DCC-NL Package

Dynamic **C**onditional **C**orrelation Model with **N**on**L**inear Shrinkage

Estimate Time-Varying Covariance Matrices Up to Dimension 1,000

By Robert F. Engle, Olivier Ledoit, Michael Wolf, and Zhao Zhao

1. Introduction

This is the Matlab software package based on the article authored by Robert F. Engle, Olivier Ledoit and Michael Wolf entitled "Large Dynamic Covariance Matrices" and accepted for publication by the *Journal of Business and Economics Statistics* in June 2017. Zhao Zhao is from the Department of Economics at the Huazhong University of Science and Technology, and she contributed to the development of this software as the Research Assistant to the paper's authors.

> *The two major advantages of such covariance matrix estimates are that: (i) they adapt through time as circumstances evolve; and (ii) no matter how large the dimension, they remain well-conditioned for the purpose of minimizing out-of-sample risk.*

2. Prerequisites

a) A valid Matlab License.

b) A Matlab Compiler to compile C code. For a list of supported C compilers compatible with your installation, visit http://www.mathworks.com/support/sysreq/previous_releases.html. Given that this is a commercial package, we cannot provide any assistance or support in this respect.

c) Access to the Matlab Optimization Toolbox, and in particular the function `fmincon.m`.

d) Having downloaded the QuEST package, which is freely available from the official website of the second author in the Department of Economics at the University of Zurich: http://www.econ.uzh.ch/en/people/faculty/wolf/publications.html#9. Specifically, it is the code for the paper "Ledoit O. and Wolf, M. (2017). Numerical implementation of the QuEST function. Computational Statistics & Data Analysis 115, 199-223." Keep note of the path where you unzip the code, as you will need it later.

e) Having downloaded the package for direct kernel estimation of the optimal nonlinear shrinkage formula, which is freely available in the same location at the University of Zurich: http://www.econ.uzh.ch/en/people/faculty/wolf/publications.html#9. Specifically, it is the code for the paper "Ledoit O. and Wolf, M. (2017). Direct nonlinear shrinkage estimation of large-dimensional covariance matrices. Working Paper ECON 264, Department of Economics, University of Zurich." Keep note of the path where you unzip the code, as you will need it later.

f) The Oxford MFE toolbox of Professor Kevin Sheppard, which is available from the bitbucket repository http://bitbucket.org/kevinsheppard/mfe_toolbox/get/default.zip. Once you have unzipped it, the first order of business is to run `buildZipFile.m` in order to create the directory with all the `dlls`. ***TOP TIP:*** some versions of `buildZipFile.m` assume that the extension for a compiled MEX file on an Apple Mac is `.mexa64`, but on your systems it might be `.mexmaci64` instead. So in that case hack into line 10 of `buildZipFile.m` and fix it yourself. (Given that Prof. Sheppard is the one who maintains his own toolbox, we cannot provide any assistance or support with this step.)

g) For matrix dimension 1,000 we recommend at least 32GB of RAM, and preferably 64GB to be safe. But you should definitely try it out on your system. And it is always possible to work in dimensions below 1000 if need be.

3. Installation + Validation

- The DCC-NL package is distributed with a MEX file compiled for Mac called `dcc_likestream.mexmaci64` and its counterpart for Windows, which is called `dcc_likestream.mexw64`. If these do not work on your system, you have to generate your own MEX files with Matlab using the C code which is also provided. This is done by typing the instruction: `mex dcc_likestream.c`. It requires your having set up a C compiler inside Matlab in the first place as per item b) of Section 2.
- To check that all is in order, run `DCC_NLdemo.m`. Beforehand, make sure you have adjusted the paths to the QuEST package (on line 12) and the MFE package (on line 16) at the top of the routine as required (cf. Section 2 above).
- If you have insufficient memory, you can always cut off some columns of the demo data set to reduce matrix dimension.
- For validation purposes, the final output in dimension 100 (left) / 1000 (right) should look like this:

```
0.0711    0.0247    0.0305  |  0.1357    0.0352    0.0406
0.0247    0.0974    0.0309  |  0.0352    0.0659    0.0323
0.0305    0.0309    0.0734  |  0.0406    0.0323    0.0545
```

4. How to Estimate Large Dynamic Covariance Matrices

The basic syntax for invoking the central function is as follows:

```
sigmahat = DCCcov(x);
```

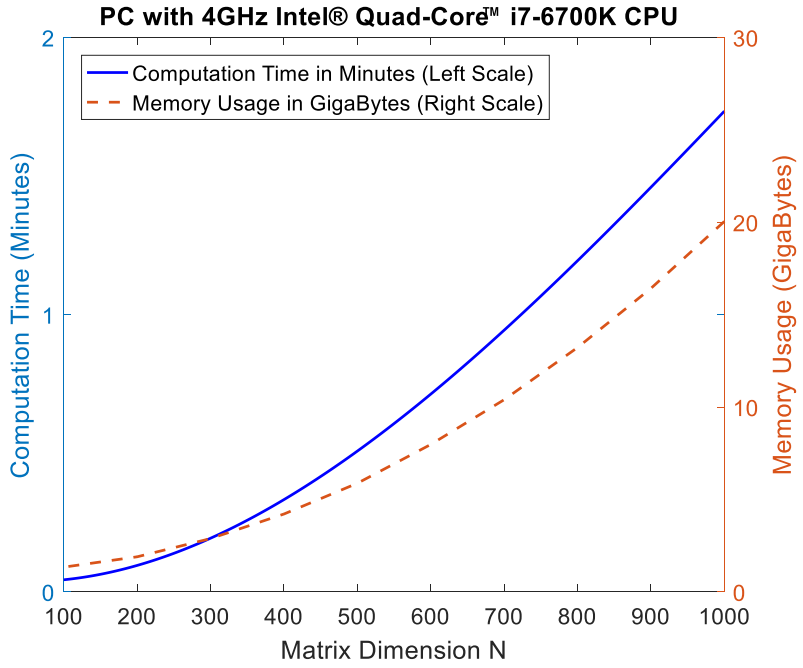
where:

