

A replication of “A quasi-maximum likelihood approach for large, approximate dynamic factor models” (Review of Economics and Statistics, 2012)

Riccardo Lucchetti and Ioannis A. Venetis

Abstract

The authors replicate and extend the Monte Carlo experiment presented in Doz, Giannone and Reichlin (A Quasi-Maximum Likelihood Approach For Large, Approximate Dynamic Factor Models, *Review of Economics and Statistics*, 2012) on alternative (time-domain based) methods for extracting dynamic factors from large datasets; they employ open source software and consider a larger number of replications and a wider set of scenarios. Their narrow sense replication exercise fully confirms the results in the original article. As for their extended replication experiment, the authors examine the relative performance of competing estimators under a wider array of cases, including richer dynamics, and find that maximum likelihood (ML) is often the dominant method; moreover, the persistence characteristics of the observable series play a crucial role and correct specification of the underlying dynamics is of paramount importance.

(Replication Study)

JEL C15 C32 C55 C87

Keywords Dynamic factor models; EM algorithm; Kalman filter; Principal components

Authors

Riccardo Lucchetti, DISES, Università Politecnica delle Marche, Ancona, Italy
Ioannis A. Venetis, ✉ University of Patras, Greece, ivenetis@upatras.gr

Citation Riccardo Lucchetti and Ioannis A. Venetis (2020). A replication of “A quasi-maximum likelihood approach for large, approximate dynamic factor models” (Review of Economics and Statistics, 2012). *Economics: The Open-Access, Open-Assessment E-Journal*, 14 (2020-14): 1–14. <http://dx.doi.org/10.5018/economics-ejournal.ja.2020-14>

1 Introduction

A central empirical finding from the use of dynamic factor models (DFMs) in applied macroeconomics is that a few factors can explain a large fraction of the total variance of many macroeconomic series. The factor structure is typically used for signal extraction and forecasting, although in some cases maximum likelihood-based methods can be used to test hypotheses of economic interest. Extensive surveys of the DFM literature can be found in Bai and Ng (2008), Stock and Watson (2011) and Stock and Watson (2016).

Maximum likelihood estimation for DFMs with a high-dimensional panel of time series becomes rapidly infeasible given the large number of parameters. However, in Doz et al. (2011) a solution was put forward to overcome this problem via a two-step hybrid approach that links the simplicity and speed of principal components to the efficiency of the Kalman smoother; later, Doz et al. (2012) refined this technique by considering quasi-maximum likelihood estimation where factor estimates are computed iteratively using the Kalman smoother on the state-space representation via the EM algorithm.

Both approaches are robust to cross-sectional misspecification, time-series correlation of the idiosyncratic components, and non-Gaussianity. Since then, state-space based maximum likelihood methods have been increasingly adopted in the applied literature, e.g. Luciani (2015) and Scotti (2016), among others. The asymptotic properties of EM estimation of these models were recently investigated in Barigozzi and Luciani (2019a, 2019b) who also provide simulation-based evidence.

In this paper, we replicate and extend the Monte Carlo experiment presented in Doz et al. (2012) on the suitability of ML for extracting dynamic factors from large datasets, by using different software, a larger number of replications and a wider set of scenarios. We find that most of the statements in the original article are supported, but the relative quality of full ML estimation, compared to the alternative procedures proposed in the same paper, could be substantial in some cases, whereas in other settings it may not be so decisive to outweigh the extra computational cost.

For our replication exercise, we used the DFM package for gretl (see Lucchetti and Venetis (2019)), which implements three estimators for DFMs, namely the benchmark principal components estimator (*PC*), the two-step Kalman smoother based estimator (*TS*) put forward by Doz et al. (2011) and the quasi-maximum likelihood EM based estimator (*ML*) by Doz et al. (2012).

The paper is organised as follows: section 2 describes the data generating process, section 3 offers a narrow sense replication of the Monte Carlo study in Doz et al. (2012) and section 4 extends (wide sense replication) the experiment, in a direction which should address common concerns of practitioners. Section 5 briefly concludes the paper by summarising our findings.

