Cimadomo et al., 2020), LSTMs (Hopp, 2021a, 2021b), and many more. Each of these methodologies has characteristics which make them suitable for use in the nowcasting context, which comes with its own particular data challenges, discussed below. But Richardson et al. (2021) additionally examined using common machine learning algorithms in predicting New Zealand GDP growth.

It is drawing upon this literature that the 12 methodologies examined in this analysis were chosen: Bayesian VAR, DFM, LSTM, MF-VAR, MIDAS, and MLP were all included as they appear frequently in the nowcasting literature; ARMA was included as a baseline model; OLS and ridge regression were included as perhaps the most popular regression technique in the case of the former and as an augmentation of OLS which could render it more suitable for nowcasting in the case of the latter; decision trees, gradient boosted trees, and random forest were included as three popular machine learning techniques (Sarker, 2021).

As mentioned earlier, nowcasting comes with its own set of data challenges, which each methodology needed to be able to handle. Details of this process for each methodology are outlined in the next section. The first challenge is mixed-frequency data, where all variables in the model are not recorded in the same frequency. In this analysis, for example, a mixture of quarterly and monthly variables was used to estimate a quarterly variable, GDP growth. The second is "ragged-edges", or differences in missing variables at the end of series due to different publication schedules for each. The model needs some way to be able to handle partially complete data at the ends of time series. The third is the "curse of dimensionality", where there may be relatively more input variables to a model compared with training observations (Buono et al., 2017). This can lead to estimation and other errors in some methodologies, such as causing multicollinearity in OLS. If nowcasting is ever to leverage the power of big data and not be restricted to a handful of explanatory variables, this last challenge will be of particular importance.

2.2 Methodologies

The following sections will provide background information as well as references for further reading for each methodology. Due to the quantity of methodologies included in the analysis, it is not possible to include a comprehensive quantitative explanation of each. For those interested, in-depth explanations of this nature are available via the references. The particular programming implementations utilized for each will also be discussed. Methodologies are presented in alphabetical order.

2.2a ARMA

ARMA models are the simplest nowcasting approach examined in this paper, modelling a time series in terms of two main elements: an autoregressive (AR) component, where future values in a series are a function of its own p prior values, and a moving-average (MA) component, where the error terms of the series are a function of q prior error terms. For more information on the use of ARMA models for modelling stationary time series, see Mills (2019). ARMA is a univariate approach for modelling a time series, meaning that, unlike the other 11 methodologies, the ARMA model included only GDP growth's own history as an input variable. This parsimonious nature makes the model an attractive and commonly-used benchmark in nowcasting applications, such as in Cimadomo et al. (2020) or Ministry of Transport, New Zealand Government (2016).

For this analysis, the *auto.arima* function of the *pmdarima* (Smith, 2021) Python library was used to determine the p and q lag orders of the ARMA model on the training set. See section three for more information on the meaning of "training set". The *auto.arima* function determines the lag orders by fitting models with different lag permutations and recording their Akaike Information Criteria (AIC), then selecting the orders which minimize this value. See Liew (2004) for more information on AIC as well as lag selection in ARMA models in general. The *ARIMA* function of *pmdarima* was then used to fit and generate final predictions.

2.2b Bayesian mixed-frequency vector autoregression

Standard vector autoregression (VAR) is similar to the univariate AR model discussed previously, but generalized to consider multiple time series. Whereas in the univariate case, a variable's value is a function of its p prior values, in the multivariate case, a set of variables' prior values are a function of the set's prior values. Essentially, a vector is considered in the modelling rather than a scalar. For more information on VAR models, see (Stock and Watson, 2001).

In contrast to standard VARs, where model parameters are estimated and taken as fixed values, Bayesian VARs consider the parameters as random variables with an assigned prior probability. This approach helps to mitigate over-parameterization, the third data issue discussed in section 2.1. Standard VARs struggle with this issue due to the high number of parameters required to estimate them, usually restricting their use to applications with less than 10 input variables (Bańbura et al., 2010). The introduction of Bayesian shrinkage has been shown to increase forecast accuracy in VAR models with as little as six input variables, many fewer than may be found in a typical nowcasting model. For more information on the concepts of Bayesian shrinkage and Bayesian statistics as applied to regression problems, see De Mol et al. (2008). Bayesian VAR's power in modelling complex dynamic systems and in handling over-parametrization and collinearity have made them a popular choice in the field of nowcasting, see for instance Bańbura et al. (2010), Cimadomo et al. (2020), or Schorfheide and Song (2015).

