This form documents the artifacts associated with the article (i.e., the data and code supporting the computational findings) and describes how to reproduce the findings.

Part 1: Data

☐ This paper does not involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).

☒ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

Abstract

Availability

☒ Data **are** publicly available.
☐ Data **cannot be made** publicly available.

If the data are publicly available, see the *Publicly available data* section. Otherwise, see the *Non-publicly available data* section, below.

Publicly available data

☒ Data are available online at: Michael W.~McCracken's website (https://research.stlouisfed.org/econ/mccracken/fred-databases/). We used the May 2022 version of FRED-MD.

☒ Data are available as part of the paper's supplementary material: We include FRED-MD_2022-05_preprocessed.csv (pre-processed data) and FRED-MD_2022-05.csv (raw data).

☐ Data are publicly available by request, following the process described here:

☐ Data are or will be made available through some other mechanism, described here:

Non-publicly available data

Description

File format(s)

☒ CSV or other plain text.
☐ Software-specific binary format (.Rda, Python pickle, etc.): pkcle
☐ Standardized binary format (e.g., netCDF, HDF5, etc.):
☐ Other (please specify):

Data dictionary

☐ Provided by authors in the following file(s):
☐ Data file(s) is(are) self-describing (e.g., netCDF files)
☒ Available at the following URL: https://research.stlouisfed.org/econ/mccracken/fred-databases/.

Additional Information (optional)

Part 2: Code

Abstract

The main functions are `lassoVAR.R` and `sqrtLassoVAR.R`, which are designed to estimate a VAR by weighted Lasso, post-Lasso and sqrt-Lasso going equation by equation. The basic inputs are a time series dataset

(expecting: time as rows, series as columns) and the number of lags used in the autoregression (default: one). To get the weighted lasso, call `fit <- lassoVAR(data = data, q = 1, post = FALSE)`, with post-Lasso arising from `post = TRUE` (default). The call for sqrt-Lasso is analogous. The basic outputs are the estimated coefficients (a matrix) and intercepts (a vector; included by default), which are extracted as `fit$that` and `fit$intr`, respectively. These functions rely on `helper_functions.R`. The main and helper functions are used in the simulation study and empirical illustration in the paper. The simulations are executed via `runSim_v03.R`, which calls on `simData.R` for the different simulation designs. The empirical illustration is conducted via `FRED-MD_forecasting_v02.R`.

Description

Code format(s)

- ☒ Script files
  - ☒ R
  - ☐ Python
  - ☒ Matlab
  - ☐ Other:
- ☐ Package
  - ☐ R
  - ☐ Python
  - ☐ MATLAB toolbox
  - ☐ Other:
- ☐ Reproducible report
  - ☐ R Markdown
  - ☐ Jupyter notebook
  - ☐ Other:
- ☐ Shell script
- ☐ Other (please specify):

Supporting software requirements

Version of primary software used R version 4.4.1.

Libraries and dependencies used by the code For calling main functions: glmnet (4.1.8), MASS (7.3.60.2), Matrix (1.7.0), stats (4.4.1).

For simulating data: Matrix (1.7.0), mvtnorm (1.3.2).

For parallel computing in simulation study/empirical illustration: doRNG (1.8.6), doParallel (1.0.17), foreach (1.5.1).

Supporting system/hardware requirements (optional)

The simulations and empirical illustration were run on a Linux server with 88 CPUs and 125 GB of RAM.

Parallelization used

- ☐ No parallel code used
- ☒ Multi-core parallelization on a single machine/node
  - Number of cores used: The main functions `lassoVAR.R` and `sqrtLassoVAR.R` involve no parallelization. The simulations and empirical illustration were run on a Linux server with 88 CPUs and 125 GB of RAM.
- ☐ Multi-machine/multi-node parallelization
  - Number of nodes and cores used:

License

- ☒ MIT License (default)
- ☐ BSD
- ☐ GPL v3.0
- ☐ Creative Commons
- ☐ Other: (please specify)

Additional information (optional)

Part 3: Reproducibility workflow

Workflow

To reproduce the simulations: Execute `runSim_v03.R` to run the main text simulations and `markupDependence_v01.R` for the mark-up dependence simulations, specifically. To have access to `lassoVAR.R` and `sqrtLassoVAR.R`, these scripts must be executed from the parent folder. Use `setwd("..")` to back up. When executed in a Linux environment, the script will automatically save the workspaces in the simulations folder. If a different environment is used, the workspace must be saved manually (via `same.image(...)`). The figures included in the paper arise from calling the scripts `createFigs_v07.R` and `markupDependenceFigs_v02.R` from their folder, which load the relevant workspaces from the simulations folder and produce figures in the `simulations/img` folder.

To reproduce the empirical illustration: Execute `FRED-MD_forecasting_v02.R` (from the parent folder via `setwd("../..")`) to conduct the forecasting exercise. The script loads the pre-processed data from the `application/FRED/data` folder and automatically saves the workspace (if in Linux). Run `FRED-MD_figures_v02.R` to produce the figures in the empirical illustration in the `application/FRED/img` folder.

Scope

The provided workflow reproduces:

- ☐ Any numbers provided in text in the paper
- ☐ The computational method(s) presented in the paper (i.e., code is provided that implements the method(s))
- ☒ All tables and figures in the paper
- ☐ Selected tables and figures in the paper, as explained and justified below:

Location

The workflow is available:

- ☒ As part of the paper's supplementary material.
- ☐ In this Git repository:
- ☐ Other (please specify):

Format(s)

- ☐ Single master code file
- ☐ Wrapper (shell) script(s)
- ☒ Self-contained R Markdown file, Jupyter notebook, or other literate programming approach
- ☐ Text file (e.g., a readme-style file) that documents workflow
- ☐ Makefile
- ☐ Other (more detail in *Instructions* below)

Instructions

Expected run-time

Approximate time needed to reproduce the analyses on a standard desktop machine:

- ☐ < 1 minute
- ☐ 1-10 minutes
- ☐ 10-60 minutes
- ☐ 1-8 hours
- ☐ > 8 hours
- ☒ Not feasible to run on a desktop machine, as described here: The simulations and empirical illustration were run on a Linux server with 88 CPUs and 125 GB of RAM. In this environment, the runtimes for the main simulations, mark-up dependence simulations, and empirical illustration are approximately 32 hours, 3.5 hours, and 4 hours, respectively.

Additional information (optional)

Notes (optional)