Harvard Extension Data Science

Dynamic Modeling and Forecasting in Big Data

Instructor: William Yu

Assignment 7

Dynamic Factor Model (DFM) vs ARIMA

- In H07b_dfactor.R script, we learned the dynamic factor model. Now, revisit and run through the script again and import the W09c_macro.xlsx and GDPC1 (real GDP level data) via quantmod library.
- As the same as the script, we will keep the trainset data from 1990Q1 to 2023Q2. And the testset will be from 2023Q3 to 2024Q2. Our goal is to forecast real GDP level in the testset.
- *ARIMA Forecast*
- Use the auto.arima function to forecast real GDP for the next 4 quarters based on the trainset data (gdp = GDPC1['1990-01-01/2023-04-01']). And calculate the RMSE in the testset. For example, you can use the following:
  rmse_arima = sqrt(mean((fcst_arima-as.vector(GDPC1[299:302,]))^2))
- *Dynamic Factor Model (DFM) Forecast*
- Similar to the script in H07b_dfactor.R, use the macro2 data to compute principal components (PCs). But here, instead of using 8 PCs, let's try 5 PCs.
- Use these 5 PCs as explanatory variables to forecast GDPC1 in-sample (fit01).
- Following the script, use VAR (4 lag, type="both") to model these 5 PCs and forecast these 5 PCs for the next 4 quarters.
  Hint: You can check AR06 to see how to extract VAR forecast values.
- Form your factor forecast into a 4 by 5 matrix or data frame (say it is called pc_fcst), where 4 means 4 quarter forecast and 5 means 5 PCs.
- Note that testset forecast (Yhat; 4 by 1) is a matrix multiplication (Xtest (4 by 5 PC forecasts) %*% parameters (5 by 1 coefficients from fit01 in the trainset). Or you can simply use the following:

  ```
  input=data.frame(pc.fcst)

  colnames(input)=c("PC1","PC2","PC3","PC4","PC5")

  fcst_dfm = predict(fit01, input)
  ```

- Calculate the RMSE for real GDP in the testset with this DFM forecast.
- Compare these two RMSEs. Which one is a better model? Briefly explain why.
- Feel free to play around with different settings (different lags, different # of PCs) of the models.