MONASH University

Reconciliation Forecasting for Land Transfer Duty

Hoang Do
Nobel Prize, PhD

Report for
Department of Treasury and Finance

5 May 2024

MONASH BUSINESS SCHOOL

Department of Econometrics & Business Statistics

(03) 9905 2478
vdoo0002@student.monash.edu

ABN: 12 377 614 012

Contents

1 Abstract

- Overview: Briefly summarize the purpose, methodology, key findings, and implications of the forecast.
- Key Recommendations: High-level actionable recommendations based on the forecast results.

2 Introduction

2.1 Background:

Overview of LTD and its economic significance.

2.2 Objective:

Purpose of forecasting LTD for DTF. Outline the specific goals of the project, such as predicting revenue from LTD or analyzing trends.

3 Rationale for the Forecast

3.1 Need for Forecasting:

Discuss why forecasting LTD is critical for budgetary planning and economic analysis at DTF.

3.2 Benefits:

How the forecasting results will aid in policy making, financial planning, etc.

4 Data Description

4.1 Data Source:

Detail the sources of the LTD data, including collection methods, frequency, and historical range.

4.2 Variables Description:

Describe each variable used in the analysis, including dependent and independent variables.

5 Initial Data Analysis (IDA)

5.1 Data Preparation and Cleaning:

Since initial data for both aggregate and unit LTD is pre-processed and cleaned by LTD, not much further cleaning required. In this case, only missing values and data format check is required

5.2 Descriptive Statistics:

Basic statistics like mean, median, etc., and initial observations.

5.3 Data Integrity Checks:

Ensuring data completeness, accuracy, and consistency.

6 Exploratory Data Analysis (EDA)

6.1 Visualization:

Use plots (time series plots, histograms, etc.) to visualize trends, cycles, and outliers in LTD data.

6.2 Correlation Analysis:

Identify relationships between LTD and potential predictors.

6.3 Preliminary Findings:

Summarize insights gained about the data and any implications for modeling.

7 Methodology

7.1 Hierarchical time series

Upon analyzing the data characteristics and the disaggregation of LTD, it becomes evident that a three-level hierarchy structure can be established. There is utility in generating forecasts at various levels of aggregation, driven by diverse reasons and objectives. For instance, forecasting solely at the total or top level may lead to inaccuracies due to the limited number of time series encompassed. Additionally, each level of aggregation may exhibit distinct characteristics; for instance, transactions involving residential properties might vary from those involving non-residential properties due to differences in market dynamics or market size for each property type.

In an ideal scenario, forecasts from different levels of aggregation could seamlessly sum up to the top level. However, practical implementation often reveals incoherent among independently produced forecasts. Consequently, it becomes vital for forecasts to align and aggregate according to the hierarchical structure organizing the array of time series. As a solution to this challenge, reconciliation forecasting emerges as one of the most prevalent and effective methodologies employed today.

Within the domain of hierarchical time series forecasting, there are three traditional single level approaches for generating forecasts for hierarchical time series. The first, known as the bottom-up approach, initiates by producing forecasts for each series at the lowest level and subsequently

aggregates these to generate forecasts for the upper levels of the hierarchy. Conversely, the top-down approach starts with a forecast at the highest level, which is then disaggregated to lower levels using predetermined proportions—typically based on historical data distributions. Lastly, the middle-out approach amalgamates elements of both the bottom-up and top-down methods.

7.2 Forecast reconciliation

Recall from the data structure and insights from EDA section, we can construct this hierarchical structure for LTD:

$$\begin{bmatrix} \text{Total}_t \\ \text{Non-residential}_t \\ \text{Residential}_t \\ \text{Commercial}_t \\ \text{Industrial}_t \\ \text{Other}_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{Residential}_t \\ \text{Commercial}_t \\ \text{Industrial}_t \\ \text{Other}_t \end{bmatrix}$$

or in a more compact notation:

$$\text{LTD}_t = \text{Sb}_t,$$

where S represents the summing matrix defining how bottom-level series are aggregated.