2 Setup of the experiment

The models that we consider can be written in state-space representation as

$$\begin{matrix} x_t \\ N \times 1 \end{matrix} = \Lambda_0 f_t + \Lambda_1 f_{t-1} + \cdots + \Lambda_s f_{t-s} + e_t = \chi_t + e_t \quad (1)$$

$$\begin{matrix} f_t \\ q \times 1 \end{matrix} = A_1 f_{t-1} + A_2 f_{t-2} + \cdots + A_p f_{t-p} + u_t \quad (2)$$

where x_t is a vector of N standardised observable variables¹ and f_t is the q -element vector of (unobserved) common dynamic factor; the term χ_t is the so-called “common component” and the shocks to the observation Equation (1), e_t , are known as the idiosyncratic component, and are assumed to be uncorrelated with f_t at all leads and lags. It is assumed that the elements of e_t are weakly correlated both cross-sectionally and serially, so that the factors f_t summarise the most relevant cross-covariance properties of the variables. Both processes f_t and e_t are assumed to be second-order stationary.

As for s , the lag order in the observation Equation (1), the original study we are replicating only considers a static version of the observation Equation (1) (that is, $s = 0$). Part of the added value of the present study is to remove this limitation in favour of the fully generalised version with $s \geq 0$.

The lag order s in Equation (1) is assumed to be finite here. A sizeable body of literature (mainly by Forni, Hallin, Lippi and several co-authors) has analysed the so-called *generalised* DFM, in which this assumption is relaxed, and estimation is typically carried out via spectral methods. See Forni et al. (2000, 2015, 2017).

Although most of the literature on DFMs (including the article we are replicating) pays little attention to the issue of identification of the model above, it should be mentioned that the reduced form of Equations (1)–(2) is a VARMA process, for which notoriously many observationally-equivalent representations exist. A comprehensive analysis of the identification problem in the context of DFMs, with different approaches to its solution, is given in Bai and Wang (2015). However, while the existence of several observationally-equivalent parametrisations for the same DGP are a serious problem for parameter estimation, forecasting is not affected by the existence of several observationally equivalent parametrisations (see for example Lütkepohl (2006), page 290). Since our focus is on the recovery of the space spanned by the dynamic factors f_t , the under-identification problem is not an issue.

In accordance with Doz et al. (2012), we consider the following data generating process in our Monte Carlo study: the idiosyncratic shocks were generated as

$$e_t = De_{t-1} + v_t, \quad v_t \sim N.i.d(\mathbf{0}_N, \mathcal{T}) \quad (3)$$

where D is a diagonal matrix.² We call d_i its i -th diagonal entry. As for the matrix \mathcal{T} ,

$$\mathcal{T}_{i,j} = \tau^{|i-j|} (1 - d_i d_j) \sqrt{\gamma_i \gamma_j} \quad (4)$$

$$\gamma_i = \frac{\beta_i}{1 - \beta_i} \text{Var}(\chi_{i,t}) \quad (5)$$

where $\chi_{i,t} = \lambda'_{0,i} f_t + \dots + \lambda'_{s,i} f_{t-s}$ denotes the i^{th} common component. The Toeplitz matrix \mathcal{T} is the covariance matrix of the vector v_t , so the parameter τ controls for the amount of cross-correlation in the idiosyncratic errors, $\text{Corr}(e_{i,t}, e_{j,t}) = \tau^{|i-j|}$. The parameter $\beta_i = \frac{\text{Var}(e_{i,t})}{\text{Var}(\chi_{i,t})}$

¹ This hypothesis is of course not restrictive, since a constant term could be added to Equation (1), and standardisation is only necessary if the variables are expressed in different units.

² This choice was made for conformity to the original article. In principle, since the EM-ML estimator we are discussing here can be seen as a QMLE estimator, one could make different distributional assumptions, especially allowing for leptokurtic innovations. Tables A7 and A8 in the online appendix contain results of our experiment with idiosyncratic component errors drawn independently from the t_4 distribution. We found no substantial changes.

denotes noise variance proportion for the i -th variable and is drawn from the uniform distribution $\beta_i \sim i.i.d. U([u, 1 - u])$. When $u = 0.5$, “observables” $x_{i,t}$ have cross-sectionally homoskedastic idiosyncratic components. The exact factor model corresponds to the case $\tau = 0$ and $D = 0$. The elements of the loadings matrices Λ_i were drawn from a standard Normal distribution.³ The above data generating process is considerably broader than the one adopted by Doz et al. (2012), which can be derived as a special case of Equations (1)-(5). Similar experiments have also been performed by Breitung and Tenhofen (2011) and Bai and Li (2016).⁴

The aim of the experiment is to carry out a comparison between estimators on their ability to reconstruct the latent factors. As is well known, estimated factors (and factor loadings) are rotations of the underlying latent factors (and underlying latent loadings respectively), so the comparison will be based on a statistic that measures the closeness of the space spanned by the “true” simulated factors f_t and their estimators \hat{f}_t . The three estimators considered are principal components (PC), the two-step estimator (TS) and the maximum likelihood estimator (ML). In the latter two cases, by the term “estimator”, we refer to the smoothed estimates of the state vector f_t in (2)).

