f-SVAR Toolbox User Manual

Estimating Dynamic Connectivity States in Neuroimaging Time-Series Data

Chee-Ming Ting
Center for Biomedical Engineering, Universiti Teknologi Malaysia
King Abdullah University of Science and Technology, Saudi Arabia
cmting@utm.my

Siti Balqis Samdin
Center for Biomedical Engineering, Universiti Teknologi Malaysia

Hernando Ombao
Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University of Science and Technology, Saudi Arabia

Department of Statistics, University of California, Irvine
hernando.ombao@kaust.edu.sa/hombao@uci.edu

Version 1.0 (27 Nov 2017)

1. Description

f-SVAR (Factor switching vector autoregressive (VAR) model) Toolbox is a Matlab package to estimate state-driven changes in directed brain connectivity networks with a large number of nodes, from neuroimaging time series data (fMRI, EEG etc). It provides statistically reliable and computationally efficient way for estimating change-points of brain connectivity regimes/states and massive directed dependencies associated in each regime. It is based on a non-stationary factor model with regime-switching in latent factor dynamics. It implements a three-step estimation procedure:

(1) Initialization based on a factor model (FM) estimated by principal component analysis (PCA) method.

(2) Identification of connectivity regimes in a low-dimensional latent factor space based on the Markov-switching factor model. By a state-space formulation, the latent state sequence is extracted by Kalman filtering and smoothing algorithms, and the state-specific factor VAR coefficient matrices are updated by expectation and maximization (EM) algorithm.

(3) Projection of high-dimensional state VAR connectivity matrices in observation space based on an f-SVAR model by plugging-in subspace estimates in steps (1)-(2).

- (i) Coupled estimator, with common factor loadings (f-SVAR-ComQ)
- (ii) Decoupled estimator, with state-dependent factor loadings (f-SVAR-DepQ)

2. Input Data Format

The data are defined by variables *Y* and *Yr* which contain the time series data, and in matrix or array format

- *Y*: N x T matrix of time series data concatenated across signal replicates (multiple epochs or subjects)
- *Yr*: N x Tr x Rs array of time series data per each replicate
- *N:* Number of time series
- *Rs:* Number of replicates
- *T:* Number of time points of concatenated data (*T= Tr* x *Rs*)
- *Tr*: Number of time points per replicate

3. Main Functions

3.1 Initialization (`fsvarInit.m`)

This function initializes parameters of f-SVAR model for EM estimation based on estimates of a common factor model on concatenated data. It implements two steps: (1) Estimate a factor model by PCA with sub-function `fmest.m`. (2) Initialize VAR coefficient matrix at each state based on K-means clustering of sliding-windowed VAR coefficients of latent factors estimated in step (1).

```
[fsvar0,fm] = fsvarInit(Y,T,opts)
```

Input arguments:

- ***Y*** - *N* x *T* time series data
- ***T*** - Total number of time points
- ***opts***: structure with model specification
  - ***p*** - VAR model order
  - ***K*** - Number of states/regimes
  - ***r*** - Number of factors
  - ***wlen*** - window size for sliding-windowed VARs
  - ***shift*** - window shift for sliding-windowed VARs

Output arguments

- ***fsvar0***: Struct containing initial estimates of f-SVAR model parameters
  - ***A*** - VAR coeff. mat
  - ***V*** - State cov. mat
  - ***H*** - Obs mapping
  - ***R*** - Obs cov. mat
  - ***x0*** - Initial state
  - ***Z,pi*** - State transition matrix

- ***fm***: Struct containing estimates of factor model
    - ***Q*** - factor loading matrix
    - ***f*** - factor time series
    - ***e*** - residuals
    - ***cov_f*** - covariance matrix of factors
    - ***cov_e*** - residual covariance matrix

3.2 Estimation of f-SVAR model by EM `(fsvarest.m)`

This function estimates hidden state sequence by Kalman filtering and smoothing, and updates model parameters with EM algorithm. It receives input from output `fsvar0` of `fsvarInit.m`

```
[fsvar,Path,L] = fsvarest(Yr,fsvar0,opts)
```

Input arguments:

- ***Yr*** - time series data
- ***fsvar0***: struct with initialized parameters
    - ***A*** - VAR coeff matrix (state transition x(t-1)->x(t))
    - ***V*** - State noise covariance matrix
    - ***H*** - Observation mapping matrix of x(t)->y(t)
    - ***R*** - Observation noise covariance matrix
    - ***x0*** - Initial state
    - ***Z*** - Markov transition matrix
    - ***pi*** - Initial state probability
- opts: struct of EM algorithm settings
    - ***ItrNo*** - maximum iteration of EM algorithm
    - ***eps*** - Stop EM when eps< likelihood improvement
    - ***p, K, r*** – model specification

