The sparse dynamic factor model: A regularised quasi-maximum likelihood approach

Luke Mosley[1], Tak-Shing T. Chan[1] and Alex Gibberd[1*]

[1*]Department of Mathematics and Statistics, Lancaster University, Lancaster, LA1 4YW, United Kingdom.

*Corresponding author(s). E-mail(s): a.gibberd@lancaster.ac.uk;
Contributing authors: l.mosley@lancaster.ac.uk; t.t.chan@lancaster.ac.uk;

Abstract

The concepts of sparsity, and regularised estimation, have proven useful in many high-dimensional statistical applications. Dynamic factor models (DFMs) provide a parsimonious approach to modelling high-dimensional time series, however, it is often hard to interpret the meaning of the latent factors. This paper formally introduces a class of sparse DFMs whereby the loading matrices are constrained to have few non-zero entries, thus increasing interpretability of factors. We present a regularised M-estimator for the model parameters, and construct an efficient expectation maximisation algorithm to enable estimation. Synthetic experiments demonstrate consistency in terms of estimating the loading structure, and superior predictive performance where a low-rank factor structure may be appropriate. The utility of the method is further illustrated in an application forecasting electricity consumption across a large set of smart meters.

Keywords: Sparsity, Dynamic factor model, Time series, High-dimensional, Energy

1 Introduction

Originally formalised by Geweke (1977), the premise of the dynamic factor model (DFM) is to assume that the common dynamics of a large number of stationary zero-mean time series $\boldsymbol{X}_t = (X_{1,t}, \ldots, X_{p,t})^\top$ stem from a relatively small number of unobserved (latent) factors $\boldsymbol{F}_t = (F_{1,t}, \ldots, F_{r,t})^\top$ where $r \ll p$ through the linear system

$$\boldsymbol{X}_t = \boldsymbol{\Lambda}\boldsymbol{F}_t + \boldsymbol{\epsilon}_t\,, \tag{1}$$

for observations $t = 1, \ldots, n$. The matrix $\boldsymbol{\Lambda}$ provides a direct link between each factor in $\boldsymbol{F}_t$ and each variable in $\boldsymbol{X}_t$. The larger the loading $|\Lambda_{i,j}|$ for variable i and factor j, the more correlated this variable is with the factor. The common component $\boldsymbol{\chi}_t = \boldsymbol{\Lambda}\boldsymbol{F}_t$ captures the variability in the time series variables that is due to the common factors, while the idiosyncratic errors $\boldsymbol{\epsilon}_t = (\epsilon_{1,t}, \ldots, \epsilon_{p,t})^\top$ capture the features that are specific to individual series, such as measurement error. What makes the factor model in (1) a *dynamic* factor model is the assumption that the factors, and possibly the idiosyncratic errors may be temporally dependent, i.e., are time series themselves.

Arguably, it was the application of Sargent et al (1977) showing how just two dynamic factors were able to explain the majority of variance in headline US macroeconomic variables that initiated the DFMs popularity. The DFM is nowadays ubiquitous within the economic statistics community, with applications in nowcasting/forecasting (Giannone et al, 2008; Banbura et al,

2010; Foroni and Marcellino, 2014), constructing economic indicators (Mariano and Murasawa, 2010; Grassi et al, 2015), and counterfactual analysis (Harvey, 1996; Luciani, 2015). Examples in other domains include psychology (Molenaar, 1985; Fisher, 2015), the energy sector (Wu et al, 2013; Lee and Baldick, 2016) and many more, see Stock and Watson (2011) and Poncela et al (2021) for detailed surveys of the literature.

The DFM can be used in both an exploratory (inferential) setting, as well as a predictive (forecasting) mode. When dealing with the former its use is analogous to how one might apply principal component analysis (PCA) to understand the directions of maximum variation in a dataset, of course, the DFM does not just describe the cross-correlation structure, like PCA, but also the autocovariance. The loadings matrix $\boldsymbol{\Lambda}$ is usually used to assess how one should interpret a given (estimated) factor. Unfortunately, as in PCA, the interpretation of the factors in a traditional DFM is blurred as all variables are loaded onto all factors.

Our contributions

This paper seeks to bring modern tools from sparse modelling and regularised estimation to bear on the DFM. Specifically, we formalise a class of sparse factor models whereby only a subset of the factors will be active for a given variable—we assume the matrix $\boldsymbol{\Lambda}$ is sparse. Unlike regular sparse principal component analysis (SPCA) approaches, we take a penalised likelihood estimation approach, and noting that the likelihood is incomplete, we suggest a novel expectation-maximisation (EM) algorithm to perform estimation. The algorithms developed are computationally efficient, and give users a new method for imposing weakly informative (sparse) structural priors on the factor model. The data-driven estimation of the loadings support contrasts with the hard constraints that are more traditional in the use of DFMs.

The analysis within this paper is empirical in nature, we consider three aspects: i) how our EM algorithm performs in recovering the true sparsity pattern in the factor loadings; ii) how the model contrasts with alternative models in a predictive setting, e.g. where we want to forecast either all the p time series, or just a subset of these; and iii) how the model and estimation routine can be used in practice to extract insights from complex real-world datasets. The first two points are illustrated through extensive synthetic experiments, whilst for the latter, we give an example application to a set of smart meter data from across our university campus. To our knowledge this is the first time a DFM has been used to study building level energy data, and illustrates some of the benefits that come from imposing sparsity in terms of increasing the interpretability of the model.

2 Background and related work

Canonically, the dynamics of the latent factors in the DFM are specified as a stationary process. Here we focus on the popular VAR(1) model:

$$\boldsymbol{F}_t = \boldsymbol{A}\boldsymbol{F}_{t-1} + \boldsymbol{u}_t\,, \tag{2}$$

where $\boldsymbol{u}_t$ is a zero-mean series of disturbances with covariance matrix $\boldsymbol{\Sigma}_u$. Furthermore, the idiosyncratic errors $\boldsymbol{\epsilon}_t$ in (1) are commonly assumed to be zero-mean and cross-sectionally uncorrelated, meaning their covariance matrix, which we denote $\boldsymbol{\Sigma}_\epsilon$, is diagonal. Models with these assumptions are termed *exact*. Even if we relax the assumptions to allow for cross-correlated idiosyncratic errors (called an *approximate* DFM), consistent estimation of the factors is still possible as $(n, p) \to \infty$ (Doz et al, 2011). Therefore, the 'curse of dimensionality', often a burden for analysing time series models, can actually be beneficial in DFMs.

Estimation

The measurement equation (1) along with the state equation (2) form a state space model. A simple approach to estimate factor loadings is to consider the first r eigenvectors of the sample covariance matrix of $\boldsymbol{X}$, essentially applying PCA to the time series. This has been extensively reviewed in the literature (Stock and Watson, 2002; Bai, 2003; Doz and Fuleky, 2020). When mild conditions are placed on the correlation structure of idiosyncratic errors, the PCA estimator is the optimal non-parametric[1] estimator

[1] In the sense that temporal dependence is not restricted to that encoded via a parametric model.

for a large approximate DFM. With even tighter conditions of spherical idiosyncratic components, i.e. they are i.i.d. Gaussian, then the PCA estimator is equivalent to the maximum likelihood estimator (Doz and Fuleky, 2020). The problem with using non-parametric PCA methods to estimate the loading structure is that there is no consideration of the dynamics of the factors or idiosyncratic components. In particular, there is no feedback from the estimation of the state equation (2) to the measurement equation (1). For this reason, it is preferable to use parametric methods that are able to account for temporal dependencies in the system.

An alternative approach is proposed in Giannone et al (2008) whereby the initial estimates of the factors and loadings are derived from PCA, the VAR(1) parameters are estimated from these preliminary factors, before updating the factor estimates using Kalman smoothing. This two-stage approach has been theoretically analysed in Doz et al (2011) and successfully applied to the field of nowcasting in many national statistical institutes and central banks. The Kalman smoothing step in particular is very helpful for handling missing data, whether it be backcasting missing at the start of the sample, forecasting missing data at the end of the sample[2] or interpolating arbitrary patterns of missing data throughout the sample.

Bańbura and Modugno (2014) build on the DFM representation of Watson and Engle (1983) and adopt an EM algorithm to estimate the system (1)-(2) with a quasi maximum likelihood estimation (QMLE) approach. Doz et al (2012), Bai and Li (2016), and Barigozzi and Luciani (2022) provide theoretical results whereby, as $(n, p) \to \infty$, the QMLE estimates (based on an exact Gaussian DFM) are consistent under milder assumptions allowing for correlated idiosyncratic errors. The EM approach to estimation is beneficial as it allows feedback between the estimation of the factors and the loadings, and thus handle arbitrary patterns of missing data.

[2]Missing data at the end of the sample, commonly referred to as the 'ragged edge' problem, is very common in macroeconomic nowcasting applications. It is caused by time series used in the model having differing publication delays, and hence forming a ragged edge of missingness at the end of sample.

Relation to current work

In the literature, the idea of a sparse DFM is not new. A classic approach is to use factor rotations that aim to minimise the complexity in the factor loadings to make the structure simpler to interpret. See Kaiser (1958) for the well-established varimax rotation and see Carroll (1953) and Jennrich and Sampson (1966) for the well-established quartimin rotation. Rotations utilising ℓ_1 and ℓ_p norms are considered in Freyaldenhoven (2023); Liu et al (2023). For a recent discussion paper on the varimax rotation see Rohe and Zeng (2020). An alternative approach based on LASSO regularisation is to use SPCA (Zou et al, 2006) in place of regular PCA on the sample covariance matrix in the preliminary estimation of factors and loadings, i.e. in stage one of the two-stage approach by Giannone et al (2008). For factor modelling, it has been used by Croux and Exterkate (2011) in a typical macroeconomic forecasting setting where they consider a robustified version. Kristensen (2017) use SPCA to estimate diffusion indexes with sparse loadings. Despois and Doz (2022) prove that SPCA consistently estimates the factors in an approximate factor model if the ℓ_1 penalty is of $\mathcal{O}(p^{-1/2})$. They also compare SPCA with factor rotation methods and show an improved performance when the true loadings structure is sparse. Recently, Uematsu and Yamagata (2022) demonstrated consistency of a related SPCA approach in the weak-factor model setting, linking the sparsity of the loadings with the growth rate (in p) of the eigenvalues of the covariance. Finally, the work of Bai and Ng (2008) looks at selecting predictors within a factor model framework using methods such as lasso, and hard-thresholding.

Unlike previous research, our methodology implements regularisation within an EM algorithm framework, allowing us to robustly handle arbitrary patterns of missing data, model temporal dependence in the processes, and impose weakly informative (sparse) prior knowledge on the factor loadings. We argue that in settings where autocorrelation is moderately persistent, that the feedback provided through our EM procedure is important in aiding recovery of the factor loadings, as well as producing accurate forecasts.

The rest of the paper is structured as follows. In Section 3 we formalise our DFM model and

the sparsity assumptions placed on the loading matrices. Sect. 4 presents a regularised likelihood estimator for the model parameters, and introduces an EM algorithm to enable finding feasible estimates, and discusses how we implement the method using the R package `sparseDFM` (Mosley et al, 2023). Numerical results, including simulation studies and real data analysis, are presented in Sects. 5 and 6, respectively. The paper concludes with a discussion of the results, and how the models and estimators can be further generalised to provide flexibility to users.