If we denote by \mathbf{F} and $\hat{\mathbf{F}}$ the matrices holding f_t and \hat{f}_t , respectively, the statistic we use is the trace R^2 , that can be written as,⁵

$$TR = \frac{\text{tr}[\mathbf{F}'\hat{\mathbf{F}}(\hat{\mathbf{F}}'\hat{\mathbf{F}})^{-1}\hat{\mathbf{F}}'\mathbf{F}]}{\text{tr}(\mathbf{F}'\mathbf{F})} \quad (6)$$

The notation \mathbf{F} and $\hat{\mathbf{F}}$ is not to be mistaken for the static factors matrix. For $s = 0$, matrix \mathbf{F} holds simulated f_t while $\hat{\mathbf{F}}$ holds either the $r = q$ static factors $\hat{f}_{t,PC}$ estimated by PC or the q dynamic factors $\hat{f}_{t,TS}$ or $\hat{f}_{t,ML}$ depending on the estimation method. For $s > 0$ matrix \mathbf{F} holds $f_t, f_{t-1}, \dots, f_{t-s}$ while $\hat{\mathbf{F}}$ holds the $r = q(s+1)$ static factors estimated by PC or $\hat{f}_t, \hat{f}_{t-1}, \dots, \hat{f}_{t-s}$ when TS or ML estimation has been employed.

The TR statistic tends to 1 as the canonical correlation between estimated and true factors increases. On the other end, values of TR close to 0 indicate high discrepancies between the space spanned by the actual and the estimated factors. Thus, it is an appropriate quantitative measure since common factors are identified only up to a rotation.⁶

In addition, as an anonymous reviewer pointed out, a measure which is equally important to report is the MSE between the estimated common component and the simulated one since it is not affected by identification issues. At each replication $b = 1, \dots, 5000$, we compute the following

³ Again, this choice was made for conformity to the original article. As noted by Poncela and Ruiz (2020), drawing from other distributions such as the Uniform distribution, so as to ensure that the factors are truly pervasive, could perhaps be more appropriate. An online appendix offers a table with more elaborate factor loadings for the case $s > 0$ but we find no significant differences from the normal distribution case. The same is true for results pertaining to the dynamically simpler case of $s = 0$.

⁴ In Forni et al. (2018) a comparison between alternative modelling techniques is also performed, but with a focus on forecasting.

⁵ It satisfies $0 \leq TR \leq 1$.

⁶ We are using this measure of the ability to reconstruct the factor space in accordance with the original article, but we would like to stress that this is a purely descriptive statistic and we are not in the position to judge if differences between methods are statistically significant.

standardized mean squared errors (MSEs),

$$MSE_{method}^{(b)} = \frac{\sum_{i=1}^N \sum_{t=1}^T (\chi_{i,t} - \hat{\chi}_{i,t}^{method})^2}{\sum_{i=1}^N \sum_{t=1}^T \chi_{i,t}^2} \quad (7)$$

and then we calculate the average relative MSE of ML over either method=PC or TS or Methods 4 and 5 as reported in the extended replication section

$$MSE_{rel} = \frac{1}{B} \sum_{b=1}^B \frac{MSE_{ML}^{(b)}}{MSE_{method}^{(b)}} \quad (8)$$

3 Replication of the original results

The first exercise we perform is a narrow sense replication exercise, in which the relative performance of three different methods for factor extraction is investigated.

The original experiment assumed $s = 0$ and $p = 1$ as well as $A_1 = \alpha \cdot \mathbf{I}_q$, $D = d \cdot \mathbf{I}_N$ with α , d being scalar-valued. The parameter u is set to 0.1 in all experiments, so that the idiosyncratic components e_t can have substantially different variances and the observable series x_t exhibit different signal-to-noise ratios.

The three Tables 2, 4, 6 present a reproduction of all Monte Carlo results from the original study of Doz et al. (2012). The original study's results are shown in Tables 1, 3, 5 and are placed side-by-side in all tables with our replicating study's corresponding estimates. The data generation process (DGP) is exactly the same as that of the original paper, the only difference is that we ran the Monte Carlo experiment with 5000 replications, instead of 500.