For this analysis, the *estimate_mfbvar* function of the *mfbvar* (Ankargren et al., 2021) R library was used to estimate and predict on a Bayesian VAR model. A Minnesota prior coupled with the inverse Wishart prior for the form of the error variance-covariance matrix was used, as performed in Cimadomo et al. (2020).

2.2c Decision tree

The decision tree is a commonly used, non-parametric algorithm in machine learning, due in part to its simplicity and interpretability. Decision trees are often employed as part of an ensemble approach, combining many decision trees as weak learners to produce a strong learner. Two of those tree-based ensemble approaches, gradient boosted trees and random forest, will be examined in the coming sections. The basic premise of a decision tree does not differ from the standard semantic interpretation of the term; all data begin as one group at the "root" of the tree and are then split into "branches" at different nodes depending on their characteristics and the information gain from that split. This splitting can be very general, with for instance only a single split, or, at its most extreme, continuing until every observation sits alone on its own "leaf". Decision trees are normally not equipped to handle time series data, see section four for more information on how this was addressed for the decision tree and other methodologies which do not natively handle time series. Simple decision trees have not been used for nowcasting, though tree-based ensembles in nowcasting were examined in Soybilgen and Yazgan (2021) or Tiffin (2016). For more information on decision trees, see Patel and Prajapati (2018) or scikit-learn (2021a).

Decision trees have hyperparameters which usually need to be tuned depending on the application. In machine learning, hyperparameter refers to parameters which determine the macrostructure of a model and not the model's coefficients and parameters themselves. An example with decision trees is the max depth of the tree, or the number of splits the tree can have, which is then a given condition of the structure of the model independent of the data used to train it. Coefficients within the model, i.e., how to split the data, are then determined from the training data the model is fit with. Hyperparameter tuning refers to the process of establishing a performance metric, e.g., mean absolute error (MAE) or root mean square error (RMSE) in a regression application, testing out different hyperparameter combinations, recording their performance according to the performance metric, and selecting a final value for the algorithm's hyperparameters. For more information on hyperparameter tuning, see Probst et al. (2018).

For this analysis, the *DecisionTreeRegressor* function of the *sklearn* (scikit-learn, 2021b) Python library was used. See section four for more information on how hyperparameters were selected.

2.2d Dynamic factor model

Dynamic factor models (DFM) are commonly used in time series forecasting and nowcasting. They operate under the assumption that one or several latent underlying factors explain the development of multiple time series, which are a product of these latent factors plus an idiosyncratic error term. The latent factor could represent, for instance, the business cycle. DFMs are often estimated by assigning variables to various "blocks", which can represent different aspects of the economy. For example, when nowcasting GDP, all variables can be assigned to one global block, while a subset of variables can be assigned to another block, representing, e.g., a geography or economic sector. For more information on the use of blocks in estimating DFMs, see Hallin and Liška (2011).

DFMs are one of the most commonly applied methodologies in nowcasting. Some examples include nowcasting Canadian GDP growth (Chernis and Sekkel, 2017), global trade growth (Cantú, 2018; Guichard and Rusticelli, 2011), Russian GDP growth (Porshakov et al., 2016), and German GDP growth (Marcellino and Schumacher, 2010), among many others. For more information on DFMs, see any of Bok et al. (2018), Cantú (2018), Giannone et al. (2005), or Stock and Watson (2002).

For this analysis, the *dfm* function of the *nowcastDFM* (Hopp and Cantú, 2020) R library was used. This library was developed as an R implementation of the original MATLAB code published alongside Bok et al. (2018). This implementation of the DFM is modelled in state-space form under the assumption that all variables share common latent factors in addition to their own idiosyncratic components. Subsequently, the Kalman filter is applied and parameter estimates are obtained via maximum likelihood estimation. For more information on this particular modelling approach, see Bańbura and Rünstler (2011), Bok et al. (2018), and Cantú (2018). Blocks used were the same specified in Bok et al. (2018). Using a single