3 The sparse DFM

Consider the p-variate time series $\{\boldsymbol{X}_t\}$ and r factors $\{\boldsymbol{F}_t\}$ related according to the model

$$\begin{aligned} \boldsymbol{X}_t &= \boldsymbol{\Lambda}_0 \boldsymbol{F}_t + \boldsymbol{\epsilon}_t \\ \boldsymbol{F}_t &= \boldsymbol{A}\boldsymbol{F}_{t-1} + \boldsymbol{u}_t \ , \end{aligned} \tag{3}$$

where $\{\boldsymbol{\epsilon}_t\}$ and $\{\boldsymbol{u}_t\}$ are multivariate white noise processes. For simplicity we assume $E[\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t^\top] = \Sigma_\epsilon = \text{diag}(\boldsymbol{\sigma}_\epsilon^2)$ and $\boldsymbol{\sigma}_\epsilon^2 \in \mathbb{R}_+^p$ is a vector of idiosyncratic variances. Similarly, let $E[\boldsymbol{u}_t\boldsymbol{u}_t^\top] = \Sigma_u$ and assume the eigenvalues of the VAR matrix are bounded $\|\boldsymbol{A}\| < 1$, thus the latent process is assumed stationary. This model corresponds to an exact DFM, where all the temporal dependence is modelled via the latent factors.

In this context, our notion of sparsity relates to the assumption that many of the entries in $\boldsymbol{\Lambda}_0$ will be zero. For instance, let the support of the kth column of the loading matrix be denoted

$$\mathcal{S}_k := \text{supp}(\Lambda_{0;\,\cdot,k}) \subseteq \{1,\dots,p\} \ ,$$

such that $s_k = |\mathcal{S}_k|$. We refer to a DFM as being sparse if $s_k < p$ for some or all of the $k = 1,\dots,r$ factors. In practice, this is an assumption that many of the observed series are driven by only a few (r) latent factors, and that for many series only a subset of the factors will be relevant.

3.1 Consistency and pervasiveness

In the sparse situation, whereby $s_k < p$, we will be able to model only a subset of the observations with each factor. To enable us to model all p variables and gain information relating to the r factors as n, p increase we assume a couple of conditions on the specification. First, that the support of the observations, and the union of factor supports is equal, i.e. $\cup_{k=1}^r \mathcal{S}_k = \{1,\dots,p\}$, thus all observations are related to at least one of the factors. Second, that the support for each factor grows with the number of observed variables, in that $\{s_k\}$ is a non-decreasing sequence in p for each of the k factors. Assumptions of this form would allow us, in principle, to assess the consistency of factor estimation as p grows.

This asymptotic analysis in p (and n) contrasts with the traditional setting with a fixed p—for which the factors cannot be consistently recovered and can only be approximated, with error that depends on the signal-to-noise ratio $\|\boldsymbol{\Lambda}_0\Sigma_F\boldsymbol{\Lambda}_0^\top]\|/\|\Sigma_\epsilon\|$, where $\Sigma_F = E[\boldsymbol{F}_t\boldsymbol{F}_t^\top]$ (Bai and Li, 2016). Intuitively, this is due to the fact that if p is fixed, then we cannot learn anything more about the factor at a specific time t, as we do not get more information on the factors as n increases, instead we just get more samples (at different time points) relating to the series $\{\boldsymbol{F}_t\}$. When we go to the doubly asymptotic, or just $p \to \infty$ setting, then if the number of factors r is fixed or restricted to slowly grow in n then we can not only recover structures relating to $\{\boldsymbol{F}_t\}$, e.g. the specification of $\boldsymbol{A}$, but we can also get more information relating to the factor at the specific time t (Bai and Li, 2016; Barigozzi and Luciani, 2022). One way to ensure this growing information about the factors is to assume that they are in some sense pervasive—the more variables p we sample, the more this tells us about the r factors. We note, that for a more formal analysis of the DFM, a usual pervasiveness assumption placed on the loading conditions is given by Doz et al (2011), whereby $\lim_{p\to\infty} p^{-1}\lambda_{\min}(\boldsymbol{\Lambda}_0^\top\boldsymbol{\Lambda}_0) > 0$, i.e. the average loading onto the least-influential factor is bounded away from zero.

In this paper, we choose to focus on the empirical performance of our estimator, thus we do not formalise the sparsity assumptions further. However, it is worth noting our empirical studies meet the pervasiveness assumptions regarding the support of the factor loadings.

3.2 Identifiability

In the following section, we will consider a QMLE estimator for the factor model based on assuming Gaussian errors $\boldsymbol{\epsilon}_t$ and $\boldsymbol{u}_t$, it is thus of interest to

consider how the associated likelihood relates to the factors and their loadings. Adopting a Gaussian error structure and taking expectations over the factors, the likelihood for (3) is given by

$$\mathcal{L}(\boldsymbol{\Lambda}) \propto \log\det(\boldsymbol{\Lambda}^\top \boldsymbol{\Sigma}_F \boldsymbol{\Lambda} + \boldsymbol{\Sigma}_\epsilon) - \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{\Lambda}^\top \boldsymbol{\Sigma}_F \boldsymbol{\Lambda} + \boldsymbol{\Sigma}_\epsilon)^{-1} \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{X}_t \boldsymbol{X}_t^\top\right] .$$

An obvious identifiability issue arises here, such that if $\tilde{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} Q$, $\tilde{\boldsymbol{F}}_t = Q\boldsymbol{F}_t$, for any unitary matrix $Q^\top = Q^{-1}$, we have $\mathcal{L}(\tilde{\boldsymbol{\Lambda}}) = \mathcal{L}(\boldsymbol{\Lambda})$. Now consider the case of $\tilde{\boldsymbol{\Lambda}}_0$, i.e. performing a rotation on the true loadings, denote the set of all possible equivalent loading as

$$\mathcal{E} := \{\boldsymbol{\Lambda}_0 Q \mid Q^\top = Q^{-1} \ , \ Q \in \mathbb{R}^{r\times r}\} \ . \tag{4}$$

The invariance of the likelihood to elements of this set mandates that theoretical analysis of the DFM is typically constructed in a specific frame of reference, cf. Doz et al (2011, 2012); Bai and Li (2016); Barigozzi and Luciani (2022).

Interestingly, our sparsity assumptions restrict the nature of this equivalence class considerably, in that only loading matrices with sparse structure are permitted. In general, there will still be multiple sparse representations that are allowed, and the issue of the scale invariance remains, however, the latter can be fixed by imposing a further constraint on the norms of the loading matrices. In this work, we demonstrate empirically that it is possible to construct estimators that are consistent up to rotations that maintain an optimal level of sparsity, in the sense that the true loading matrix is given by

$$\boldsymbol{\Lambda}_0 \in \arg\min_{\boldsymbol{\Lambda}\in\mathcal{E}} \sum_{k=1}^{r} \|\boldsymbol{\Lambda}_{\cdot,k}\|_0 \, , \tag{5}$$

where $\|\boldsymbol{\Lambda}_{\cdot,k}\|_0 := |\mathrm{supp}(\boldsymbol{\Lambda}_{\cdot,k})|$ counts the number of non-zero loadings. More generally (see Fig. 1) we could consider selecting on the basis of the ℓ_q norm, $\|\boldsymbol{\Lambda}\|_q := (\sum_{ik} \Lambda_{i,k}^q)^{1/q}$, the ℓ_1 norm may still provide selection, however, the ℓ_2 norm provides no selection as it maintains the rotational invariance of the likelihood. In this paper, we restrict our equivalence set on the basis of the ℓ_0 norm, as above, that is, we specify the true loading matrices as those that maintain the highest number of zero values after consideration for all unitary linear transformations.

To illustrate how the sparsity constraint (5) breaks the more general invariance that regular DFMs suffer, we can consider the quantity $\|\boldsymbol{\Lambda}_0^* Q_{\mathrm{rot}}(\theta)\|_q$, where $Q_{\mathrm{rot}}(\theta) \in \mathbb{R}^{2\times 2}$ is a rotation matrix with argument $\theta \in (-\pi, \pi)$, and $\boldsymbol{\Lambda}_0^* \in \mathbb{R}^{10\times 2}$ has the first column half filled with ones, and the rest zero, the second column is set to be one minus the first. As we see from Fig. 1, without the additional restriction on our specification of $\boldsymbol{\Lambda}_0$, via Eq. 5, we would not be able to determine a preference for any particular element from the set $\mathcal{E} := \{\boldsymbol{\Lambda}_0^* Q_{\mathrm{rot}}(\theta) : \mid \theta \in \{-\pi, \pi\}\}$.

Empirically, these identifiability issues mean we are unable to recover the desired sign of the factor loadings in our experiments, whilst columns in the loading matrix may also be permuted, e.g. factor k can be swapped (under permutation of the columns in the loading matrix) with factor l, for any $k, l \in \{1, \ldots, r\}$. These are the same identifiability issues which we face in PCA, whereby the eigenvectors can be exchanged in terms of order and direction.

On review, we were made aware that the idea of using the ℓ_1 penalty to select from a range of rotations was previously been proposed in Freyaldenhoven (2023) where one can see a plot similar to Fig 1, the use of more general ℓ_p rotations of the loadings was considered in Liu et al (2023). Those authors suggest to rotate a preliminary estimate of the loadings, based on say PCA, however, we propose to combine the choice of which elements are sparse in conjunction with the estimation. As such, unlike the two-stage estimate and rotate approaches e.g. Kaiser (1958), we are able to achieve exact sparsity (loadings are set exactly to zero) in our estimates.

It is important to note, that whilst we can obtain sparse estimates, and these are restricted relative to the regular (dense) DFM, pitfalls in terms of identification are still present. In particular, if the columns of the true loading matrix are not orthogonal, then the initialisation of our procedure (based on PCA) may lead the estimator to fall into a local maxima that will not reflect the non-orthogonal nature of the true loadings. To ensure identifiability in a more general setting we require at least r^2 restrictions on the model. For example Bai and Ng (2013) consider a variety of identifiability assumptions for

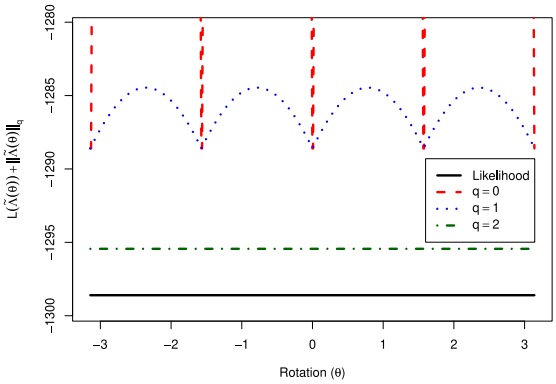

Fig. 1: The impact of rotation on the function $\mathcal{L}(\tilde{\mathbf{\Lambda}}(\theta)) + \|\tilde{\mathbf{\Lambda}}(\theta)\|_q$, in the case of $q = 0, 1$ the set of feasible $\mathbf{\Lambda}_0$ from (5) is restricted to the points $\theta \in \{0, \pm\frac{1}{2}\pi, \pm\pi\}$ corresponding to either swapping columns, or flipping signs.

PC estimators, and Barigozzi and Luciani (2022) for a discussion on identifiability with QMLE. Whilst our constraint on the ℓ_0 norm places constraints on the loadings, the specific position of the non-zero elements remains flexible and the relation to identifiability constraints is complex. In terms of specification of the true loadings, Freyaldenhoven (2023, Section 4.) provides discussion of a variety of sparsity patterns which would enable identification (in population). However, we leave more detailed theoretical analysis of this aspect to future work.