As indicated by Hyndman and Athanasopoulos (2021), reconciliation forecasting involves the introduction of a mapping matrix, denoted as **G**, to the base forecast, which is determined by the adopted methodology. This matrix, when multiplied by **SG**, yields a coherent set of forecasts.

However, the traditional single level approaches may have their limitations since only base forecast at one level is used. In response to this, Wickramasuriya et al. (2019) introduced the *MinT* (Minimum Trace) optimal reconciliation methodology, which devises a **G** matrix aimed at minimizing the total forecast variance within the coherent forecast set.

This leads to the need for estimating W_h, the forecast error variance of h-step-ahead base forecasts. There are four simplifying approximations in place that have been shown to work well:

- **OLS**: $\mathbf{W}_h = \text{k}_h\mathbf{I}$
- **Variance Scaling**: $\mathbf{W}_h = \text{k}_h\text{diag}(\hat{\mathbf{W}}_1)$
- **Structural Scaling**: $\mathbf{W}_h = \text{k}_h\mathbf{\Lambda}$
- **MinT Shrinkage**: $\mathbf{W}_h = \text{k}_h\mathbf{W}_1$

7.3 Model Selection

As depicted in , land transfer duty (LTD) exhibits mutual correlations with variables such as sales and the home value index (HVI). It is important to note that these relationships are not unidirectional; factors like sales and HVI may influence LTD, but changes in LTD can reciprocally impact these variables. This dynamic interaction is also observed with economic indicators like inflation and interest rates.

The Vector Autoregression (VAR) model, which predicts future values of multiple time series based on their historical data, is well-suited for capturing these complex interactions. Furthermore, as illustrated in , the detection of cointegration patterns indicating a stable long-term equilibrium relationship among variables such as LTD, sales, and HVI, which necessitates the application of the Vector Error Correction Model (VECM). This highlights the pertinence of employing both VAR and VECM to adequately model these relationships.

One limitation of these models is their reliance on ample data to produce reliable parameters. Monthly data spanning over a decade for LTD has been found to be sufficient. Nevertheless, the subsequent section on cross-validation will elaborate on the precise sizing of the training dataset required to ensure compatibility with the VAR and VECM models.

In addition to VAR and VECM, the Autoregressive Integrated Moving Average (ARIMA) model has also been employed, primarily for purposes of comparison and validation of improvements. The efficacy of this comparison will be assessed through time series cross-validation, utilizing various accuracy metrics such as the Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Mean Absolute Scaled Error (MASE).

8 Time Series Cross-Validation

8.1 Definition and rationale

Time series cross-validation represents a sophisticated adaptation of the conventional training/test set approach for model selection, as stated by Hyndman and Athanasopoulos (2021). This methodology is particularly well-suited to time series data because it exclusively includes observations from periods prior to those being forecasted, distinguishing it from traditional cross-validation techniques.

In the scope of this project, time series cross-validation is employed to rigorously evaluate whether the VAR/VECM or ARIMA models yield more accurate forecasts for this dataset across forecasting horizons ranging from 1 to 12 steps ahead. Additionally, this approach is used to gauge the extent of improvement introduced by the reconciliation process in comparison to the b base forecasts. Another

critical application of time series cross-validation in this context is to generate rolling forecasts over a 12-month period for various time intervals. This is essential for the Department of Treasury and Finance (DTF) to analyze the effects of market dynamics or policy changes on land transfer duty (LTD).

8.2 Forecasting

To create folds for the time series cross-validation procedure, we will use `stretch_tsibble()` function with user defined value for two arguments, `.init` for size of initial training set, and `.step` to define how many steps training sets will roll forward each time.

For example, `stretch_tsibble(.init = 10, .step = 1)` produces series of training sets, where first training set has size of 10 observations and will roll 1 step forward each time, i.e. the size of second set and third set will be 11 and 12, respectively. Each folds will be denoted by a unique `.id` value.