- The input `x` is a 2-dimensional matrix containing the data, where each one of the `T` rows represents a date and each one of the `N` columns represents a random variable.
- The output `sigmahat` is a 2-dimensional matrix of dimension `N×N` containing the conditional covariance matrix estimated at date T, the last date of the sample.

The original paper used the QuEST package to estimate the unconditional covariance matrix target, and this is still possible as an option in DCCcov, but now the default is to use the direct kernel estimation method, which is orders of magnitude faster, and as accurate for practical purposes.

5. Speed

Estimating `T=1,250` time-varying covariance matrices, each one of dimension `1000×1000`, takes less than 3 minutes on an Apple Mac Pro with a 3.5 GHz Intel® Xeon™ E5 processor, or on a Windows PC with a 4.0 GHz Intel® Quad-Core™ i7-6700K processor. The mapping of matrix dimension into computational time and peak memory usage is shown in the following graph.



6. Supplement: BEKK-NL Model with Scalar Dynamics

This could also be called a symmetric scalar multivariate vech GARCH model using variance targeting. While this alternate model is neither the main focus of this package, nor of the article from which the package originates, it was nonetheless introduced in Section 7 of the paper. So we also include the software for it for completeness' sake. It follows the general pattern of DCC-NL. The transformation from DCC-NL to BEKK-NL was coded up by Dr. Zhao Zhao.

- The MEX file compiled for Mac is called `vt_likestream.mexmaci64` and the Windows equivalent `vt_likestream.mexw64`. If these do not work on your system, you have to generate your own MEX files with Matlab using the C code which is also provided. This is done by typing the instruction: `mex vt_likestream.c`.
- To check that all is in order, run `BEKK_NLdemo.m`. First make sure you have adjusted the paths to the QuEST package and the MFE package at the top of the routine as in `DCC_NLdemo.m` (cf. Section 3 above). It takes less than 2 minutes to run on our computers.
- For verification purposes, the final output in dimension 100 (left) / 1000 (right) should look like:

```
0.0657    0.0229    0.0234  |    0.1442    0.0420    0.0380
0.0229    0.0995    0.0264  |    0.0420    0.0783    0.0327
0.0234    0.0264    0.0590  |    0.0380    0.0327    0.0483
```

- The basic syntax for the estimation of large conditional covariance matrices in the BEKK-NL model is as follows:

```
sigmahat = BEKKcov(x);
```

  where the input and output have the same interpretations as in Section 4.

7. Disclaimers

No claim is made that this code is the most elegant, most efficient, or definitive implementation of the models introduced in the article. It just worked insofar as it enabled us to run our simulations and empirical applications successfully. The authors reserve the right to alter in future iterations.

Acknowledgment

Some of the code included in this Package is partially indebted to the MFE Toolbox written by Kevin Sheppard from the University of Oxford Department of Economics. We wish to thank Professor Sheppard for having made his Oxford MFE Toolbox freely available to the wider Econometrics community. To the extent that we retained certain syntactic conventions, it was based on the assumption that some researchers interested in the DCC-NL Package might have already been acquainted with the MFE Toolbox, even though the two have different objectives.

License

8. Version Tracker

Version 01: Internal release dated August 20th, 2017; not circulated publicly.
Version 02: Minor modifications such as file name changes; dated August 21st, 2017.
Version 03: Incorporates internal feedback; dated August 24th, 2017.
Version 04: Incorporates feedback from beta testers; dated September 12th, 2017.
Version 05: Add option to demean the data, add option to call the direct kernel estimator of the unconditional target matrix, and minor semantic changes; dated November 16th, 2017.

9. Contents

The zipped archive contains a total of 13 files:

I.	DCC_NLreadme.pdf	Package documentation.
II.	DCC_NLdata.mat	Test data set for multivariate GARCH model estimation.
III.	DCC_NLdemo.m	Demonstrates estimation of DCC-NL model.
IV.	DCCcov.m	Estimate conditional covariance matrices in DCC-NL model.
V.	dcc_likestream.c	C code to speed up maximization of DCC-NL log-likelihood.
VI.	dcc_likestream.mexmaci64	Matlab executable of DCC-NL log-likelihood for Mac.
VII.	dcc_likestream.mexw64	Matlab executable of DCC-NL log-likelihood for Windows.
VIII.	dcc_likelihood2.m	Log-likelihood function of DCC-NL model.
IX.	BEKK_NLdemo.m	Demonstrates estimation of BEKK-NL model.
X.	BEKKcov.m	Estimate conditional covariance matrices in BEKK-NL model.
XI.	vt_likestream.c	C code to speed up maximization of BEKK-NL log-likelihood.
XII.	vt_likestream.mexmaci64	Matlab executable of BEKK-NL log-likelihood for Mac.
XIII.	vt_likestream.mexw64	Matlab executable of BEKK-NL log-likelihood for Windows.