4 Estimation

Under the Gaussian error assumption, and collecting all parameters of the DFM (3) in $\boldsymbol{\theta} = (\mathbf{\Lambda}, \boldsymbol{A}, \mathbf{\Sigma}_{\boldsymbol{\epsilon}}, \mathbf{\Sigma}_{\boldsymbol{u}})$, we are able to write the joint log-likelihood of the data $\boldsymbol{X}_t$ and the factors $\boldsymbol{F}_t$ as:

$$\begin{aligned}
&\log \mathcal{L}(\boldsymbol{X}, \boldsymbol{F};\ \boldsymbol{\theta}) \qquad (6)\\
&= -\frac{1}{2}\log|\mathcal{P}_0| - \frac{1}{2}(\boldsymbol{F}_0 - \boldsymbol{\alpha}_0)^\top \mathcal{P}_0^{-1}(\boldsymbol{F}_0 - \boldsymbol{\alpha}_0)\\
&- \frac{n}{2}\log|\mathbf{\Sigma}_{\boldsymbol{u}}| - \frac{1}{2}\sum_{t=1}^{n} \boldsymbol{u}_t^\top \mathbf{\Sigma}_{\boldsymbol{u}}^{-1}\boldsymbol{u}_t\\
&- \frac{n}{2}\log|\mathbf{\Sigma}_{\boldsymbol{\epsilon}}| - \frac{1}{2}\sum_{t=1}^{n} \boldsymbol{\epsilon}_t^\top \mathbf{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\boldsymbol{\epsilon}_t
\end{aligned}$$

where $\boldsymbol{\epsilon}_t = \boldsymbol{X}_t - \mathbf{\Lambda}_0\boldsymbol{F}_t$, $\boldsymbol{u}_t = \boldsymbol{F}_t - \boldsymbol{A}\boldsymbol{F}_{t-1}$, and we have assumed an initial distribution at $t = 0$ of the factors as $\boldsymbol{F}_0 \sim N(\boldsymbol{\alpha}_0, \mathcal{P}_0)$.

We propose to induce sparsity in our estimates using the familiar ℓ_1 penalty, with motivation similar to that of the LASSO (Tibshirani, 1996). Alternative penalty functions are available, however, the convexity of the ℓ_1 penalty is appealing. Even though the overall objective for the parameters is non-convex, due to the rotational invariance of the log-likelihood, the convexity of the penalty ensures we can quickly and reliably apply the sparsity constraints. We will make use of this structure in the algorithms we construct to find estimates in practice. It is worth noting that our focus here is on the factor loadings, and thus this is the object we regularise, possible extensions could consider additional/alternative constraints, for instance on the latent VAR matrix.

Our proposed estimator attempts to minimise a penalised negative log-likelihood, as follows

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\log \mathcal{L}(\boldsymbol{X}, \boldsymbol{F};\ \boldsymbol{\theta}) + \alpha R(\mathbf{\Lambda})\ , \qquad (7)$$

where $\alpha \geq 0$. A larger α corresponds to a higher degree of shrinkage on the loadings, e.g. for a larger α we would expect more zero values in the loading matrices.

4.1 A regularised expectation maximisation algorithm

The regularised likelihood (7) is incomplete, as whilst we have observations, we do not observe the factors. To solve this problem, we propose to construct an EM framework where we take expectations over the factors (fixing the parameters), then conditional on the expected factors we maximise the log-likelihood with respect to the parameters $\boldsymbol{\theta}$, we iterate this process until our estimates converge.

The EM algorithm involves calculating and maximising the expected log-likelihood of the DFM conditional on the available information $\mathbf{\Omega}_n$. Given the log-likelihood in (6), the conditional expected log-likelihood is

$$\begin{aligned}
&\mathbb{E}\left[\log \mathcal{L}(\boldsymbol{X}, \boldsymbol{F};\ \boldsymbol{\theta})|\mathbf{\Omega}_n\right] = -\frac{1}{2}\log|\mathcal{P}_0| \qquad (8)\\
&- \operatorname{tr}\left\{\mathcal{P}_0^{-1}\mathbb{E}\left[(\boldsymbol{F}_0 - \boldsymbol{\alpha}_0)(\boldsymbol{F}_0 - \boldsymbol{\alpha}_0)^\top|\mathbf{\Omega}_n\right]\right\}
\end{aligned}$$

$$-\frac{n}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{u}}| - \frac{1}{2}\sum_{t=1}^{n}\mathrm{tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}\mathbb{E}\left[\boldsymbol{u}_t^\top\boldsymbol{u}_t|\boldsymbol{\Omega}_n\right]\right\}$$
$$-\frac{n}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}| - \frac{1}{2}\sum_{t=1}^{n}\mathrm{tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\mathbb{E}\left[\boldsymbol{\epsilon}_t^\top\boldsymbol{\epsilon}_t|\boldsymbol{\Omega}_n\right]\right\}\ .$$

Ultimately, we wish to impose our regularisation on the expected log-likelihood, our feasible estimator being given by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\left[-\mathbb{E}\left[\log\mathcal{L}(\boldsymbol{X},\boldsymbol{F};\ \boldsymbol{\theta})|\boldsymbol{\Omega}_n\right] + \alpha\|\boldsymbol{\Lambda}\|_1\right]\ . \tag{9}$$

4.1.1 The maximisation step

We use the following notation for the conditional mean and covariances of the state:

$$\begin{aligned}\boldsymbol{a}_{t|s} &= \mathbb{E}[\boldsymbol{F}_t|\boldsymbol{\Omega}_s]\,,\\ \boldsymbol{P}_{t|s} &= \mathrm{Cov}[\boldsymbol{F}_t|\boldsymbol{\Omega}_s]\,,\\ \boldsymbol{P}_{t,t-1|s} &= \mathrm{Cov}[\boldsymbol{F}_t,\boldsymbol{F}_{t-1}|\boldsymbol{\Omega}_s]\,.\end{aligned}$$

conditional on all information we have observed up to a time s, denoted by $\boldsymbol{\Omega}_s$.

As shown in Bańbura and Modugno (2014), the maximisation of (8) results in the following expressions for the parameter estimates:

$$\hat{\boldsymbol{\alpha}}_0 = \boldsymbol{a}_{0|n} \quad ; \quad \hat{\mathcal{P}}_0 = \boldsymbol{P}_{0|n} \tag{10}$$

and letting $\boldsymbol{S}_{t|n} = \boldsymbol{a}_{t|n}\boldsymbol{a}_{t|n}^\top + \boldsymbol{P}_{t|n}$, and $\boldsymbol{S}_{t,t-1|n} = \boldsymbol{a}_{t|n}\boldsymbol{a}_{t-1|n}^\top + \boldsymbol{P}_{t,t-1|n}$ we have

$$\hat{\boldsymbol{A}} = \left(\sum_{t=1}^{n}\boldsymbol{S}_{t-1|n}\right)^{-1}\left(\sum_{t=1}^{n}\boldsymbol{S}_{t,t-1|n}\right)\,, \tag{11}$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} = \frac{1}{n}\sum_{t=1}^{n}\left[\boldsymbol{S}_{t|n} - \hat{\boldsymbol{A}}\left(\boldsymbol{S}_{t-1,t|n}\right)\right]\,. \tag{12}$$

To minimise (9) for parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$, we should also consider there might be missing data in $\boldsymbol{X}_t$. Let us define a selection matrix $\boldsymbol{W}_t$ to be a diagonal matrix such that

$$W_{t,ii} = \begin{cases}1 & \text{if } X_{i,t} \text{ observed}\\ 0 & \text{if } X_{i,t} \text{ missing}\end{cases}$$

and note that $\boldsymbol{X}_t = \boldsymbol{W}_t\boldsymbol{X}_t + (\boldsymbol{I}-\boldsymbol{W}_t)\boldsymbol{X}_t$. The update for the idiosyncratic error covariance is then given by

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}} = \frac{1}{n}\sum_{t=1}^{n}\mathrm{diag}\Bigg[\boldsymbol{W}_t\Big(\boldsymbol{X}_t\boldsymbol{X}_t^\top - 2\boldsymbol{X}_t\boldsymbol{a}_{t|n}^\top\hat{\boldsymbol{\Lambda}}^\top + \hat{\boldsymbol{\Lambda}}\boldsymbol{S}_{t|n}\hat{\boldsymbol{\Lambda}}^\top\Big) + (\boldsymbol{I}-\boldsymbol{W}_t)\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{*}(\boldsymbol{I}-\boldsymbol{W}_t)\Bigg]\,, \tag{13}$$

where $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{*}$ is obtained from the previous EM iteration. As noted in Algorithm 1, in practice we update $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}$ after estimating $\hat{\boldsymbol{\Lambda}}$, as the former is based on the difference between the observations and the estimated common component. The following section details precisely how we practically obtain sparse estimates for the factor loadings, the estimates can then be used in (13) and thus complete the M-step of the algorithm.

4.1.2 Incorporating sparsity

In this work, we propose to update $\hat{\boldsymbol{\Lambda}}$ by constructing an Alternative Directed Method of Moments (ADMM) algorithm (Boyd et al, 2011) to solve (9) with the parameters $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\Sigma}}_u, \hat{\boldsymbol{\alpha}}_0, \hat{\mathcal{P}}_0)$ fixed. The algorithm proceeds by sequentially minimising the augmented Lagrangian

$$\begin{aligned}\mathcal{C}(\boldsymbol{\Lambda},\boldsymbol{Z},\boldsymbol{U}) := &-\mathbb{E}\left[\log\mathcal{L}(\boldsymbol{X},\boldsymbol{F};\ \boldsymbol{\theta})|\boldsymbol{\Omega}_n\right]\\ &+\alpha\|\boldsymbol{Z}\|_1 + \frac{\nu}{2}\|\boldsymbol{\Lambda}-\boldsymbol{Z}+\boldsymbol{U}\|_F^2\,,\end{aligned} \tag{14}$$

where $\boldsymbol{Z}\in\mathbb{R}^{p\times r}$ is an auxiliary variable, $\boldsymbol{U}\in\mathbb{R}^{p\times r}$ are the (scaled) Lagrange multipliers and ν is the scaling term. Under equality conditions relating the auxilary $(\boldsymbol{Z})$ to the primal $(\boldsymbol{\Lambda})$ variables, this is equivalent to minimising (9), e.g.

$$\begin{aligned}&\arg\min_{\boldsymbol{Z}=\boldsymbol{\Lambda}}\max_{\boldsymbol{U}}\mathcal{C}(\boldsymbol{\Lambda},\boldsymbol{Z},\boldsymbol{U})\\ &=\arg\min_{\boldsymbol{\Lambda}}\left[-\mathbb{E}\left[\log\mathcal{L}(\boldsymbol{X},\boldsymbol{F};\ \boldsymbol{\theta})|\boldsymbol{\Omega}_n\right]+\alpha\|\boldsymbol{\Lambda}\|_1\right]\end{aligned}$$

as (9) is convex in the argument $\boldsymbol{\Lambda}$ with all other parameters fixed, this argument holds for any $\nu > 0$ (Boyd et al, 2011; Lin et al, 2015).