The tables contain several subtables, each of which has different values of T (the sample size) for each row and of N (the number of observable variables) for each column. The first subtable contains the average of the TR statistic for the ML estimator and the second one the average number of EM iterations needed to achieve convergence, as a rough proxy for computational speed. The remaining three subtables display the relative performance of the PC and TS estimators compared to the ML estimator, measured by the relative trace ratios TR_{ML}/TR_{PC} , TR_{ML}/TR_{TS} .⁷

As can be seen, results are qualitatively identical and quantitatively very close to the original article: the maximum likelihood estimator appears to have a slight edge on the other two methods, especially so when the idiosyncratic disturbances e_t are closer to the “ideal” factor models (that is, both their cross-sectional correlation and autocorrelation are 0). It is noteworthy that the relative advantage of ML tends to vanish as either the time dimension T or the cross-sectional dimension N increase. Moreover, the extra computational cost for the extraction of factors via ML may be substantial (especially for small samples).

⁷ We dropped results with (all three methods') individual $TR < 0.05$ to avoid spuriously large trace ratios. Notice that the “failure percentages” were very few compared to the 5000 simulations in each case considered (see Table B1 of the online readme file). Further, we ensure that initial conditions satisfy stationarity restrictions.

Table 1: Doz et al. (2012, p. 1019) Table 1

	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}				
$T = 50$	0.52	0.68	0.74	0.75	0.76
$T = 100$	0.64	0.78	0.84	0.85	0.86
	Number of iterations				
$T = 50$	13	9	5	5	3
$T = 100$	13	7	4	4	3
	TR_{ML}/TR_{PC}				
$T = 50$	1.11	1.04	1.00	1.00	1.00
$T = 100$	1.09	1.02	1.01	1.00	1.00
	TR_{ML}/TR_{TS}				
$T = 50$	1.03	1.01	1.00	1.00	1.00
$T = 100$	1.02	1.00	1.00	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1019, Table 1).

Table 2: Replication of original Table 1 in Doz et al. (2012)

	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}				
$T = 50$	0.60	0.67	0.71	0.74	0.74
$T = 100$	0.72	0.78	0.82	0.84	0.84
	Number of iterations				
$T = 50$	10	6	4	3	3
$T = 100$	7	4	3	3	3
	TR_{ML}/TR_{PC}				
$T = 50$	1.03	1.01	1.00	1.00	1.00
$T = 100$	1.05	1.02	1.01	1.00	1.00
	TR_{ML}/TR_{TS}				
$T = 50$	0.99	0.99	1.00	1.00	1.00
$T = 100$	1.00	1.00	1.00	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 1$.

Table 3: Doz et al. (2012, p. 1019) Table 2

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}			
$T = 50$	0.48	0.59	0.65	0.67
$T = 100$	0.58	0.75	0.80	0.82
	Number of iterations			
$T = 50$	26	12	7	5
$T = 100$	20	9	5	4
	TR_{ML}/TR_{PC}			
$T = 50$	1.08	1.05	1.03	1.01
$T = 100$	1.10	1.06	1.02	1.01
	TR_{ML}/TR_{TS}			
$T = 50$	1.05	1.02	1.01	1.00
$T = 100$	1.07	1.03	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1019, Table 2).

Table 4: Replication of original Table 2 in Doz et al. (2012)

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}			
$T = 50$	0.53	0.63	0.66	0.69
$T = 100$	0.64	0.76	0.80	0.82
	Number of iterations			
$T = 50$	22	9	5	4
$T = 100$	15	6	4	3
	TR_{ML}/TR_{PC}			
$T = 50$	1.08	1.03	1.01	1.00
$T = 100$	1.12	1.03	1.01	1.01
	TR_{ML}/TR_{TS}			
$T = 50$	1.03	1.00	1.00	1.00
$T = 100$	1.05	1.00	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 3$.

Table 5: Doz et al. (2012, p. 1020) Table 3

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}			
$T = 50$	0.54	0.65	0.68	0.70
$T = 100$	0.66	0.78	0.81	0.82
	Number of iterations			
$T = 50$	21	9	6	5
$T = 100$	15	7	5	4
	TR_{ML}/TR_{PC}			
$T = 50$	1.14	1.06	1.03	1.01
$T = 100$	1.19	1.06	1.02	1.01
	TR_{ML}/TR_{TS}			
$T = 50$	1.07	1.02	1.01	1.00
$T = 100$	1.10	1.01	1.00	1.00

Notes: The original results as reported in Doz et al. (2012, p. 1020, Table 3).