8.2.1 Generating forecast

To enhance forecast accuracy, this methodology extends beyond a cross-sectional hierarchical structure to incorporate a temporal dimension, culminating in cross-temporal reconciliation forecasting. This approach integrates forecasts across different time structures, and its efficacy is evaluated by comparing it with cross-sectional and temporal reconciliation forecasts individually to determine any improvements.

Given the monthly frequency of the data, additional aggregation levels such as bi-monthly, quarterly, four-monthly, semi-annually, and annually are established, with the annual aggregation representing the top level of the temporal hierarchy.

Models are fitted at each cross-sectional level across these varying temporal frequencies. A `for` loop is utilized to generate forecasts and residuals, which are subsequently organized into a matrix for each temporal frequency: monthly (denoted as k1), bi-monthly (k2), quarterly (k3), four-monthly (k4), semi-annually (k6), and annually (k12). These matrices are then collectively stored within a list.

This structured approach facilitates the fitting of different models tailored to each temporal frequency, allowing for adjustments in argument values as necessary. Moreover, if a distinct model is required for various cross-sectional levels, an `if` condition is employed within each temporal frequency loop to ensure this customization. The forecasts and residuals are then allocated to `base` and `res` data structures, respectively.

8.2.1.1 ARIMA The ARIMA model will be implemented across all levels of the temporal hierarchical structure and at every cross-sectional level using the `auto.arima()` function from the `forecast` package. This function efficiently determines the optimal parameters for the autoregressive lag, differencing, and moving average components of the model by automatically adjusting them for the error terms.

Forecasts will be generated up to one year, or 12 months into the future, utilizing the `forecast()` function. This function is designed to produce both the mean point forecasts and the associated residuals, thereby providing a comprehensive output that includes predictions and their accuracy measures.

8.2.1.2 VAR/VECM Fitting VAR and VECM models to complex hierarchical structured data necessitates a more intricate approach, as there is no standard method readily available for this specific application. Additionally, these models require the incorporation of external variables, such as sales and the home value index (HVI). To address these challenges, the chosen strategy involves the creation of two separate user-defined functions for VAR and VECM. These functions are designed to process the input data and output both the mean point forecasts and residuals, mirroring the functionality provided by the `auto.arima()` function. Given the significant differences in the scales of sales, HVI, and LTD data, it is advisable to employ logarithmic transformations prior to model fitting, followed by re-transformation to their original scales. This method will be embedded within both functions to ensure accurate scale representation.

8.2.1.2.1 Vector Autoregression (VAR) Model The VAR fitting function, designated as `var_forecast_fun()`, accepts six parameters:

- `train`: A series representing LTD data across various hierarchical levels.
- `sales`: The time series data for sales.
- `hvi`: The time series data for the Home Value Index (HVI).
- `period`: The temporal frequency for generating nested Date variables, such as `month`, `2 months`, `quarter`, etc.
- `length`: The total number of rows in the input data.
- `fc_range`: The forecast range, which may include 12 months for monthly data, or 6 two-month periods for bimonthly data, etc.

The input data is consolidated into a single dataframe, or `tibble` object, with an appropriately nested Date variable. This `tibble` is subsequently transformed into a `tsibble` object utilizing the `as_tsibble()` function from the `tsibble` package. A VAR model is then applied to this `tsibble`

using the `VAR(vars(train, sales, hvi))` syntax, where parameters such as lag are automatically determined by the function. However, temporal aggregation results in insufficient observations for semi-annual and annual frequencies if the lag exceeds two. To address this limitation, an `if` condition restricts the chosen lag for these data frequencies to one.

