Dynamic Factor Models in Real Time: A New Approach and an Implementation in Matlab

Jan Brůha

Moderní nástroje pro finanční analýzu a modelování
Praha, 5 June 2014
Outline of the presentation

1. Motivation
2. My contribution
3. Implementation in Matlab
   - Simulation Study
Motivation

Stock and Watson (2011)

The premise of dynamic factor models (DFM) is that a few latent dynamic factors $f_t$ drive the comovements of high-dimensional vector of time-series variables $X_t$, which is also affected by a vector of disturbances $\varepsilon_t$:

$$X_t = \sum_{i} \Lambda_i f_{t-i} + \varepsilon_t$$

$\equiv \chi_t = \text{common component}$
Applications

Dynamic factor models have a wide range of applications:

- Near-term forecasting
- Analysis of macroeconomic comovements
- Analysis of commodity markets
- Finance: yield curve
- and more ....

Stock and Watson (2011) distinguish several generations of DFM
First generation

First generation DFM have been estimated in time-domain using MLE

\[ X_t = \Lambda_0 f_t + e_t \]
\[ f_t = \Psi_1 f_{t-1} + \ldots \Psi_j f_{t-j} + G \eta_t, \]

By the parametric specification of distribution for \( \eta_t \) (Gaussian iid errors) and \( e_t \) (usually Gaussian stationary process), the model can be converted to a state space form and the likelihood function evaluated using the Kalman filter:

- static nature:
  - no lead-lag relation among \( X_t \)
  - not always plausible, e.g. the unemployment cycles lags the output cycles in most advanced countries (Brůha & Polanský, 2014; Andrle, Brůha, Solmaz 2013, 2014).
Non-parametric ‘averaging’ errors:

- (Generalized) principal component estimation
  \[
  \min_{f_1, \ldots, f_T, \Lambda} T^{-1} \sum_{t=1}^{T} (X_t - \Lambda f_t) \Sigma_e^{-1} (X_t - \Lambda f_t)',
  \]
- Leads to the eigenvalue decomposition of \( \Sigma_e^{-1/2} \Sigma_x \Sigma_e^{-1/2} \),
- Weak assumptions on dgp for \( f_t \)
- Standard PCA if \( \Sigma_e \) is replaced by the identity matrix (Bai & Ng, 2002, 2003)
- Hard to estimate precisely \( \Sigma_e \)
  - Forni, Halli, Lippi and Reichlin (2005) suggest estimating it by non-parametric spectral density (a dynamic model)
- Again static model, even if \( \Sigma_e \) estimated from a dynamic model
Second generations /b

**Dynamic** principal component analysis:

- Based on the frequency-domain PCA by Brillinger (1964)
- Introduced to empirical economics by Forni, Halli, Lippi and Reichlin (2000)
- A non-parametric method:
  - weak assumption on data generating process (rational spectral density),
  - the spectral density can be estimated by a non-parametric (Bartlett) approach.
- The spectral density is subjected to PCA and then transformed back to time-domain filter using a two-sided linear filter
The transformation to time domain leads to a two-sided filter:

\[ \chi_t = \sum_{i=-K}^{K} \omega_k X_{t-k}, \]

where \( \chi_t \) is the common component (driven by the common factors), and \( \omega_k \) are filtered weights.

Implications:

- Forni, Halli, Lippi and Reichlin (2000) call it ‘two-sided’ DFM
  - versus the ‘static’ one-sided DFM based on generalized PCA by Forni, Halli, Lippi and Reichlin (2005) outlined above
- The common component \( \chi_t \) cannot be estimated at the beginning and at the end of the sample
- Problems for many applications (real-time data)
- **My contribution:** how to estimate the common component on the whole sample.
Third generation – state space models

They can be applied to a variety of parametric models using sophisticated tools of modern econometrics, such as

▶ the EM algorithm (Bańbura & Modugno, 2010),
▶ the Gibbs sampler (Bai & Wang, 2012),
▶ the two-step estimation (Doz, Giannone and Reichlin, 2006).

These models:

▶ are truly dynamic (unlike the first generations and one-sided PCA),
▶ can accommodate various issues (missing data, asynchronous data release, ...),
▶ the common component estimated on the whole sample,
▶ but are parametric.
Is it possible to propose:

- a *genuinely* dynamic model,
- based on non-parametric (*frequency-domain*) approach,
- with the common component estimable on the *whole sample*,
- and possibly accommodating *other issues*, such as asynchronous data releases?

... and this is my contribution.
How is it done?