Table 6: Replication of original Table 3 in Doz et al. (2012)

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
	TR_{ML}			
$T = 50$	0.56	0.65	0.68	0.70
$T = 100$	0.70	0.78	0.80	0.82
	Number of iterations			
$T = 50$	21	8	5	4
$T = 100$	13	5	4	3
	TR_{ML}/TR_{PC}			
$T = 50$	1.16	1.07	1.03	1.01
$T = 100$	1.20	1.07	1.02	1.01
	TR_{ML}/TR_{TS}			
$T = 50$	1.08	1.02	1.00	1.00
$T = 100$	1.10	1.01	1.00	1.00

Notes: Data generation process input set at: $s = 0$, $p = 1$, $\alpha = 0.9$, $d = 0$, $\tau = 0$, $u = 0.1$, $q = 3$.

4 Extended replication

In this section, we extend the original Monte Carlo experiment so as to shed more light on the relative properties of the three factor extraction methods considered in the previous section.

The additional set of simulations we run can be motivated as follows: in most cases, the interest of the practitioner is in using the estimated factors as a by-product for a model of interest, that can be used for forecasting/nowcasting purposes, structural analysis or more.

In a time-series context (especially at higher frequencies), it is very likely that the factors may affect the observables in a lag-distributed pattern, so the proper value of s in Equation (1) is larger than 0. As is well known (for example, see Bai and Ng (2007)), Equation (1) can be re-written in a “static” form

$$x_t = \mathbf{\Lambda} F_t + e_t,$$

where the vector F_t contains f_t and its lags, suitably stacked, and $\mathbf{\Lambda}$ is the horizontal concatenation of the Λ_i matrices. The dimension of the F_t vector is $q \cdot (s + 1)$. The interest of the practitioner is arguably to use a method that reconstructs as closely as possible the space spanned by F_t . The possibilities we investigate are

1. compute the q -vector \hat{f}_t via ML and then stack its lags to form \hat{F}_t ;
2. compute the q -vector \hat{f}_t via TS and then stack its lags to form \hat{F}_t ;
3. compute \hat{F}_t by extracting the first $q \cdot (s + 1)$ principal components;
4. approximate \hat{F}_t via the first principal component;
5. approximate \hat{F}_t via the first principal component and its lags.

In principle, Methods 4 and 5 may seem to be irrelevant, as in the present study the number of static factors is assumed to be known, so evaluating methods that are known *a priori* to be misspecified may seem unfair. However, we are interested in evaluating how serious this shortcoming is, in the light of the fact that in many cases it is customary for practitioners to use the first principal component as the “dominant” factor, regardless of any statistically oriented criterion, (see for example Bai and Ng (2007)).

In a way, the inclusion of Methods 4 and 5 gives an idea of the performance of different methods for correctly specified models, relative to the problems stemming from incorrect specification. Having said this, a proper analysis of the effects of misspecification of the lag orders would deserve a far more comprehensive treatment, that we leave for further research.

In order to initialise the algorithm, we use the method outlined in Lütkepohl (2014) to retrieve the dynamic factor space from principal components, that entails running a VAR(1) on estimated principal components and then performing PC extraction on the residuals. In practice:

1. given the PC estimates of the static factors \hat{F}_t , we estimate a VAR(1) via OLS:

$$\hat{F}_t = \Phi \hat{F}_{t-1} + U_t;$$

Table 7: Extended experiment

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
TR_{ML}				
$T = 50$	0.58	0.65	0.69	0.70
$T = 100$	0.70	0.79	0.81	0.82
Number of iterations				
$T = 50$	46	27	22	21
$T = 100$	25	15	13	12
TR_{ML}/TR_{TS}				
$T = 50$	1.57	1.25	1.12	1.11
$T = 100$	1.70	1.22	1.16	1.08
TR_{ML}/TR_{PC}				
$T = 50$	1.09	1.08	1.06	1.04
$T = 100$	1.13	1.09	1.05	1.03
$TR_{ML}/TR_{PC,1}$				
$T = 50$	1.76	1.65	1.62	1.59
$T = 100$	1.86	1.76	1.69	1.66
$TR_{ML}/TR_{PC,1,lags}$				
$T = 50$	1.59	1.54	1.52	1.50
$T = 100$	1.74	1.68	1.63	1.60

Notes: Comparing alternative dynamic factor extraction methods when $s = 1$. Data generation process input set at: $s = 1$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 2$.