Output arguments:

- ***Path***: Structure with estimated latent regimes *St* and factor dynamics, *xt* for each replicate *s*
    - ***Rs*** - number of replicates
    - ***fSt*** - State/Regime probability estimated by switching Kalman filter, *P(St=j|y(1:t))*
    - ***sSt*** - State/Regime probability estimated by switching Kalman smoother, *P(St=j|y(1:T))*
    - ***St_skf*** - Most likely state sequence by hard assignment, *argmax_j P(St=j|y(1:t))*
    - ***St_sks*** - *argmax_j P(St=j|y(1:T))*
    - ***fxt*** - Filtered state vector, *P(xt|y(1:t))*
    - ***sxt*** - Smoothed state vector, *P(xt|y(1:t))*

  - ***fft*** - Filtered factors (*fxt(1:r,:,:)* : *r x 1* vectors of factor components)
  - ***sft*** - smoothed factors (*sxt(1:r,:,:)*)
- ***fsvar***: Struct with EM estimated model parameters
  - ***A[:,:,j]*** - AR coefficient matrix at regime *j*
  - ***V[:,:,j]*** - State-specific noise cov at regime *j*
  - Other fixed parameters, ***H***, ***R***
  - ***L*** - Log-likelihood each iterations

3.1 Projection of High-dimensional VAR connectivity states `(fsvarproj.m)`

This function projects the state-specific VAR coeff matrix from factor space to observation space. It computes two f-SVAR estimators (1) Coupled estimator (common factor loadings) (2) Decoupled (state-dependent factor loadings) estimator. It also performs statistical testing of the AR coefficients of the decoupled estimator. It requires inputs from the outputs `fsvar,Path,fm` of `fsvarInit.m` and `fsvarest.m`.

```
[varmat] = fsvarproj(Yr,fsvar,Path,fm,opts)
```

Input arguments:

- ***fsvar*** - Struct with EM-estimated f-SVAR parameters
- ***Path*** - Struct with estimated regimes by switching Kalman filter and smoother
- ***fm*** - Struct with PCA-estimated factor model

Output arguments:

- ***varmat***: Structure with projected state connectivity matrix
  - ***Ade[:,:,j]*** - Estimated Coupled SVAR matrix (*N x Np*) at state *j*
  - ***Ade_sig[:,:,j]*** - Ade with significant coefficients
  - ***Aco[:,:,j]*** - Estimated Decoupled SVAR matrix (*N x Np*) at state *j*

4. Supporting Functions

4.1 Estimation of factor model by PCA `(fmest.m)`

This function estimates the parameters of factor model using method of principal components analysis (PCA). It is called by function `fsvarInit.m`.

```
[Q, cov_f, f, cov_e, e] = fmest(y,r)
```

Input arguments:

- ***y*** - N x T time series data
- ***r*** - known number of factors

Output arguments: Estimators of ***Q, f, cov_e, e*** (estimated residuals)

4.2 Estimation of number of factors `(fmselr.m)`

This function estimates the unknown number of factors, *r*. It evaluates a range of candidate values of *r* between 1 and *kmax*. It calls `fmest.m` to estimate parameters of factor model.

```
[r, b, bic] = fmselr(y,kmax)
```

Input arguments: ***y*** - N X T observations, ***kmax*** - maximum range of candidate *r*
Output arguments: ***r*** - Optimal number of factors based on BIC

4.2 Estimation of stationary VAR model `(varfit.m)`

This function performs ordinary least squares estimation of a stationary VAR model. It is called by `fsvarproj.m` to compute the decoupled f-SVAR estimator by fitting a stationary factor-VAR to each regime. It is also called by `fsvarInit.m` to compute the sliding-window VAR coefficients.

```
[PI,V] = varfit(p,Y)
```

Inputs arguments: ***p*** - model order, ***Y*** – N x T time series data
Outputs arguments: ***PI*** - N x Np AR coefficient matrix
V - N x N covariance matrix

5. Examples

In this section, we provide an illustration of the toolbox's functionality with two examples respectively on simulated and real fMRI data.

5.1 Real fMRI data

In this example, f-SVAR Toolbox is applied to analyze directed connectivity in resting-state fMRI. We use fMRI data of three subject (from NITRC: NYU CSC TestRetest dataset session 1, https://www.nitrc.org/projects/nyu_trt.). The data was preprocessed and mean time series of 90 AAL-template-parceled ROIs were extracted and re-clustered into 6 resting-state networks with overlapping (SCN: sub-cortical network, AN: auditory network, SMN: sensorimotor network, VN: visual network, ATN: attentional network, DMN: default mode network). This results in a data matrix *Yr* with *N*=96, *T*=197, *Rs*=3.

