MATLAB codes README file accompanying: Closed-Form Multi-Factor Copula Models with Observation-Driven Dynamic Factor Loadings

Anne Opschoor[a,b], André Lucas[a,b], István Barra[a,b], Dick van Dijk[c,b]

[a] *Vrije Universiteit Amsterdam*

[b] *Tinbergen Institute*

[c] *Erasmus University Rotterdam*

1 Files and set-up

The code distribution contains the following files:

`Testdata.csv`	File with simulated data to test the code with and obtain the values in the tables in this document.
`Main.m`	Matlab file to run using test data and estimate a range of factor models as described in the paper.
`Admin.m`	This Matlab class contains auxiliary routines to compute matrices like the duplication matrix and others, but also to compute selection matrices to keep track of which time varying loading is in which asset equation for the different factor models.
`LogLik_Factor_Copula.m`	This Matlab class contains several log-likelihood functions of 1, 2 and multifactor copula models from the paper.

2 Data

Given we are not free to distribute the TAQ data used in the paper, we provide a simulated dataset for $N = 100$ stocks and $T = 1,000$ time periods with $G = 10$ groups. The number of firms per group is equal to 10 for all groups.

The group assignment is done in `Main.m` for the stocks and reads:

```
asset_group_vec = kron([1:10]',ones(10,1));
```

For other datasets, the group allocation assignment has to be changed appropriately.

3 Main.m

The file `Main.m` reads the data, sets up the group allocation, and then estimates a 1F-equi (Model 1 and 2), 1F-Group (Model 3 and 4), 2F (Model 5 and 6), MF (Model 7 and 8), and MF-LT (Model 9 and 10) model for a Gaussian (odd model numbers) and a Student's t (even model numbers) distribution. Finally, the file computes the standard errors based on the sandwich covariance matrix and prints the results. Via the vector `model_est_ind_vec` one can select which model(s) should be estimated.

The output produced when `Main.m` has been called should be as follows:

```
Estimated parameters of Model  2 (s.e. in parentheses)
omega: 0.0201 (0.0136)
A: 0.0015 (0.0008)
B: 0.9702 (0.0202)
nu Cop: 28.8081 (1.3691)
Maximized LogLikelihood:  17906.0

Estimated parameters of Model  4 (s.e. in parentheses)
 omega_1: 0.0329 (0.0075)
 omega_2: 0.0343 (0.0077)
 omega_3: 0.0291 (0.0066)
 omega_4: 0.0245 (0.0058)
 omega_5: 0.0186 (0.0046)
 omega_6: 0.0236 (0.0056)
 omega_7: 0.0205 (0.0050)
 omega_8: 0.0284 (0.0065)
 omega_9: 0.0281 (0.0063)
 omega_10: 0.0296 (0.0068)
 A: 0.0075 (0.0008)
 B: 0.9615 (0.0084)
 nu Cop: 30.5601 (1.5984)
Maximized LogLikelihood:  18843.7
```

```
Estimated parameters of Model 10 (s.e. in parentheses)
 A: 0.0163 (0.0006)
 B: 0.9714 (0.0024)
 nu Cop: 36.5959 (1.6531)
Maximized LogLikelihood:  24139.2
```

4 LogLik_Factor_Copula.m

The core of code consists of the file `LogLik_Factor_Copula.m`. This is a class consisting of functions for computing the copula log-likelihood of all factor-copulas proposed by Opschoor et al.(2020). The file `Admin.m` contains auxiliary functions required for computing the log-likelihoods of various factor copula models.

Lines 3-28 of `LogLik_Factor_Copula.m` contain general information about the input and output of all functions within the class.

```
% General input
%   N: number of assets
%   T: number of time periods
%   u_mat: a matrix of (T by N) with PITs
%   asset_group_vec: a (N by 1) vector classifying each asset to a specific
%                    industry (e.g. 1 = financials, 2 = health, etc)
%   ind_t_dist:      1 if t-copula, Gaussian copula otherwise
%   ind_Rt: optional argument, 1 if you would like to compute R_t, 0 else


% General output
%   LLF: the MINUS copula log likelihood
%   loglike_vec: a (T x 1) vector of log-likelihood values at time t
%   R_vec/ R_mat: fitted dependence matrices (T x N x N)
%   s_vec/ s_mat: a T by p matrix of scores (with p the number of
%                unique factor loadings)
%   f_vec/ f_mat: a T by p matrix of f_values (the unique unscaled factor
%                loadings that change over time by a score update
```

```
%                (with p the number of unique factor loadings)
```

The following functions are in `LogLik_Factor_Copula.m`:

Name	Description
`LogLik_Copula_1f_eq`	computing the log-likelihood of the 1F-equi model.
`LogLik_Copula_1f_gr`	computing the log-likelihood of the 1F-group model.
`LogLik_Copula_2f`	computing the log-likelihood of the 2F model.
`LogLik_Copula_MF`	computing the log-likelihood of the MF model.
`Moment_based_omega_LT_FC`	function implementing the moments estimator for the intercept in the time-varying parameter transition equation. This routine is needed for `LogLik_Copula_LT_factor_given_omega`.
`LogLik_Copula_LT_factor_given_omega`	computing the log-likelihood of the MF-LT model. Requires `Moment_based_omega_LT_FC`.

All these `LogLik` functions have a similar set-up. As an illustration we discuss one multi-factor copula model.

4.1 The MF model

The MF model is called as

```
function [LLF,loglike_vec,R_mat,s_mat,f_mat] =
LogLik_Copula_MF(N,T,params,u_mat,asset_group_vec,ind_t_dist,ind_Rt);
```

It uses as inputs:

`N`	the number of assets
`T`	the number of time periods
`params`	the vector of parameters, holding (ω, A, B, ν), where ν is not there for the Gaussian case.
`u_mat`	the PITs of the univariate analyses as a $T \times N$ matrix.
`asset_group_vec`	a $N \times 1$ vector that allocates each asset to a certain group g.
`ind_t_dist`	boolean, if TRUE (1) then use the Student's t copula density, otherwise the Gaussian copula density.
`ind_Rt`	boolean, if TRUE (1) then store the dependency matrices $\boldsymbol{R}_t$ at each time t.

The likelihood routine makes the following steps:

1. Initialize all variables needed.

2. Loop over `1:T` to compute the time t likelihood values and updating the dynamic parameter using the score-driven updates.

3. Finalize and return the log likelihood value.

This structure is shared by all the likelihood routines. The only differences occur in step 2, where we try to efficiently evaluate the scores. We refer in the Matlab code to specific equations of Web Appendix F when calculating the score.