The augmented Lagrangian (14) can be sequentially minimised via the following updates[3]

$$\begin{aligned}\boldsymbol{\Lambda}^{(k+1)} &= \arg\min_{\boldsymbol{\Lambda}} \mathcal{C}(\boldsymbol{\Lambda}, \boldsymbol{Z}^{(k)}, \boldsymbol{U}^{(k)}) \\ \boldsymbol{Z}^{(k+1)} &= \arg\min_{\boldsymbol{Z}} \mathcal{C}(\boldsymbol{\Lambda}^{(k+1)}, \boldsymbol{Z}, \boldsymbol{U}^{(k)}) \\ &= \text{soft}(\boldsymbol{\Lambda}^{(k+1)} + \boldsymbol{U}^{(k)};\ \alpha/\nu) \\ \boldsymbol{U}^{(k+1)} &= \boldsymbol{U}^{(k)} + \boldsymbol{\Lambda}^{(k+1)} - \boldsymbol{Z}^{(k+1)}\,.\end{aligned}$$

for $k = 0, 1, 2, \ldots,$ until convergence. The first (primal) update is simply a least-squares type problem, whereby on vectorising $\boldsymbol{\Lambda}$ one finds

$$\begin{aligned}\text{vec}(\boldsymbol{\Lambda}^{(k+1)}) = &\left(\sum_{t=1}^{n} \boldsymbol{S}_{t|n} \otimes \boldsymbol{W}_t \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{W}_t + \nu \boldsymbol{I}_{pr}\right)^{-1} \\ &\text{vec}\Bigg[\sum_{t=1}^{n} \boldsymbol{W}_t \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{W}_t \boldsymbol{X}_t \boldsymbol{a}_{t|n}^{\top} \\ &+ \nu(\boldsymbol{Z}^{(k)} - \boldsymbol{U}^{(k)})\Bigg]\,. \qquad (15)\end{aligned}$$

Remark 1 (Exploiting dimensionality reduction) For the $\boldsymbol{\Lambda}^{(k+1)}$ update, the dimensionality of the problem is quite large, leading to a naïve per-iteration cost of order $\mathcal{O}(r^3p^3)$. A more efficient method for this step can be sought by looking at the specific structure of the matrix to be inverted. Define $\mathcal{A}_t = \boldsymbol{S}_{t|n}$, $\mathcal{B}_t = \boldsymbol{W}_t \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{W}_t$, and $\mathcal{C} = \sum_{t=1}^{n} \boldsymbol{W}_t \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} \boldsymbol{W}_t \boldsymbol{X}_t \boldsymbol{a}_{t|n}^{\top} + \nu(\boldsymbol{Z}^{(k)} - \boldsymbol{U}^{(k)})$, then the solution (15) can be written as

$$\begin{aligned}\text{vec}(\boldsymbol{\Lambda}) &= \left(\sum_{t=1}^{n} \mathcal{A}_t \otimes \mathcal{B}_t + \nu \boldsymbol{I}_{pr}\right)^{-1} \text{vec}(\mathcal{C}) \\ &= \mathcal{D}^{-1} \text{vec}(C)\,.\end{aligned}$$

Since $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ is diagonal in an exact DFM, $\mathcal{B}_t$ is also diagonal and thus $\mathcal{D}$ is made up of r^2 blocks such that each $(i,j)^{th}$ block is a diagonal matrix of length p for $i, j = 1, \ldots, r$. To speed up the computation, we note that $\nu \boldsymbol{I}_{pr} = \nu \boldsymbol{I}_r \otimes \boldsymbol{I}_p$ and use the properties of commutation matrices (Magnus and Neudecker, 2019, p. 54), denoted by $\boldsymbol{K}_{rp}$, to write

$$\begin{aligned}&\left(\sum_{t=1}^{n} \mathcal{A}_t \otimes \mathcal{B}_t + \nu \boldsymbol{I}_r \otimes \boldsymbol{I}_p\right)^{-1} \\ &= \left[\sum_{t=1}^{n} \boldsymbol{K}_{rp}(\mathcal{B}_t \otimes \mathcal{A}_t)\boldsymbol{K}_{pr} + \boldsymbol{K}_{rp}(\boldsymbol{I}_p \otimes \nu \boldsymbol{I}_r)\boldsymbol{K}_{pr}\right]^{-1} \\ &= \boldsymbol{K}_{rp}\left(\sum_{t=1}^{n}(\mathcal{B}_t \otimes \mathcal{A}_t) + (\boldsymbol{I}_p \otimes \nu \boldsymbol{I}_r)\right)^{-1} \boldsymbol{K}_{pr}\,. \qquad (16)\end{aligned}$$

The matrix needing to be inverted in the final line of equation (16) is now a block diagonal matrix. We can extract each of the $1, \ldots, p$ blocks separately and invert them one-by-one. The final result from (16) has the expected block structure with a diagonal matrix in each block but we can stack them into a cube to save storage. Overall, the operations can be completed with cost $\mathcal{O}(r^3 p)$. Given that this needs to be performed for every iteration of the EM algorithm, our commutation trick results in significant computational gains.

Whilst other optimisation routines could be used to estimate the sparse loadings, the ADMM approach is appealing as it allows us to split (9) into sub-problems that can easily be solved. If one wished to incorporate more specific/structured prior knowledge, this approach can easily be altered to impose these assumptions, for instance, future work could consider group-structured regularisation allowing for more informative prior knowledge on the factor loadings to be incorporated. Hard constraints, e.g. where we require a loading to be exactly zero can also be incorporated at the $\boldsymbol{Z}$ update stage by explicitly setting some entries to be zero.

4.1.3 The expectation step

So far, we have discussed how to update the parameters conditional on the quantities $\mathbb{E}[\boldsymbol{F}_t|\boldsymbol{\Omega}_n]$ $\text{Cov}[\boldsymbol{F}_t|\boldsymbol{\Omega}_n]$, and $\text{Cov}[\boldsymbol{F}_t, \boldsymbol{F}_{t-1}|\boldsymbol{\Omega}_n]$. In our application, under the Gaussian error assumption, these expectations can be easily calculated via the Kalman smoother. For completeness, we detail this step in the context of the DFM model, as well as discussing some methods to speed up the computation which make use of the exact DFM structure.

The classical multivariate Kalman smoother equations can be slow when p is large. However, since we assume $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ is diagonal, we can equivalently filter the observations $\boldsymbol{X}_t$ one element at a time, as opposed to updating all p of them together as in the classic approach (Durbin and Koopman, 2012). As matrix inversion

[3] For the full derivation of $\boldsymbol{Z}$ and $\boldsymbol{U}$ refer to Boyd et al (2011). For the full derivation of $\boldsymbol{\Lambda}$ refer to the software paper implementing this algorithm of Mosley et al (2023).

becomes scalar divisions, huge speedups are possible. This approach, sometimes referred to as the univariate treatment, sequentially updates across both time and variable index when filtering/smoothing. Let us define the individual elements $\boldsymbol{X}_t = (X_{t,1}, \ldots, X_{t,p})^\top$, $\boldsymbol{\Lambda} = (\boldsymbol{\Lambda}_1^\top, \ldots, \boldsymbol{\Lambda}_p^\top)^\top$, $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \text{diag}(\sigma_{\epsilon 1}^2, \ldots, \sigma_{\epsilon p}^2)$. Following Koopman and Durbin (2000) we expand the conditional expectations according to

$$\begin{aligned}
\boldsymbol{a}_{t,i} &= \mathbb{E}[\boldsymbol{F}_t | \boldsymbol{\Omega}_{t-1}, X_{t,1}, \ldots, X_{t,i-1}], \\
\boldsymbol{a}_{t,1} &= \mathbb{E}[\boldsymbol{F}_t | \boldsymbol{\Omega}_{t-1}], \\
\boldsymbol{P}_{t,i} &= \text{Var}[\boldsymbol{F}_t | \boldsymbol{\Omega}_{t-1}, X_{t,1}, \ldots, X_{t,i-1}], \\
\boldsymbol{P}_{t,1} &= \text{Var}[\boldsymbol{F}_t | \boldsymbol{\Omega}_{t-1}],
\end{aligned}$$

for $i = 1, \ldots, p$ and $t = 1, \ldots, n$. The univariate treatment now filters this series over indices i and t. This is equivalent in form to the multivariate updates of the classic (Shumway and Stoffer, 1982) approach, except that the t subscript now becomes a t, i subscript, and the $t|t$ subscript now becomes $t, i+1$.

$$\begin{aligned}
v_{t,i} &= X_{t,i} - \boldsymbol{\Lambda}_i \boldsymbol{a}_{t,i}, \\
C_{t,i} &= \boldsymbol{\Lambda}_i \boldsymbol{P}_{t,i} \boldsymbol{\Lambda}_i^\top + \sigma_{\epsilon,i}^2, \\
\boldsymbol{K}_{t,i} &= \boldsymbol{P}_{t,i} \boldsymbol{\Lambda}_i^\top C_{t,i}^{-1}, \\
\boldsymbol{a}_{t,i+1} &= \boldsymbol{a}_{t,i} + \boldsymbol{K}_{t,i} v_{t,i}, \\
\boldsymbol{P}_{t,i+1} &= \boldsymbol{P}_{t,i} - \boldsymbol{K}_{t,i} C_{t,i} \boldsymbol{K}_{t,i}^\top,
\end{aligned}$$

for $i = 1, \ldots, p$ and $t = 1, \ldots, n$. If $X_{t,i}$ is missing or $C_{t,i}$ is zero, we omit the term containing $\boldsymbol{K}_{t,i}$. The transition to $t+1$ is given by the following prediction equations:

$$\begin{aligned}
\boldsymbol{a}_{t+1,1} &= \boldsymbol{A} \boldsymbol{a}_{t,p+1}, \\
\boldsymbol{P}_{t+1,1} &= \boldsymbol{A} \boldsymbol{P}_{t,p+1} \boldsymbol{A}^\top + \boldsymbol{\Sigma}_{\boldsymbol{u}}.
\end{aligned}$$

These prediction equations are exactly the same as the multivariate ones (i.e., predictions are not treated sequentially but all at once). From our perspective, this univariate treatment may be more appropriately referred to as performing univariate updates plus multivariate predictions.