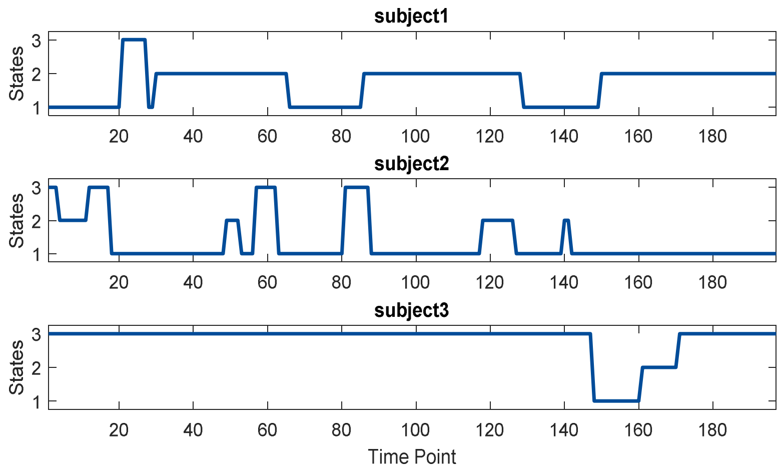
As an example, the following code excerpts from `example_fmri.m` implement the flows of the three-step estimation of f-SVAR model, and plot the results.

```
%--------------------------------------------------------------------%
%                      Estimation of f-SVAR Model                    %
%--------------------------------------------------------------------%
kmax = 10; % maximum range of canditate r
[r, ~, bic] = fmselr(Y,kmax); % Select no. of factors by BIC

opts.p = 1;      % VAR model order
opts.K = 3;      % Number of states
opts.r = r;     % Number of factors
opts.wlen = 15; % win length
opts.shift = 1; % win shift
opts.eps = 0.0001; % Stop EM if eps<like improvement
opts.ItrNo = 15; % maximum EM iterations

% Step 1: Initialization based on a factor model
[fsvar0,fm] = fsvarInit(Y,T,opts);

% Step 2: EM estimation of regime-switching factor model
[fsvar,Path,L] = fsvarest(Yr,fsvar0,opts);

% Step 3: Projection to high-dim connectivity states
[varmat] = fsvarproj(Yr,fsvar,Path,fm,opts);
```

Plot most likely state sequence estimated by switching Kalman smoother

```
figure('Name','State sequence');
for s =1:no_subs
    subplot(no_subs,1,s);
    plot(1:1:Tr,Path.St_sks(:,s)); hold on;
end
```

Figure: State sequence



Plot estimated VAR connectivity matrix at each state

```
figure('Name','Coupled f-SVAR Net');
for j=1:K
    subplot(1,K,j);
    imagesc(squeeze(varmat.Aco(:,:,j))); end

figure('Name','Decoupled f-SVAR Net');
for j=1:K
    subplot(1,K,j);
    imagesc(squeeze(varmat.Ade(:,:,j)));end

figure('Name','Significant Decoupled f-SVAR Net');
for j=1:K
    subplot(1,K,j);
    imagesc(squeeze(varmat.Ade_sig(:,:,j))); end
```