Start with the common component representation:

\[
X_t = \chi_t + \varepsilon_t, \\
\equiv \sum_i \lambda_i f_{t-i}
\]

Now consider the linear projection of \( \chi_t \) on \( X_S \equiv \{X_s\}_{s \in S} \), where \( S \) is an arbitrary index set. The linear projection is given as:

\[
E^* [\chi_t | X_S] = E \left[ \chi_t X_S^T \right] E \left[ X_S X_S^T \right]^{-1} X_S,
\]

Matrices such as \( E \left[ \chi_t X_S^T \right] \) can be derived from the spectral density of the data:

- as in ‘second-generation’ DFM (DPCA)
- hence, frequency-domain DPCA meets the linear regression.
How to get the common component on the whole sample?

The adaptation of the index set $S$:

- The index set can be adapted to the beginning and end of the sample (hence, real time data).
  - assume you wish to get $\mathbb{E}^* [\chi_t | \{X_{t-1}, X_t, X_{t+1}\}]$
  - trivial in the middle of the sample,
  - replace it by $\mathbb{E}^* [\chi_T | \{X_{T-1}, X_T\}]$ at the end of the sample!

- The index set can be adapted for missing data (e.g. due to asynchronous data release).
Finite-sample properties

The estimation of covariance matrices $\mathbb{E} [\chi_t X_S^T]$ and $\mathbb{E} [X_S X_S^T]$ can be imprecise in finite samples.

How to overcome this complication:

- Use robust regression in computing the projection $\mathbb{E}^* [\chi_t | X_S]$!
- I use a ridge regression with good properties.
Matlab codes

I build a set of Matlab routines that can be used to run a set of non-parametric DFM

sfactor.m computes the static PCA

dfactor.m computes the ‘two-sided’ version of the DPCA
  ▶ for particular choices, equivalent to the ‘one-sided’ DFM representation,

tfactor.m implements the approach outlined in this presentation.

I plan to put these routines on Matlab Exchange File soon
A simulation study

Forni, Halli, Lippi and Reichlin (2005) proposed four computational experiments with DFM.

**M1** read as

\[ x_{it} = \lambda_i f_t + \alpha c_i \varepsilon_{it} \text{ with } (1 - 0.5L)f_t = u_t, \]

and the shocks \( \varepsilon_{it}, u_t, \lambda_i \) are independent standard normal variables, \( c_i \sim \mathcal{U}[0.1, 1.1] \) and \( \alpha \) is calibrated so that the average idiosyncratic-common variance ratio is 1.

**M2** reads as:

\[
\begin{align*}
    x_{it} &= \sum_{k=0}^{3} a_{ik} u_{1,t-k} + \sum_{k=0}^{3} b_{ik} u_{2,t-k} + \alpha c_i \varepsilon_{it}, \\
\end{align*}
\]

where \( a_{ik}, b_{ik} \) and the shocks are standard normal variables and \( c_i \) and \( \alpha \) are as above.
M3 reads as:

\[ x_{it} = \sum_{k=l_i}^{l_i+2} \lambda_{k-l_i,i} f_{t-k} + \xi_{it}, \]

where \((1 - .5L)f_t = u_t\), \(\xi_{it} = \alpha c_i (\varepsilon_{it} + \varepsilon_{i+1t})\), \(u_t\) and \(\varepsilon_{it}\) are independent standardized normal random variables,

\[
l_i = \begin{cases} 
0 & \text{for } i \leq m \\
1 & \text{for } i \in \{m + 1, \ldots, 2m\} \\
2 & \text{for } i \geq 2m + 1 
\end{cases}
\]

M4 is the same as M3, but without cross-sectional dependence \((\xi_{it} = \alpha c_i \varepsilon_{it})\).
Model M1 – known covariance matrices

Marginal efficiency gains for this model

![Graph showing in-sample MSE and prediction MSE](image-url)
Model M1 – unknown covariance matrices

Gains disappear if covariance matrices are to be estimated

---

**In sample MSE**

<table>
<thead>
<tr>
<th>Time</th>
<th>Mean square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>T/2</td>
<td>1.5</td>
</tr>
<tr>
<td>T-1</td>
<td>1.5</td>
</tr>
<tr>
<td>T</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Prediction MSE**

<table>
<thead>
<tr>
<th>Prediction horizon (T+h)</th>
<th>Mean square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>1.5</td>
</tr>
</tbody>
</table>

- Forni et al (2005)
- LS projection (K=0)
- LS projection (K=1)
- LS projection (K=2)
- LS projection (K=3)
Model M3 – known covariance matrices

Large efficiency gains at the beginning of the sample

In sample MSE

Prediction MSE

Forni et al (2005)
LS projection (K=0)
LS projection (K=1)
LS projection (K=2)
LS projection (K=3)
Model M3 – unknown covariance matrices

The efficiency gains survive even in small samples ($T=50$, $N=20$)
The tool presented here was partly developed under the CNB research project B1/10. Nevertheless, the views expressed here are mine and do not necessarily reflect the position of the Czech National Bank.

I thank to Michal Andrle for comments and encouragement. However, any errors are solely my own responsibility.
Thank you for your attention.

jan.bruha@cnb.cz