Unlike Shumway and Stoffer (1982), the measurement update comes before the transition; however, we can revert to doing the transition first if our initial state means and covariances start from $t = 0$ instead of $t = 1$. Likewise, univariate smoothing is defined by:

$$\begin{aligned}
\boldsymbol{L}_{t,i} &= \boldsymbol{I}_m - \boldsymbol{K}_{t,i} \boldsymbol{\Lambda}_i, \\
\boldsymbol{b}_{t,i-1} &= \boldsymbol{\Lambda}_i^\top C_{t,i}^{-1} v_{t,i} + \boldsymbol{L}_{t,i}^\top \boldsymbol{b}_{t,i}, \\
\boldsymbol{J}_{t,i-1} &= \boldsymbol{\Lambda}_i^\top C_{t,i}^{-1} \boldsymbol{\Lambda}_i + \boldsymbol{L}_{t,i}^\top \boldsymbol{J}_{t,i} \boldsymbol{L}_{t,i}, \\
\boldsymbol{b}_{t-1,p} &= \boldsymbol{A}^\top \boldsymbol{b}_{t,0}, \\
\boldsymbol{J}_{t-1,p} &= \boldsymbol{A}^\top \boldsymbol{J}_{t,0} \boldsymbol{A},
\end{aligned}$$

for $i = p, \ldots, 1$ and $t = n, \ldots, 1$, with $\boldsymbol{b}_{n,p}$ and $\boldsymbol{J}_{n,p}$ initialised to 0. Again, if $X_{t,i}$ is missing or $C_{t,i}$ is zero, drop the terms containing $\boldsymbol{K}_{t,i}$. Finally, the equations for $\boldsymbol{a}_{t|n}$ and $\boldsymbol{P}_{t|n}$ are:

$$\begin{aligned}
\boldsymbol{a}_{t|n} &= \boldsymbol{a}_{t,1} + \boldsymbol{P}_{t,1} \boldsymbol{b}_{t,0}, \\
\boldsymbol{P}_{t|n} &= \boldsymbol{P}_{t,1} - \boldsymbol{P}_{t,1} \boldsymbol{J}_{t,0} \boldsymbol{P}_{t,1}.
\end{aligned}$$

These results will be equivalent to $\boldsymbol{a}_{t|n}$ and $\boldsymbol{P}_{t|n}$ from the classic multivariate approach, yet obtained with substantial improvement in computational efficiency. In order to calculate the cross-covariance matrix $\boldsymbol{P}_{t,t-1|n}$, we use De Jong and Mackinnon (1988)'s theorem:

$$\boldsymbol{P}_{t,t-1|n} = \boldsymbol{P}_{t|n} (\boldsymbol{P}_{t|t-1})^{-1} \boldsymbol{A} \boldsymbol{P}_{t-1|t-1}. \tag{17}$$

4.2 Parameter tuning

There are two key parameters that need to be set for the DFM model. The first is to select the number of factors, and the second is to select an appropriate level of sparsity. One may argue that these quantities should be selected jointly, however, in the interests of computational feasibility, we here propose to use heuristics, first selecting the number of factors, and then deciding on the level of sparsity. This mirrors how practitioners would typically apply the DFM model, where there is often a prior for the number of relevant factors (or more usually an upper bound). Both the number of factors, and the structure of the factor loadings impact the practical interpretation of the estimated factors.

4.2.1 Choosing the number of factors

To calculate the number of factors to use in the model we opt to take the information criteria approach of Bai and Ng (2002). There are several criteria that are discussed in the literature, for example, the paper of Bai and Ng (2002) suggests

three forms[4]. For this paper, we use the criteria of the following form:

$$IC(r) = \log V_r(\bar{\boldsymbol{F}}, \bar{\boldsymbol{\Lambda}}) + r \left(\frac{n+p}{np} \right) \log \min(n, p) \,, \tag{18}$$

where

$$V_r(\bar{\boldsymbol{F}}, \bar{\boldsymbol{\Lambda}}) = \frac{1}{np} \sum_{i=1}^{p} \sum_{t=1}^{n} \mathbb{E}[\bar{\epsilon}_{i,t}^2]$$

and $\bar{\epsilon}_{i,t} = X_{t,i} - \bar{\boldsymbol{\Lambda}}_{i,\cdot} \bar{\boldsymbol{F}}_t$ is found using PCA when applied to the standardized data. The preliminary factors $\bar{\boldsymbol{F}}$ correspond to the principal components, and the estimated loadings $\bar{\boldsymbol{\Lambda}}$ corresponding to the eigenvectors. Should the data contain missing values, we first interpolate the missing values using the median of the series and then smooth these with a simple moving window.

Remark 2 We note that ideally one may wish to apply the EM procedure to get more refined estimates of both the factors and loadings, however, in the interests of computational cost and in-line with current practice we propose to use the quick (preliminary) estimates above, denoted with $\bar{\boldsymbol{\Lambda}}$ rather than $\hat{\boldsymbol{\Lambda}}$.

4.2.2 Tuning the regulariser

Once a number of factors r has been decided, we tune α by performing a simple search over a logarithmically spaced grid and minimise a Bayesian Information Criteria defined as

$$BIC(\alpha) = \log \left(V_\alpha(\hat{\boldsymbol{F}}, \hat{\boldsymbol{\Lambda}}) \right) + \frac{\log(np)}{np} \sum_{k=1}^{r} \hat{s}_k \,, \tag{19}$$

where $\hat{s}_k$ is the number of non-zero entries in the kth column of the estimated loading matrix. In this case, we run the EM algorithm until convergence (usually after a dozen or so iterations) and then evaluate the BIC using the resulting $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{\Lambda}}$, this procedure is repeated for each α in the grid. An example of the resulting curve can be seen in the empirical application of Section 6. To limit searching over non-optimal values, an upper limit for α is set whereby, if the loadings for a particular factor are all set to zero, then we terminate the search.

[4]In our experiments and applications, we compared all criteria and they typically give similar results within ± 1 of each other, for simplicity, only one IC is presented here.

Remark 3 Tuning both the number of factors, and the regulariser for these models is a topic of open research and discussion. Indeed, whilst the criteria of Bai and Ng (2002) are well used, there is still lively debate about what is an appropriate number of factors, and this usually determined by a mix of domain (prior) knowledge and heuristics such as those presented above. The heuristics provided here seem reasonable in the applications and experiments we consider, however, we do not claim they are optimal for all scenarios.

4.3 Implementation

We have implemented the estimation routine as part of the R package `sparseDFM` available via CRAN. The EM routine and ADMM updates are implemented in C++ using the Armadillo library. Initialisation of the ADMM iterates utilises a warm start procedure whereby the solution at the previous iteration of the EM algorithm initialises the next solution. Furthermore, warm-starts are utilised when searching over an α tuning grid. As noted in other applications (Hu et al, 2016) starting the ADMM procedure can lead to considerable speed-ups. With regards to the augmentation parameter ν in the ADMM algorithm, we simply keep this set to 1 for the experiments run here, however, it is possible that tuning this parameter could lead to further speedups.

On the first iteration of the algorithm, the EM procedure is initialised by a simple application of PCA to the standardised data, analagously to how the preliminary factors and loadings $\bar{\boldsymbol{\Lambda}}$ were found in Section 4.2. A summary of the EM algorithm as a whole is given in Algorithm 1.

5 Synthetic experiments

We provide a Monte-Carlo numerical study to show the performance of our QMLE estimator in terms of recovery of sparse loadings and the ability of the sparse DFM to forecast missing data at the end of the sample. In particular, we simulate from a ground-truth model according to:

$$\begin{aligned} \boldsymbol{X}_t &= \boldsymbol{\Lambda} \boldsymbol{F}_t + \boldsymbol{\epsilon}_t \,, & \boldsymbol{\epsilon}_t &\sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}) \,, \\ \boldsymbol{F}_t &= \boldsymbol{A} \boldsymbol{F}_{t-1} + \boldsymbol{u}_t \,, & \boldsymbol{u}_t &\sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}}) \,, \end{aligned}$$

for $t = 1, \ldots, n$ and $\boldsymbol{X}_t$ having p variables. We set the number of factors to be $r = 2$ and consider

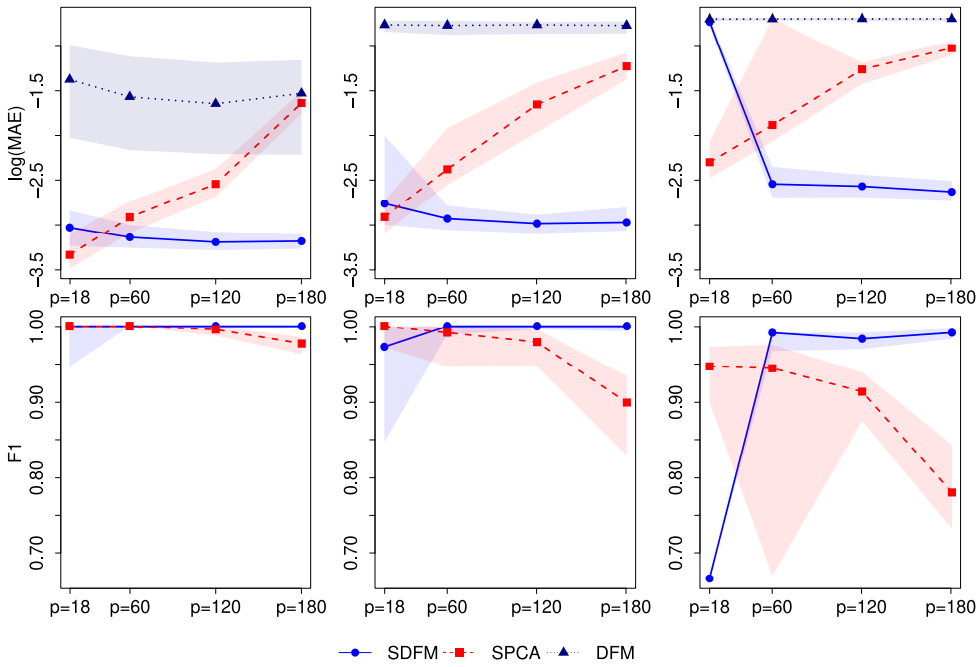

Fig. 2: Median log-MAE score (top panel) and median F1 score (bottom panel) for recovering factor loadings across 100 experiments with a shaded confidence band of the 25th and 75th percentile. The plots represent a setting with a fixed $n = 100$ and varying number of variables p and where the cross-correlation parameter in the VAR(1) process is set to $\rho = 0$ (left plot), $\rho = 0.6$ (middle plot) and $\rho = 0.9$ (right plot).

true model parameters of the form:

$$\begin{aligned}
\boldsymbol{\Lambda} &= \boldsymbol{I}_2 \otimes \mathbf{1}_{p/2} = \begin{bmatrix} \mathbf{1}_{p/2} & \mathbf{0}_{p/2} \\ \mathbf{0}_{p/2} & \mathbf{1}_{p/2} \end{bmatrix}, \\
\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} &= \boldsymbol{I}_p, \\
\boldsymbol{A} &= \begin{bmatrix} a & 0 \\ \rho & 0 \end{bmatrix}, \\
\boldsymbol{\Sigma}_u &= \begin{bmatrix} 1-a^2 & 0 \\ 0 & 1-\rho^2 \end{bmatrix}. \qquad (20)
\end{aligned}$$

The loadings matrix $\boldsymbol{\Lambda}$ is a block-diagonal matrix which is $1/2$ sparse with $p/2$ ones in each block. We set up the VAR(1) process of the factors in this way such that we can adjust the cross-correlation parameter ρ between the factors while having factors that always have variance one. This allows us to understand how important a cross-correlation at non-zero lags structure is when assessing model performance. We vary the ρ parameter between $\rho = \{0, 0.6, 0.9\}$, going from no cross-correlation to strong cross-correlation between the factors. We set the covariance of the idiosyncratic errors to be $\boldsymbol{I}_p$ in order to have a signal-to-noise ratio between the common component $\boldsymbol{\Lambda}\boldsymbol{F}_t$ and the errors $\boldsymbol{\epsilon}_t$ equal to one.

5.1 Recovery of sparse loadings

We apply our sparse DFM (SDFM) estimator to simulated data from the data generating process above to assess how well we can recover the true loadings matrix $\boldsymbol{\Lambda}$. We compare our method to sparse principal component analysis[5] (SPCA) applied to $\boldsymbol{X}_t$ to test which settings we are performing better in. We tune for the best ℓ_1-norm parameter in both SDFM and SPCA using the BIC function (19) by searching over a large grid of logspaced values from 10^{-3} to 10^2. We also make comparisons to the regular DFM approach of Bańbura and Modugno (2014) to test the importance of using regularisation when the true loading structure is sparse.