2. we compute the covariance matrix of the residuals $\hat{\Omega} = T^{-1}\hat{U}'\hat{U}$;
3. we extract the q largest eigenvalues of $\hat{\Omega}$ and the corresponding eigenvectors \hat{V} ;
4. the preliminary estimate of the dynamic factors is then given by $\tilde{f}_t = \hat{V}'\hat{F}_t$.

We performed the experiment with a wide variety of alternative values for the DGP parameters; to be specific, we experimented with different values for α , d , q and s , thereby considering different scenarios on factor persistence, idiosyncratic error persistence and lag specifications in the observation equation; the full results are not tabulated here for the sake of conciseness, but are available upon request. The results for the experiment we chose to show, as representative of the whole set of results, are displayed in Table 7.

Table 8: Extended experiment. Relative MSE statistics

	$N = 10$	$N = 25$	$N = 50$	$N = 100$
MSE stat for ML				
$T = 50$	0.77	0.75	0.74	0.74
$T = 100$	0.76	0.75	0.75	0.74
Number of iterations				
$T = 50$	45	26	22	20
$T = 100$	24	15	13	12
relative MSE(ML/TS)				
$T = 50$	0.92	0.95	0.97	0.97
$T = 100$	0.91	0.95	0.96	0.97
relative MSE(ML/PC)				
$T = 50$	1.01	0.99	0.99	0.99
$T = 100$	1.00	1.00	1.00	1.00
relative MSE(ML/PC,1)				
$T = 50$	0.93	0.91	0.90	0.89
$T = 100$	0.92	0.90	0.89	0.88
relative MSE(ML/PC,1,lags)				
$T = 50$	0.94	0.92	0.91	0.90
$T = 100$	0.92	0.90	0.89	0.89

Notes: Comparing alternative dynamic factor extraction methods using relative MSE calculations when $s = 1$. Data generation process input set at: $s = 1$, $p = 1$, $\alpha = 0.9$, $d = 0.5$, $\tau = 0.5$, $u = 0.1$, $q = 2$.

As shown in Table 7, ML clearly dominates TS and all the methods based on a limited set of principal components. The only way to achieve a qualitatively similar approximation to the space spanned by the true factors is by using the full set of principal components, compared to which the advantage of using ML becomes smaller, especially for larger values of T and N .

The EM-based ML method exhibits a noticeably larger overhead in terms of computational complexity than in the original Monte Carlo experiment by Doz et al. (2012). This was to be expected, since the structure of the model we are considering is significantly more complex in terms of lag specification ($s = 1$ in Equation (1)).

Perhaps the relative superiority of PC to TS may be considered surprising at first sight. However, the two methods use a different number of factors, since PC uses $q \cdot (s + 1)$ factors and estimates consistently the factor space, whereas TS only uses q factors (plus their lags). Further, the first step

of the Doz et al. (2011) TS method consists of the factors estimated by PC, but their model is a static one (no lag in the measurement equation), which is not the case here. So this result should be taken as an indication that TS may perform rather poorly in finite samples when $s > 0$ and suitable initialization conditions should be devised to alleviate factor extraction issues for small cross-sections, at least when $N \leq 25$ and $T \leq 100$.

It is interesting to compare the three PC-based models under the viewpoint of misspecification: clearly, Method 3 (PC) is a correct way to estimate the full set of static factors, whereas Methods 4 and 5 (PC,1 and PC,1,lags) are not, in that they are based on a specification of Equation (1) where s or q are too small. It is evident — last two sub-panels of Table 7 — that the effects of misspecification are quite dramatic and larger values of T and N do not mitigate the shortcomings.

Table 8 reports the relative MSE (Rel-MSE) of the ML common component estimator over the TS, PC, PC,1 and PC,1,lags estimators. The ML estimator behaves very similarly to the PC estimator however there is a clear improvement in common components estimation in MSE terms when it comes to the remaining methods of common components estimation.

It would seem that, as long as your objective is the estimation of the common component χ_t , estimation of static factors via PC is just as good as ML (which, however, dominates TS). However, if you're interested in attaching some kind of meaning to the estimated factors f_t via some identification restrictions, then ML appears to be the method of choice.

