The MVGC Multivariate Granger Causality Matlab® Toolbox

The MVGC Matlab® Toolbox is designed to facilitate Granger-causal analysis with multivariate and possibly multi-trial time series data. It is not "black box" software. There is no GUI, but rather a set of functions designed to be used in your own Matlab® programs. Annotated demonstration scripts are available which may be used as templates to assist in this task.

The toolbox uses a novel, accurate and highly efficient approach to numerical computation and statistical inference of Granger causality, conditional and unconditional, in both time and frequency domains, as described in the accompanying reference document [1]. The toolbox is intended to supersede the popular Granger Causal Connectivity Analysis (GCCA) Toolbox, and to a large extent subsumes, enhances and extends GCCA functionality. The software is developed and maintained by Lionel Barnett at the Sackler Centre for Consciousness Science (SCCS), University of Sussex, supported by the Dr. Mortimer and Theresa Sackler Foundation. For copyright and licencing terms please see the bottom of this page.

Contents

System requirements

The MVGC Toolbox has been mostly developed and tested on a Linux x86_64 (glnxa64) system running Matlab® version R2011a; with the potential exception of some C code (see below) it should be largely platform-independent. We cannot, however guarantee that the toolbox will run without problems on earlier or later versions of Matlab®.

Matlab® Toolboxes: The MVGC Toolbox requires the Statistics Toolbox™ for some essential functionality. The Control System Toolbox™ is recommended; the core MVGC function `var_to_autocov` requires a discrete-time Lyapunov equation solver and, if installed, will use the Control Syetem Toolbox `dlyap` function (if that link worked you've got it!). If not, the default is currently to use a slower Matlab®-scripted solver. A faster experimental solver is also available; see `var_to_autocov` for more details. The spectral estimation routine `tsdata_to_cpsd` requires the Signal Processing Toolbox™, but is not considered essential functionality.

Help and documentation

Formatted reference pages for all MVGC functions are available in the Function Reference section of the MVGC help in the Matlab Help Browser (*note*: in Matlab® version 2013a and later, this is accessed via the *Supplemental Software* link at the bottom of the Help Browser home page). There is also a utility function `helpon` which displays these pages in the Help Browser; to get help on an MVGC function or script called *`name`*`.m` simply type

>> helpon *name*

at the Matlab® command prompt. (Unfortunately Matlab does not currently facilitate `F1`-key context-sensitive help for

user-defined pages.) For more detail and theory see the reference document [1], and see also the `docs`, `docs/html` and `demo` subdirectories of the MVGC root directory. The Release Notes should be checked too for further relevant information and recent changes to this software.

Getting started

The MVGC Toolbox is installed simply by unzipping/untarring the distribution file in a convenient parent directory. This will create a subdirectory called `mvgc_version`, the *MVGC root directory*, in the parent directory. In Matlab® navigate to the MVGC root directory and run the `startup` script to intialise the toolbox and integrate the toolbox help and documentation into the Matlab® Help system. Note that, subsequent to installation, the `startup` script will be run automatically if Matlab® is started in the MVGC root directory.

The easiest way to get started with the MVGC Toolbox is via the annotated demos in the `demo` subdirectory of the MVGC root directory, in particular the `mvgc_demo` script. These may be used as templates for your own code. Having said that, the MVGC Toolbox is *not* to be regarded as "black box" software! For successfull use some basic understanding of the principles underlying Granger-causal inference and the computational approach of the MVGC Toolbox is requisite. It is thus *strongly recommended* that the user take some time to review the reference document [1], with particular attention to Section 3 on design principles of the MVGC Toolbox. The schema of MVGC computational pathways provides a useful overview.

Common variable names and data structures

The following are common names for variables and data structures in the MVGC toolbox.

```
n,nvars    number of variables

m,nobs     number of observations (time steps)

N,ntrials  number of trials (realisations) of a process

p,q,...    number of lags or frequencies

x,y,z,...  vector of sub-variables indices

fs         sample rate (e.g. in Hz)

fres       frequency resolution

X,Y,Z,...  Time series data: an n x m matrix for single-trial, or an
           n x m x N matrix for multi-trial data. The first index
           references variables, the second observations and the third (if
           present) trials.

A          VAR coefficients: an n x n x p matrix containing coefficients
           for an n-variable VAR(p). The first index references target
           ("to") variables, the second source ("from") variables and the
           third lags.

SIG        Residuals covariances: and n x n symmetric positive-definite
           matrix containg residuals covariances for an n-variable VAR.

G          Autocovariance sequence: an n x n x (q+1) matrix containing
           autocovariances of a time series up to q lags. The first index
           references target ("to") variables, the second source ("from")
```

```
            variables and the third lags. The k-th block G(:,:,k)
            corresponds to k-1 lags; the first block G(:,:,1) is thus a
            covariance matrix.

S           Cross-power spectral density (cpsd): an n x n x (q+1) matrix
            containing cpsds of a time series at a resolution of q. The
            first index references target ("to") variables, the second
            source ("from") variables and the third frequency. Frequencies
            generally run from zero up to (and including) the Nyqvist
            frequency; i.e. block S(:,:,1) corresponds to frequency zero
            and block S(:,:,q+1) corresponds to the Nyqvist frequency.

H           VAR transfer function: an n x n x (q+1) matrix containing
            transfer function components of a VAR at a resolution of q. The
            first index references target ("to") variables, the second
            source ("from") variables and the third frequency.

F           Time-domain multivariate Granger causality: for the pairwise-
            conditional case, an n x n matrix containing pairwise-
            conditional causalities; the first index then references target
            ("to") variables and the second source ("from") variables, with
            NaNs on the diagonal.

f           Frequency-domain (spectral) multivariate Granger causality: for
            the pairwise-conditional case, an n x n x (q+1) matrix
            containing pairwise-conditional spectral causalities; the first
            index references target ("to") variables, the second source
            ("from") variables and the third frequency, with NaNs on the
            diagonal for each frequency block f(:,:,k).
```

Function arguments

Default input arguments, if present (which will be indicated in the corresponding Function Reference page), may always be entered as an empty matrix `[]` or, for strings, the empty string `''`. Thus, for example, the function `var_to_autocov`

```
[G,info] = var_to_autocov(A,SIG,acmaxlags,acdectol,aitr,maxiters,maxrelerr)
```

could be called as

```
G = var_to_autocov(A,SIG,[],[],true);
```

String arguments are always case-insensitive.