Figure: Coupled f-SVAR Net

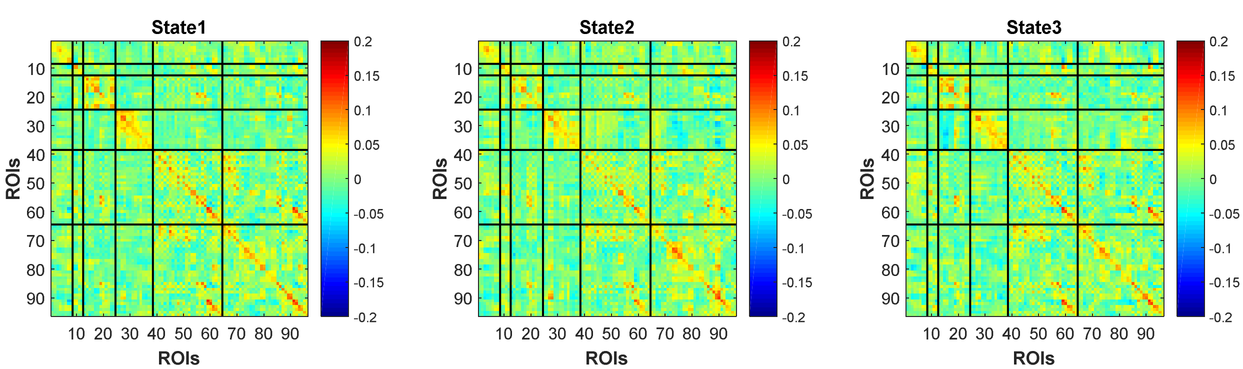


Figure: Decoupled f-SVAR Net

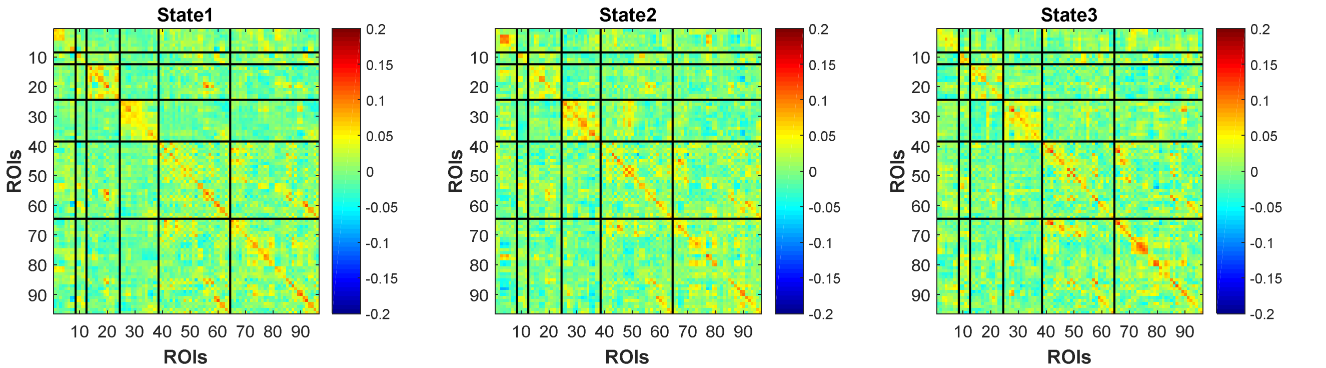
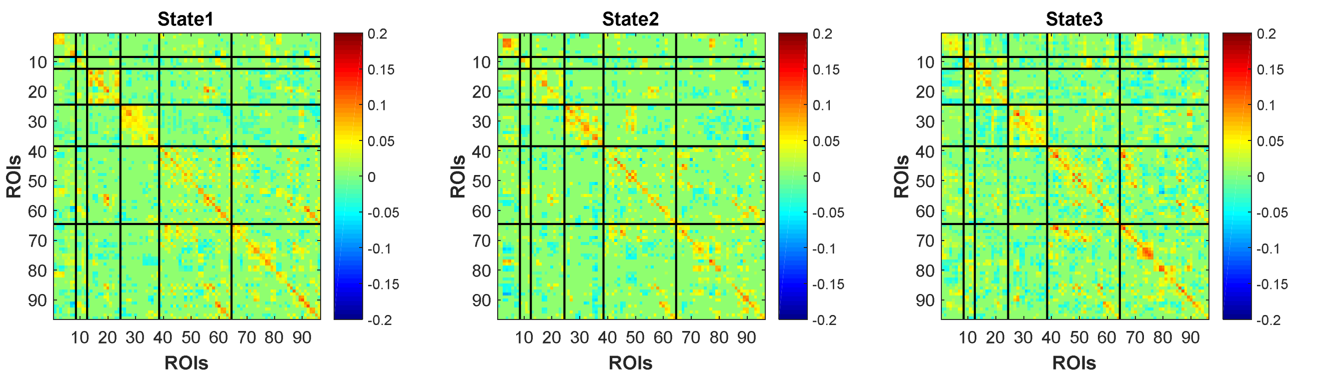
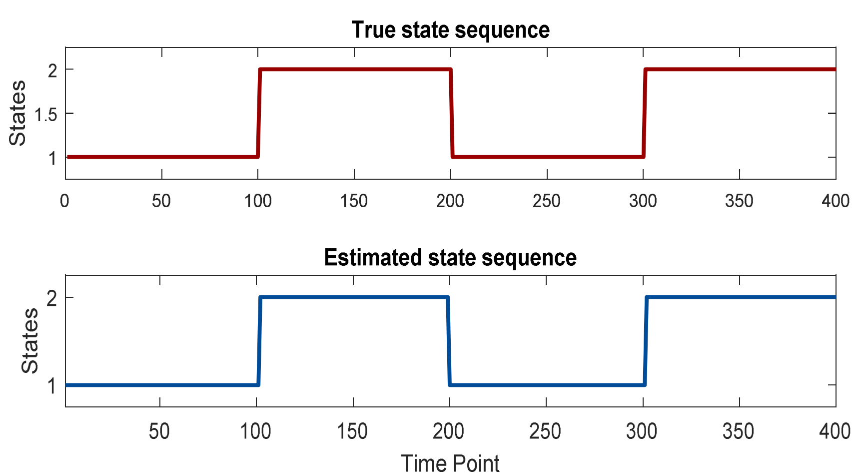