Our set of extra results can be summarised as follows: the most important element to consider is persistence, either as a characteristic of latent factors or as the lag length in the observation Equation (1).⁸ When persistence is high, factors are estimated more accurately with maximum likelihood than any of the other methods we considered; the only exception is principal components for the full set of static factors, which on the other hand has the disadvantage of requiring a set of $q \cdot (s + 1)$ series instead of just q . Anyhow, the near-equivalence of ML and PC requires T and/or N to be rather large; in mid-sized samples ML retains a noticeable edge.

5 Conclusions and possible extensions

We use an independent software implementation to replicate the simulation results of Doz et al. (2012) regarding factor estimation in dynamic settings using the benchmark principal component method, a two step Kalman smoother based method and EM-based maximum likelihood estimation.

Our narrow sense replication exercise fully confirms the results in the original article. As for our extended replication experiment, we find that ML is the dominant method in a wide array of situations, notably when persistence is substantial. Those results go one step further than the closing remark by Doz et al. (2012) “...Efficiency improvements are relevant when the factor extraction is difficult, that is, when there are more common factors to estimate...”, by including a richer set of dynamic specifications and the possibility of misspecification.

The present work could be extended in several directions: for instance, the lag orders s and p and the number of dynamic factors q are assumed to be known and fixed *ex ante*; clearly, this

⁸ See also Table A7 in the online appendix where factors are constructed as “more pervasive” along with a distributed lag effect on observables, more elaborate persistence profile for the factors and different persistent characteristics on the idiosyncratic components.

choice was made for conformity to the experiment we are replicating and we acknowledge it is unrealistic and of limited usefulness for practitioners. It may be conjectured that different methods could display different performance if the lag orders have to be selected or if q has to be estimated through criteria such as the one put forward in Hallin and Liška (2007).

The urgency of such an extension is forcefully argued in Poncela and Ruiz (2020), in the light of another important theoretical question: simultaneous identification of p (the VAR order in the state transition Equation (2)) and s (the lag order of the observation Equation (1)) is an open issue that deserves further research. The purpose of our replication exercise is to extend the original results in Doz et al. (2012) by considering richer dynamics under the assumption of correct specification, but we acknowledge that near-underidentification could occur in finite samples for various values of p and s . How problematic this would be for the main purpose tackled here (the estimation of the space spanned by the dynamic factors) remains to be seen and will be the object of further research. However, it seems safe to conjecture that, as long as the object of interest is the space spanned by the dynamic factors f_t (as opposed to parameter point estimation), overestimating p and/or s should not lead to serious problems.

Another possible extension – for comparison purposes — could be using a DGP taken from some empirical exercise. These extensions, however, would broaden the present article's scope considerably and are left for future work.

Acknowledgements We would like to thank an anonymous reviewer and one of the original authors of the paper for their critical reading and helpful suggestions. We also thank Pilar Poncela and Esther Ruiz for their detailed discussion (<http://www.economics-ejournal.org/economics/discussionpapers/2020-7>) that led to improve our initial draft. Finally, we thank the lead Co-Editor for Replications: Prof. Bob Reed, for the careful review and constructive suggestions that not only improved our draft but also significantly helped us to revise the manuscript. Needless to say, all remaining errors and inaccuracies are ours.

References

- Bai, J., and Li, K. (2016). Maximum likelihood estimation and inference for approximate factor models of high dimension. *The Review of Economics and Statistics*, 98(2): 298–309. URL https://doi.org/10.1162/REST_a_00519.
- Bai, J., and Ng, S. (2007). Determining the number of primitive shocks in factor models. *Journal of Business & Economic Statistics*, 25(1): 52–60. URL <https://doi.org/10.1198/073500106000000413>.
- Bai, J., and Ng, S. (2008). Large dimensional factor analysis. *Foundations and Trends in Econometrics*, 3(2): 89–163. URL <http://dx.doi.org/10.1561/08000000002>.
- Bai, J., and Wang, P. (2015). Identification and estimation of dynamic factor models. *Journal of Business & Economic Statistics*, 33(2): 221–240. URL <https://doi.org/10.1080/07350015.2014.941467>.
- Barigozzi, M., and Luciani, M. (2019a). Quasi maximum likelihood estimation and inference of large approximate dynamic factor models via the EM algorithm. *arXiv preprint arXiv:1910.03821*. URL <https://arxiv.org/pdf/1910.03821>.
- Barigozzi, M., and Luciani, M. (2019b). Quasi maximum likelihood estimation of non-stationary large approximate dynamic factor models. *arXiv preprint arXiv:1910.09841*. URL <https://arxiv.org/pdf/1910.09841>.
- Breitung, J., and Tenhofen, J. (2011). GLS estimation of dynamic factor models. *Journal of the American Statistical Association*, 106(495): 1150–1166. URL <https://doi.org/10.1198/jasa.2011.tm09693>.