The estimation accuracy is assessed with mean absolute error (MAE) between the true loadings according to $(rp)^{-1}\|\hat{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}\|_1$. We also provide

[5]The SPCA algorithm is implemented using the *elasticnet* R package available on CRAN.

Algorithm 1 EM algorithm for SDFM

Input: $\boldsymbol{X}$, α
Output: $\boldsymbol{\Lambda}$, $\boldsymbol{A}$, $\boldsymbol{\Sigma}_\epsilon$, $\boldsymbol{\Sigma}_u$

Initialize $\boldsymbol{\theta} = (\boldsymbol{\Lambda}, \boldsymbol{A}, \boldsymbol{\Sigma_\epsilon}, \boldsymbol{\Sigma_u})$ via cubic spline fitting (for missing value imputation) followed by PCA and a VAR fit
repeat
  Obtain $\boldsymbol{a}_{t|n}$ and $\boldsymbol{P}_{t|n}$ via univariate Kalman filtering and smoothing ▷ E-step
  Calculate $\boldsymbol{P}_{t,t-1|n}$ via Eq. (17) ▷ M-step
  Update $\boldsymbol{A}$ and $\boldsymbol{\Sigma_u}$ via Eqs. (11) and (12)
  Initialize $\boldsymbol{\Lambda}^{(0)} = \boldsymbol{Z}^{(0)} = \boldsymbol{U}^{(0)} = 0$
  for $k = 0, ...,$ until convergence **do**
    $\boldsymbol{\Lambda}^{(k+1)} = \arg\min_{\boldsymbol{\Lambda}} \mathcal{C}(\boldsymbol{\Lambda}, \boldsymbol{Z}^{(k)}, \boldsymbol{U}^{(k)})$ via Eqs. (15) and (16)
    $\boldsymbol{Z}^{(k+1)} = \text{soft}(\boldsymbol{\Lambda}^{(k+1)} + \boldsymbol{U}^{(k)};\ \alpha/\nu)$
    $\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \boldsymbol{\Lambda}^{(k+1)} - \boldsymbol{Z}^{(k+1)}$
  end for
  Update $\boldsymbol{\Sigma_\epsilon}$ via Eq. (13)
until convergence

results for the F1 score for the sparsity inducing methods of SDFM and SPCA to measure how well the methods capture the true sparse structure. Due to invariance issues discussed, the estimated loadings may not be on the same scale as the true loadings, we thus first re-scale the estimated loadings such that their norm is equal to that of the simulated loadings, i.e. $\|\hat{\boldsymbol{\Lambda}}\|_2 = \|\boldsymbol{\Lambda}\|_2$. The estimated loadings from each model are identified up to column permutations and therefore we permute the columns of $\hat{\boldsymbol{\Lambda}}$ to match the true order of $\boldsymbol{\Lambda}$. We do this by measuring the 2-norm distance between the columns of $\hat{\boldsymbol{\Lambda}}$ and $\boldsymbol{\Lambda}$ and iteratively swapping to match the smallest distances.

Figure 2 displays the results for the loadings recovery where we have fixed the number of observations to be $n = 100$ and vary the number of variables between $p = \{18, 60, 120, 180\}$ along the x-axis and the cross-correlation parameter in the VAR(1) process between $\rho = \{0, 0.6, 0.9\}$ going from the left to middle to right plot respectively. The top panel shows the median MAE score (in logarithms) over 100 experiments while the bottom panel shows the F1 scores. We provide confidence bands for both representing the 25th and 75th percentiles. It is clear from the plots that the sparsity inducing methods of SDFM and SPCA are dominating a regular DFM when the true loadings structure is in fact sparse. It is also clear that SPCA performs poorly, compared with SDFM, when the cross-section of the data increases for a fixed n. This is even more noticeable from the F1 score when ρ increases. This highlights the importance of the SDFM's ability to capture correlations between factors at non-zero lags. Unlike SPCA, the EM algorithm of SDFM allows feedback from the estimated factors when updating model parameters, allowing it to capture these factor dependencies. We see improved scores in MAE as the cross-section increases for SDFM. This follows the intuition of the EM algorithm framework as we learn more about the factors as the dimension $p \to \infty$. We should remark that for most scenarios the F1 score for SDFM is almost one, however, when $p = 18$ and ρ is high, the score does drop. In this setting a low value for α minimises BIC, meaning almost no sparsity is applied (a very similar result to a regular DFM fit). Here, the two factors are highly correlated and there is not enough cross-section to determine factor structure. In practice it is likely that cross-section will be large and hence this result is not too concerning.

5.2 Bias-Variance Tradeoff

In SDFM there is a potential trade-off whereby sparse loading matrices lead the update for each factor $\boldsymbol{a}_{t|i}$ to rely on a subset of the data-points. The mechanism for this reliance can be seen in our univariate smoother, where the residual $v_{t,i}$ is calculated from a subset of the factors when $\boldsymbol{\Lambda}_i$ is sparse, which then leads into the updates for conditional expectations. If this sparsity isn't well calibrated (aligned to true $\boldsymbol{\Lambda}$) then we may expect to see an increase in the variance of the estimated factors for a very sparse $\hat{\boldsymbol{\Lambda}}$. To examine the impact of sparsity in more detail, we run a set of experiments where the solution path of SDFM is evaluated across a range of $\alpha \geq 0$. For each value, we calculate the (empirical) mean-square error, bias, and variance for both the loadings, and the estimated factors given by the Kalman smoother, i.e., $\hat{F}_t = \boldsymbol{a}_{t|n}$. We examine the behaviour of our estimator on both a sparse and non-sparse DFM model. Specifically, we assume a true loading given by

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{a}_{p/2} & -\boldsymbol{b}_{p/2} \\ \boldsymbol{b}_{p/2} & \boldsymbol{a}_{p/2} \end{bmatrix}$$

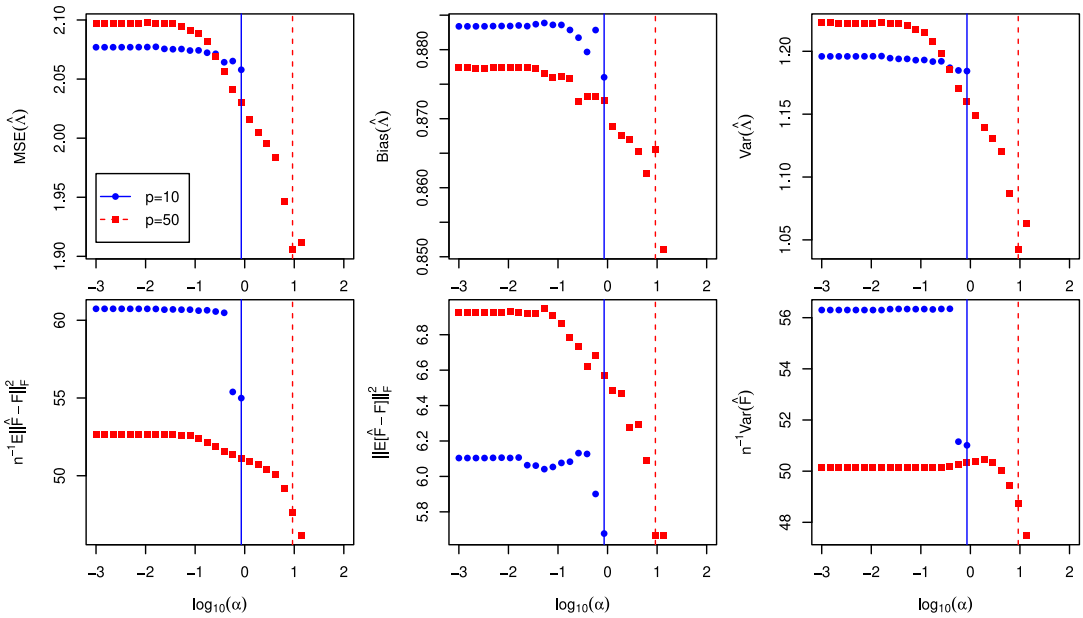

Fig. 3: Bias and variance for estimated loading matrices and factors (using SDFM) as a function of regularisation α in the case where the true $\mathbf{\Lambda}$ is sparse ($s = p$). Vertical lines indicate points in the regularisation path where the average estimated sparsity matches the true level of sparsity.

where $\boldsymbol{a}_{p/2}$ is the $p/2$ dimensional vector with each entry set to a. Assuming sparsity we set $a = 5, b = 0$, and thus half the elements are equal to zero, in the dense case, we set $b = -a$. The idiosyncratic errors are assumed to have unit variance, whilst the factors are assumed to be independent white noises (again with unit variance).

The results of this experiment, with expectation approximated by averaging over $n_{\text{sim}} = 500$ simulations are given in Figures 3, 4. To make comparisons more meaningful when comparing across dimensionalities, the estimates (and true) loadings have been scaled such that $\|\Lambda_{\cdot,k}\|_2 = 1$ for each $k = 1, 2$. It should also be noted that the results (in the figures) do not extend beyond $\alpha_{\max}$, which is defined as the smallest α over the n_{sim} experiments such that one estimated column of the loading matrix was set entirely to zero. On the other hand, when $\alpha = 0$ we recover the standard (dense) QMLE estimates for the DFM.

In the sparse case (Fig. 3), we repeat the experiment for two settings of $p = 10$ and $p = 50$. For each α, we additionally track the average sparsity of the solution, with the lambda which achieves the true level of sparsity (on average) given by the vertical lines. We see that the sparsity aids estimation reducing both bias and variance in the loadings until the optimal level of sparsity is attained. This is as one may expect, since the shrinkage imposed by the ℓ_1 penalty aligns with an appropriate (sparse) prior in this setting. We remark that the variance is the dominance term in the MSE for the loadings. When examining the bias of the factors (middle-bottom panel), the empirical expectation is taken by looking at the difference between the simulated and estimated factors, where we should remember that the simulated factors are themselves a random variable (as opposed to a constant for the loading matrix). An interesting observation for the factors, is that as the level of sparsity approaches the optimal level (for $p = 50$), we see that the variance of the factor estimates increases slightly. This aligns with the discussion earlier, whereby the factor estimates may be seen to be obtained by putting more weight on a subset of data-points. However, when the level of sparsity is optimal (and the estimated loadings experience low error), estimation