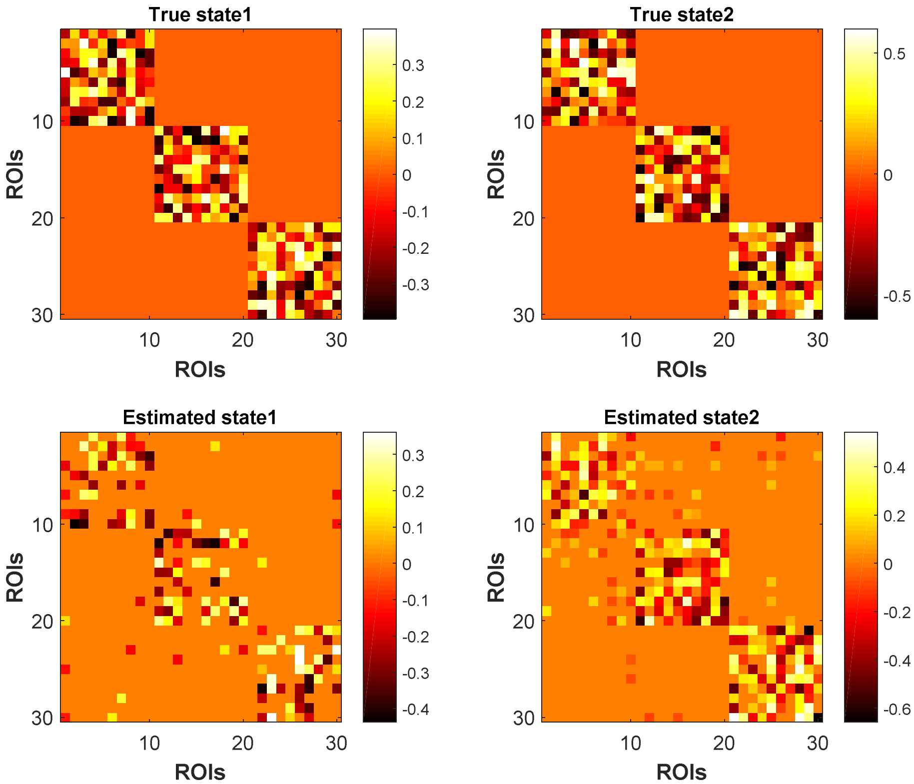
Figure: Significant Decoupled f-SVAR Net



5.2 Simulation

In the example `example_simulation.m`, f-SVAR is applied to simulated high-dimensional data of 2-state regime-switching VAR(1) with block-diagonal VAR structure to emulate modularity of brain network. Each state has VAR coefficient matrix with distinct strength of connections, i.e. low and high connectivity for state 1 and 2 respectively. The AR coefficients are randomly drawn from uniform distribution. The goal is to recover the true state sequence and the simulated coefficient matrix in each state.





References

1. C.-M. Ting, H. Ombao, S. B. Samdin and Sh-H. Salleh, “Estimating dynamic connectivity states in fMRI using regime-switching factor models,” *IEEE Trans. Med. Imag.,* 2017.
2. S. B. Samdin, C.-M. Ting, H. Ombao, and Sh-H. Salleh, “A unified estimation framework for state-related changes in effective brain connectivity,” *IEEE Trans. Biomed. Eng*., vol. 64, no. 4, pp. 844–858, 2017.
3. C.-M. Ting, A.-K. Seghouane, S.-H. Salleh, and A. M. Noor, “Estimating effective connectivity from fMRI data using factor-based subspace autoregressive models,” *IEEE Sig. Process. Lett*., vol. 22, no. 6, pp. 757–761, 2014.
4. K. P. Murphy, “Switching Kalman filters,” Tech. Rep., UC Berkeley,1998.
5. J. Bai, “Inferential theory for factor models of large dimensions,” *Econometrica*, vol. 71, no. 1, pp. 135–171, 2003.
6. C. Lam, Q. Yao, and N. Bathia, “Estimation of latent factors for high-dimensional time series,” *Biometrika,* vol. 98, no. 4, pp. 901–918, 2011.