- Doz, C., Giannone, D., and Reichlin, L. (2011). A two-step estimator for large approximate dynamic factor models based on Kalman filtering. *Journal of Econometrics*, 164(1): 188–205. URL <https://doi.org/10.1016/j.jeconom.2011.02.012>.
- Doz, C., Giannone, D., and Reichlin, L. (2012). A quasi-maximum likelihood approach for large approximate dynamic factor models. *The Review of Economics and Statistics*, 94(4): 1014–1024. URL https://www.mitpressjournals.org/doi/pdf/10.1162/REST_a_00225.
- Forni, M., Giovannelli, A., Lippi, M., and Soccorsi, S. (2018). Dynamic factor model with infinite-dimensional factor space: Forecasting. *Journal of Applied Econometrics*, 33(5): 625–642.
- Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). The generalized dynamic-factor model: Identification and estimation. *The Review of Economics and Statistics*, 82(4): 540–554. URL <https://www.jstor.org/stable/2646650>.
- Forni, M., Hallin, M., Lippi, M., and Zaffaroni, P. (2015). Dynamic factor models with infinite-dimensional factor spaces: One-sided representations. *Journal of Econometrics*, 185(2): 359–371. URL <https://doi.org/10.1016/j.jeconom.2013.10.017>.
- Forni, M., Hallin, M., Lippi, M., and Zaffaroni, P. (2017). Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis. *Journal of Econometrics*, 199(1): 74–92. URL <https://doi.org/10.1016/j.jeconom.2017.04.002>.
- Hallin, M., and Liška, R. (2007). Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478): 603–617. URL <https://doi.org/10.1198/016214506000001275>.
- Lucchetti, R., and Venetis, I. A. (2019). Dynamic factor models in gretl. The DFM package. gretl working papers 7, Università Politecnica delle Marche (I), Dipartimento di Scienze Economiche e Sociali. URL <http://docs.dises.univpm.it/web/quaderni/pdfgretl/gretl007.pdf>.
- Luciani, M. (2015). Monetary policy and the housing market: A structural factor analysis. *Journal of Applied Econometrics*, 30(2): 199–218. URL <https://doi.org/10.1002/jae.2318>.
- Lütkepohl, H. (2006). Forecasting with VARMA models. In G. Elliott, C. Granger, and A. Timmermann (Eds.), *Handbook of Economic Forecasting*, volume 1 of *Handbook of Economic Forecasting*, chapter 6, pages 287–325. Elsevier. DOI [https://doi.org/10.1016/S1574-0706\(05\)01006-2](https://doi.org/10.1016/S1574-0706(05)01006-2).
- Lütkepohl, H. (2014). Structural vector autoregressive analysis in a data rich environment: A survey. Discussion Papers of DIW Berlin 1351, DIW Berlin, German Institute for Economic Research. URL <https://ideas.repec.org/p/diw/diwwpp/dp1351.html>.
- Poncela, P., and Ruiz, E. (2020). A comment on the dynamic factor model with dynamic factors. *Economics Discussion Papers, No 2020-7*, Kiel Institute for the World Economy. URL <http://www.economics-ejournal.org/economics/discussionpapers/2020-7>.
- Scotti, C. (2016). Surprise and uncertainty indexes: Real-time aggregation of real-activity macro-surprises. *Journal of Monetary Economics*, 82: 1–19. URL <https://doi.org/10.1016/j.jmoneco.2016.06.002>.
- Stock, J.H., and Watson, M.W. (2011). Dynamic factor models. In M. J. Clements, and D. F. Hendry (Eds.), *Oxford Handbook on Economic Forecasting*, chapter 2, pages 35–39. Oxford: Oxford University Press. URL <https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780195398649.001.0001/oxfordhb-9780195398649-e-3>.
- Stock, J.H., and Watson, M.W. (2016). Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In J. B. Taylor, and H. Uhlig (Eds.), *Handbook of Macroeconomics*, volume 2, chapter 8, pages 415–525. Netherlands, Amsterdam: Elsevier. URL <https://doi.org/10.1016/bs.hesmac.2016.04.002>.