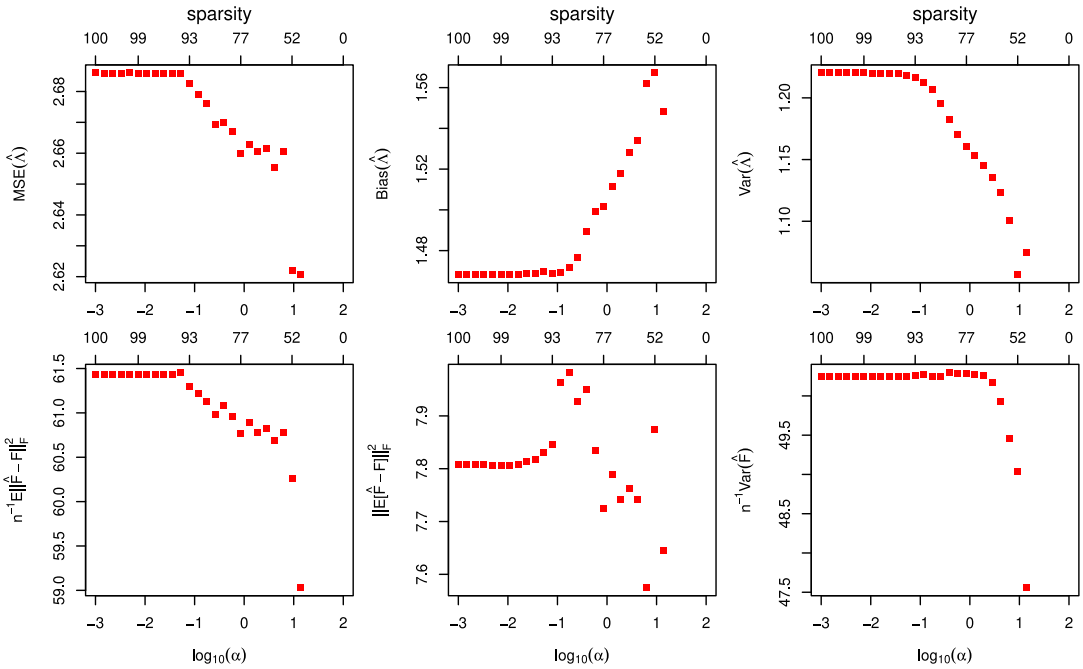

Fig. 4: Bias and variance for estimated loading matrices and factors as a function of regularisation α in the case where the true $\mathbf{\Lambda}$ is dense ($p = 50, n = 50$). The average level of sparsity in the estimates for a given α is plotted on the top axes.

of the factors is still improved relative to the dense model.

The dense case compliments the above analysis, as our prior is not aligned with the true loading matrix, which is dense. The results for this setting are given in Fig. 4, for $p = 50$, $n = 50$. In this case, the bias-variance trade-off is more visible, e.g., the bias of the loadings increases (as one may expect) as the regulariser gets stronger, and the solution gets sparser (the average sparsity of the solution is given on the top axis). Interestingly, we see that the overall benefit of the regulariser is still present, in that the MSE and variance of the loadings are reduced when a sparse representation is adopted (estimated). Again, in contrast to the sparse case (Fig. 3), we see that the ''bias" of the factors is increased relative to the sparse case, however, the MSE of the factors is still reduced when using the regularised estimator, as compared to the regular DFM (where $\alpha = 0$).

5.3 Forecasting performance

An important motivation for the EM approach is that our framework can readily handle patterns of missing data. To examine this in more detail, we examine the ability of SDFM to forecast missing data at the end of the sample. We simulate data according to the generating process above (Eq. 20) with $n = 200$, $p = 64$ and consider $\rho = \{0, 0.6, 0.9\}$, and assume different patterns of missing data at the end of the sample. We consider a 1-step ahead forecast case where we set 25%, 50%, 75% and then 100% of variables to be missing in the final row of $\boldsymbol{X}$. When allocating variables to be missing we split the data up into the two loading blocks and set the first 25%, 50%, 75% and 100% of each loading block to be missing. For example, the variables 1 to 8 and 33 to 40 are missing in the 25% missing scenario. We are interested in forecasting the missing data in the final row of $\boldsymbol{X}$ and we calculate the average MAE over 100 experiments.

We make comparisons with a sparse vector-autoregression (SVAR) model[6] as this is a very

[6]The SVAR algorithm is implemented using the *BigVAR* R package available on CRAN. This has a built-in cross-validation mechanism to tune for the best ℓ_1-penalty parameter which we use in our simulations.

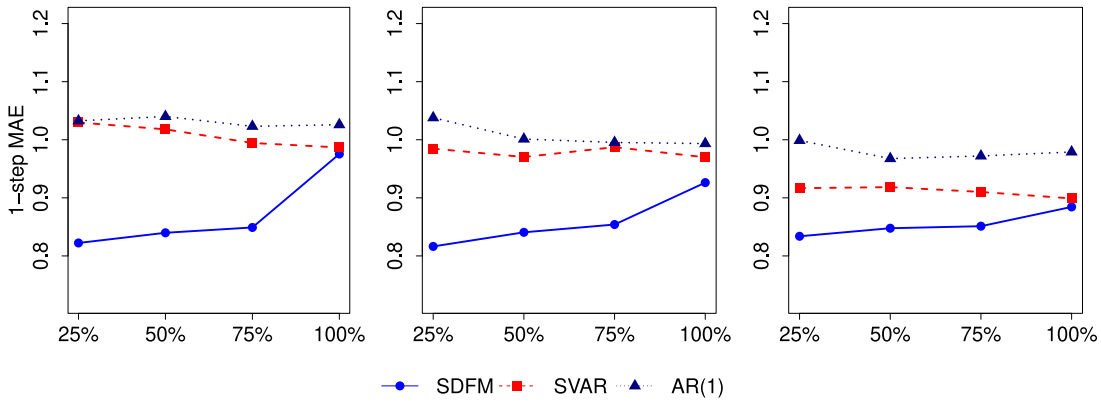

Fig. 5: Average MAE score forecasting, as a function of the level of missing data in the last sample. From left-right: $\rho = 0$, $\rho = 0.6$, $\rho = 0.9$. Plot indicates the 50th percentile of performance across 100 experiments with $n = 100$, $p = 64$.

popular alternative forecasting strategy for high-dimensional time series that is based on sparse assumptions. As our factors are generated using a VAR(1) process with a sparse auto-regression matrix, we are interested to see whether SVAR will be able to capture the cross-factor auto-correlation when producing forecasts. We also apply a standard AR(1) process to each of the variables needing to be forecasted as a benchmark comparison.

Figure 5 displays the results of the simulations plotting MAE for each of the 3 methods and each simulation setting. In all settings we find SDFM to outperform both SVAR and AR(1). When ρ is set to be 0.9, we find SVAR does improve its forecasting performance as opposed to when $\rho = 0$ as the VAR(1) process driving the factors becomes more prominent. The results confirm SDFM's ability to make use of variables that are present at the end of the sample when forecasting the missing variables. We see this by the rise in MAE when 100% of the variables are missing at the end of the sample and the model can no longer utilise available data in this final row. The MAE remains fairly flat as the amount of missingness rises from 25% to 75% showing SDFM's ability to forecast correctly even when there is small amount of data available at the end of the sample.

5.4 Computational efficiency

To assess the computational scalability, we simulate from a sparse DFM where $\boldsymbol{\Lambda} = \boldsymbol{I}_r \otimes \mathbf{1}_{p/r}$ and $\boldsymbol{\Sigma}_\epsilon = \boldsymbol{I}_p$, and the factors are a VAR(1) with $\boldsymbol{A} = 0.8 \times \boldsymbol{I}_r$ and $\boldsymbol{\Sigma}_u = (1 - 0.8^2) \times \boldsymbol{I}_r$. We record the number of EM iterations and the time they take for each ℓ_1-norm parameter α up to the optimal ℓ_1-norm parameter $\hat{\alpha}$ and then take the average time of a single EM iteration. We repeat the experiment ten times for each experimental configuration.

The results are presented in Figure 6, which demonstrates scalability as a function of n, and p, under different assumptions on the number of factors $r = 2, 4, 6, 8$. As expected, the cost is approximately linear in n and p, with increasing cost as a function of the number of factors r. The results demonstrate the utility of using the univariate smoothing approach as well as the matrix decomposition when calculating required inversions.

6 The dynamic factors of energy consumption

This section details application of SDFM to a real-world problem, namely the forecasting and interpretation of energy consumption. Beyond forecasting consumption in the near-term future, our aim here is to also characterise the usage in terms of what may be considered typical consumption profiles. These are of specific interest to energy managers and practitioners, as understanding how energy is consumed in distinct buildings can help target interventions and strategy to reduce waste. We also highlight how SDFM, and in particular

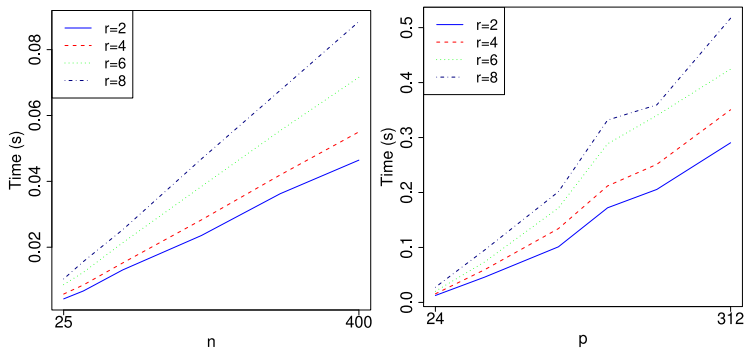

Fig. 6: Summary of computational cost. Top: as a function of n, with fixed $p = 24$. Bottom: as a function of p, with fixed $n = 100$. Average performance across 10 experiments.

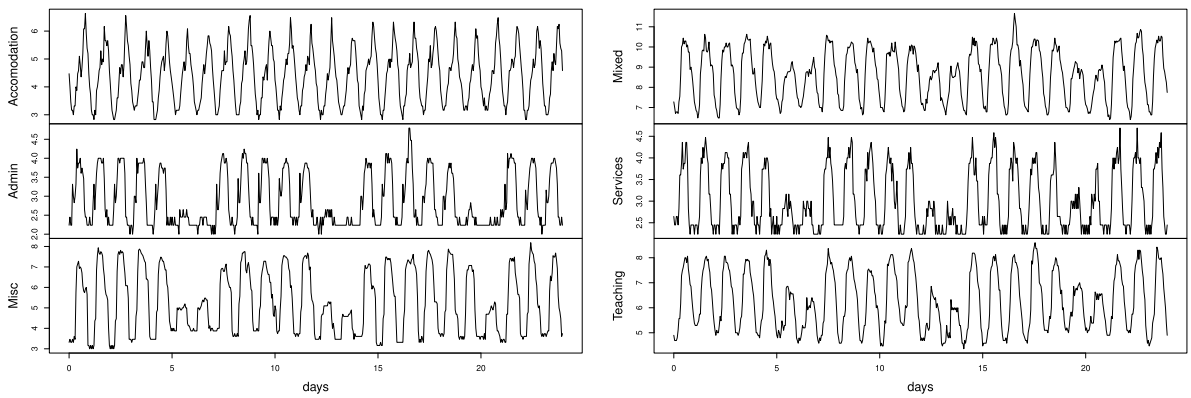

Fig. 7: Example of time series readings for the 24 days under analysis. The figures present the square root of the consumption in each hour ($\sqrt{6\text{kWh}}$) for different types of building, and illustrate the diverse nature of consumption.

our EM algorithm, can be used to impute missing data and provide further insight.

6.1 Data and preprocessing

In this application, the data consists of one month of electricity consumption data measured across $p = 42$ different buildings on our universities campus. This data is constructed based on a larger dataset, which monitors energy at different points throughout a building, in our case, we choose to aggregate the consumption so that one data stream represents the consumption of a single building. The data is gathered at 10 minute intervals (measuring consumption in kWh over that interval), resulting in $n = 3,456$ data points spanning 24 days worth of consumption in November 2021, we further hold out one day $n_{\text{test}} = 144$ data points to evaluate the out-of-sample performance of the DFM model. An example of time series from the dataset is presented in Figure 7. There are many alternative ways one may wish to model this data, however, one of the key tasks for energy managers is to understand how consumption in this diverse environment is *typically* structured. This is our primary objective in this study, i.e. we wish to extract typical patterns of consumption that can well represent how energy is used across the campus. To this end, we decide not to remove the relatively clear seasonal (daily) patterns in consumption prior to fitting the factor model, the hope being, that these patterns will somehow be pervasive in the derived factors.