The toolbox encourages the use of standard Matlab® syntax for missing output arguments, which may result in more efficient execution. For example, the function `tsdata_to_var`

```
[A,SIG,E] = tsdata_to_var(X,p,regmode)
```

could be called as

```
[~,SIG] = tsdata_to_var(X,p,'LWR')
```

if only the output `SIG` is required, which will result in faster code.

Miscellaneous issues

Stationarity:

Granger-causal analysis based on VAR modelling presupposes that the time series data in question represent a *stationary* (multivariate) process. Of course in practice this may not be a tenable assumption. One way to deal with this issue, easily implementable in the MVGC Toolbox, is by *windowing* the time series data; that is, performing causal analysis on (possibly sliding) short time segments ("windows") of the data that, hopefully, are near-enough stationary. This approach is all the better if mutiple synchronised trial data is available. There will, however, inevitably be a trade-off between time window size and quality of statistical inference given the resultant smaller data samples. The demo script `mvgc_demo_nonstationary` illustrates how to put this technique into practice.

Data preprocessing:

We remark that we do not see it as the role of the MVGC Toolbox to provide functionality for preprocessing time series data (e.g. to improve stationarity); there is other software available for this. Regarding preprocessing, however, it is worth a warning that some common preprocessing procedures - such as *filtering* - have the potential to seriously disrupt Granger causal inference (see the remarks in the `mvgc_demo` script) and may, furthermore, cast doubt on the interpretation of results. Our view is that preprocessing should be handled with trepidation and preferably kept to a minimum, for instance to achieve acceptable stationarity.

VAR modelling?

It should be borne in mind that, notwithstanding that (almost) any wide-sense stationary multivariate stochastic process may be modelled as a VAR, this is not to say that a VAR will necessarily be a *good* ("parsimonious") model for your data! Indeed, excessively high model orders (*cf.* `tsdata_to_infocrit`) may well be an indication that a VAR model is simply unsuitable (or it may be indicative of non-stationarity). For example, theory tells us that a VAR process has exponentially decaying autocovariance in the long term. If your data exhibits "long term memory" (i.e. power-law autocovariance decay) then, e.g. a fractional ARIMA model may be better suited to your data and VAR-based Granger-causal analysis is probably not the way to go. Similarly, a VARMA process with non-trivial moving-average component may yield excessively high model order when modelled as a VAR. It would certainly be pleasant to be able to calculate Granger causality directly for such alternate models. There has been some progress in this area - indeed, we're working on it ourselves (see e.g. L. Barnett and T. Bossomaier, "Transfer entropy as a log-likelihood ratio" [`preprint`], *Phys. Rev. Lett.* **109(13)**, 2012) - but that is for a future release of the toolbox.

GCCA compatibility mode

Although the MVGC native approach to Granger causality computation is obviously recommended, former users of the Granger Causal Connectivity Analysis (GCCA) Toolbox may wish to refer to the `mvgc_demo_GCCA` script, which demonstrates and explains usage of the MVGC Toolbox in "GCCA compatibility mode".

C code

The MVGC Toolbox is almost entirely written in Matlab® code; however, for efficiency reasons, a few routines may be coded in C (at present only the `genvar` function, used to generate test VAR data). If corresponding `mex` files for your platform are not included in your distribution of this software (the `startup` script will issue a warning) you should try to build them using the `mvgc_makemex` function. In general, however, a missing MVGC `mex` file is not a show-stopper; (slower) Matlab®-scripted code with equivalent functionality should always be available and invoked automatically.

Note 1: The toolbox is currently distributed with pre-built and tested `mex` files for 64-bit Unix (including Linux), Windows and Mac, as these were the only test platforms available to us. If Matlab® crashes on you, there is a very good chance that a pre-built `mex` is to blame. In this case (assuming you have a Matlab®-compatible C compiler available) you should try running `mvgc_makemex` with the `force_recompile` flag set.

Note 2: The pre-built Windows 64-bit `mex` files were compiled with Microsoft® Visual Studio 2010. Apparently code

compiled with this compiler requires the Microsoft® Visual Studio 2010 runtime components. There is not much we can do about this; if you do not have Microsoft® Visual Studio 2010 installed on your 64-bit Windows system you can install the required components from here, or recompile the `mex` files using a different compiler, again by running `mvgc_makemex` with the `force_recompile` flag.

References

[1] L. Barnett and A. K. Seth, The MVGC Multivariate Granger Causality Toolbox: A New Approach to Granger-causal Inference, *J. Neurosci. Methods* 223, 2014.

Note: We are in the process of making this publication open-access; in the meantime, a preprint (included in the distribution) is available [here] (`PDF` format - set Matlab default viewer in `File -> Preferences -> Help -> PDF Reader`).

Contact the authors

For general support issues, comments, questions, bug reports and suggested enhancements, please email m v g c t o o l b o x AT s u s s e x . a c . u k. We would especially like to know if you have found the toolbox useful in your research.

back to top