Post model fitting, forecasts and residuals are computed as standard. Nevertheless, due to the lag configuration, there might be instances of missing residual values. Given that the `FoReco` package does not support missing values in the residual input matrices, these missing values are substituted with the calculated mean of the residuals. Since the number of these substituted values ranges from one to three, their impact on the reconciled results is anticipated to be minimal.

Ultimately, the function is structured to output both the mean point forecast and residuals, aggregated into a list format for subsequent extraction.

8.2.1.2.2 Vector Error Correction Model (VECM) The VECM fitting function, denoted as `vecm_forecast_fun`, accepts the same parameters as `var_forecast_fun` along with an additional `lag` parameter. This parameter facilitates the selection of varying lag lengths. The procedure for transforming the input data into a `tsibble` object remains consistent with that of the VAR model.

A distinct feature that differentiates VECM from VAR is its accommodation for non-stationary variables or those exhibiting cointegration patterns. The Johansen test, executed via the `ca.jo()` function from the `urca` package, assesses the presence of cointegration. The `K` parameter of `ca.jo()`, which determines the lag order in the VAR model employed within the Johansen procedure, is specified by the `lag` argument. This parameter can also be chosen using cross-validation to ascertain the optimal value for each temporal frequency or cross-sectional level, employing information criteria such as Akaike (AIC) or Bayesian (BIC).

Subsequent to the cointegration test, a significance level of 5% helps determine the value of `r`, representing the number of cointegrating relationships to be included in the VECM model. Should the Johansen test indicate an absence of a cointegration pattern, particularly in annual frequency data due to insufficient observations, a VAR model is utilized instead. If cointegration is confirmed, the VECM fitting function used is `vec2var()` from the `vars` package.

Similar to the VAR model, the VECM procedure also involves the generation of forecasts and residuals. Additionally, it addresses missing residual values caused by lag length, in a manner analogous to the `var_forecast_fun` procedure. The output of this function comprises a list containing the mean point forecast and residuals.

8.3 Reconciliation Process

The reconciliation procedure is consistent across ARIMA and VAR/VECM cross-validation models. Initially, forecasts and residuals from all cross-sectional levels and temporal frequencies are amalgamated into `base` and `res` datasets. The `FoReco` package facilitates the reconciliation process, offering integrated solutions for cross-sectional, temporal, and cross-temporal reconciliation.

- **`htsrec()`**: Implements cross-sectional reconciliation utilizing `shr`, which denotes the use of a shrunk covariance matrix—specifically MinT-shr.
- **`thfrec()`**: Facilitates temporal reconciliation employing `struc` for the computation of structural variances.
- **`octrec()`**: Conducts optimal cross-temporal reconciliation, using `struc` to calculate cross-temporal structural variances.
- **`tcsrec()`**: Executes heuristic first-temporal-then-cross-sectional reconciliation, combining structural variances for temporal aggregation with a bottom-up approach for cross-sectional aggregation.

The output of this reconciliation process is a matrix where rows represent different cross-sectional levels, and columns span from a 1-step to a 12-step-ahead forecast, reflecting the original scale of land transfer duty (LTD).

Subsequently, this matrix is integrated into a nested structure within an array corresponding to its designated `.id` value. This entire forecasting and reconciliation process is then repeated for subsequent folds or `.id` values.

9 Results

9.1 Model Performance:

Presentation of model accuracy and comparison.

9.2 Forecast Results:

Detailed discussion of the forecasted values and their confidence intervals.

10 Discussion

10.1 Interpretation of Results:

Analysis of what this forecasts mean for DTF.

10.2 Limitations and Assumptions:

Any limitations encountered during the forecasting process.

11 Conclusion and Recommendations

11.1 Summary:

Recap the findings and their implications.

11.2 Future Work:

Suggestions for improving future forecasts.

12 Appendices

12.1 Additional Data:

Any supplementary data or detailed tables.

12.2 Code:

Include or reference the R scripts used for analysis.

13 References

13.1 Bibliography:

Cite all data sources, literature, and software used in the report the insights generated by your work.

Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on .