Whilst we do have metadata associated with each of these buildings for sensitivity purposes we choose to omit this in our discussions here, the buildings are presented as being approximately categorised under the following headings:

Accomodation: Student residences, and buildings primarily concerned with accomodation/student living.
Admin: Office buildings, e.g. HR, administration, and central university activities.
Misc: Other student services, e.g. cinema, shopping, sports facilities.
Mixed: Buildings which mix teaching and accomodation. For instance, seminar rooms on one floor with accomodation on another.
Services: Management buildings, porter/security offices.
Teaching: Teaching spaces like lecture theatres, seminar rooms.

6.2 Factor estimates and interpretation

To estimate factors we first choose a number of factors according to criterion (18), which leads to 4 factors being specified as seen in Fig. 8. Next, we apply SDFM via the EM procedure in Algorithm 1. We run the algorithm to scan across a range of α parameters, and in this case, the BIC criteria suggests to impose moderate sparsity corresponding to $\alpha \approx 0.01$. One may note in Figure 8 that there is a second dip in the BIC criteria around $\alpha \approx 0.03$ after which the BIC rapidly rises until the cutoff constraint, after which all $\hat{\Lambda}_{ij}$ are set to zero. In this case, the sparsity pattern of the two above values of α appear very similar, and the loading of the variables on the factors appears largely stable as a function of α. To give some intuition, the loadings $\hat{\mathbf{\Lambda}}$ for $\alpha = 0.01$ and $\alpha = 0$ are visualised in Fig 9, a visualisation for the corresponding factors $\boldsymbol{a}_{t|n}$ are given in Fig. 10.

For brevity, we focus on analysing the results of the SDFM model. Of particular interest for the energy manager is the interpretation of consumption that the model provides, where the impact of sparsity is most prominent for the third and fourth factors in this case. A visualisation of the factor behaviour on a typical weekday is given in Figure 11 where there is a clear ordering in the uncertainty surrounding the factor behaviour,

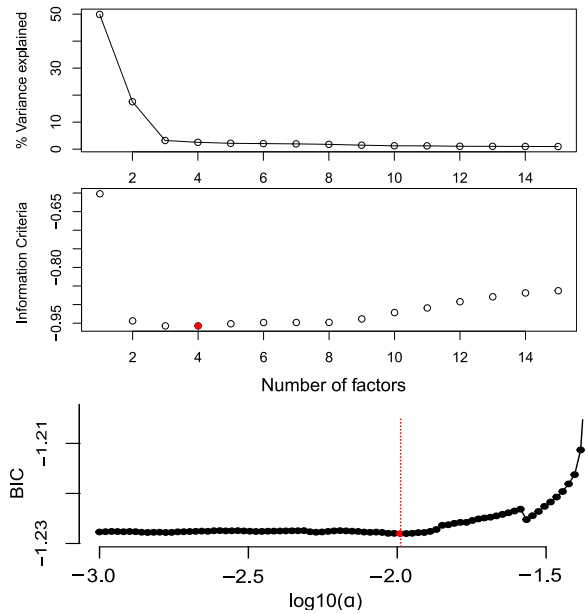

Fig. 8: Top: Proportion of variance explained, based on PCA applied to the scaled and pre-processed (interpolated) dataset. Middle: Information Criteria (18) as a function of number of retained factors r. Bottom: BIC as a function of α, vertical line indicates minimiser and the α used in the subsequent analysis.

e.g. Factor one has small confidence intervals, whereas Factor 4 has more uncertain behaviour, especially during the working day. Interestingly, the SDFM only really differs from the regular DFM in these third and fourth factors, where the latter exhibits slightly greater variation in behaviour. The SDFM is able to isolate these further factors to specific buildings. For example, the building identified by the circle in Fig. 9 is known to be active primarily throughout the night, and we see its factor loadings reflect this, e.g. the regular working day cycles for Factor 1 are not present, however, the evening and early morning features (Factors 3, and 4) are represented. For the teaching buildings, we see that the loading on Factor 2, and 3, are negative, indicating a sharp drop-off in energy consumption in the evening/overnight, again, this aligns with our expectations based on the usage of the facilities.

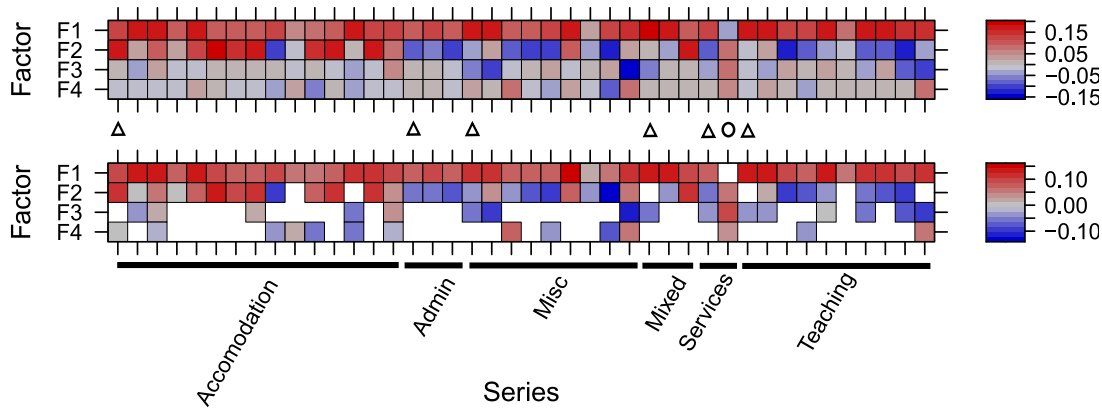

Fig. 9: Estimated factor loadings for the regular DFM (top) and SDFM (bottom). Series are categorised according to one of six building types, triangles indicate the example series plotted in Fig 7.

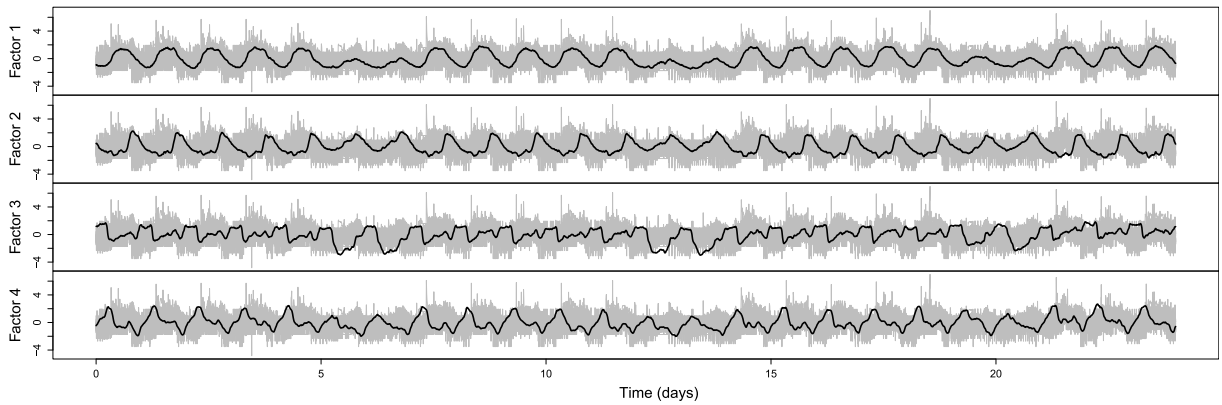

Fig. 10: Estimated factors (black) with original data (grey) as a function of time using the optimal $\alpha = 0.01$ chosen according to BIC. When multiplied by the factor loadings (top) gives the estimated common component.

6.3 Forecasting performance

The primary motivation for applying SDFM in the context of this application is to aid in interpreting and understanding the consumption across campus. However, it is still of interest to examine how forecasts from the DFM compare with competitor methods. For consistency, we here provide comparison to the AR(1) and sparse VAR methods detailed earlier. These models all harness a simple autoregressive structure to model temporal dependence, specifically regressing only onto the last set of observations (or factors), i.e. they are Markov order 1. Our experiments asses performance of the models in forecasting out-of-sample data, either $h = 6$ steps ahead (1 hour), or $h = 36$ steps ahead (6 hours). The forecasts are updated in an expanding window manner, whereby the model parameters are estimated on the 24 days of data discussed previously, the forecasts are then generated sequentially based on $n + t = 1, \ldots, n_{\text{test}} = 144 - h$ observations. An example of the forecasts generated (and compared to the realised consumption) is given in Figure 12. A striking feature of the DFM based model is its ability to (approximately) time the increases/decreases in consumption associated with the daily cycle. These features in the AR(1) and sparse VAR model are only highlighted after a period of h steps has passed, e.g. the models cannot anticipate the increase in consumption.

A more systematic evaluation of the forecast performance is presented in Figure 13, where the average error is calculated for each building, for each of the different models. We see that for the

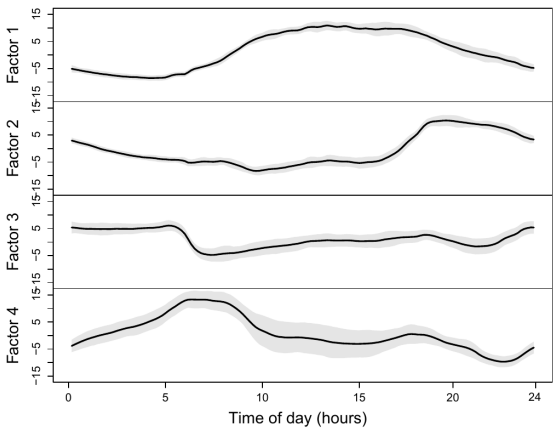

Fig. 11: Average factor profile as a function of time-of-day, $t = 0$ corresponding to midnight. The solid line is a pointwise average of the factor $\hat{\boldsymbol{a}}_{t|n}$ across the 18 weekdays in the sample, confidence intervals are constructed as ± 1.96 the standard-deviation.

1 hour ahead forecasts, all methods perform similarly, with the SDFM winning marginally, and the AR(1) forecasts demonstrating more heterogeneity in the performance. There is no clear winner across all the buildings, for most (30) buildings the SDFM forecasts prove most accurate, with the AR being best on 2, and the SVAR winning on the remaining 10. Moving to the 6 hour ahead forecasts, the dominance of the SDFM becomes clear, winning across 39 of the buildings, and the AR method winning on 3. Interestingly, the SVAR fails to win on any building, falling behind the simpler AR approach. This suggests, that in this application the activity of one building may not impact that of another across longer time-frames, however, the behaviour of the latent factors (common component) does provide predictive power.

One could reasonably argue that we should not use these competitor models in this way for forecasting, e.g. we would likely look to add seasonal components corresponding to previous days/times, and/or potentially a deterministic (periodic) trend model. However, these extensions can also potentially be added to the DFM construction. Instead of absolutely providing the best forecasts possible, this case-study aims instead to highlight the differences in behaviour across the different classes of models (univariate, multivariate sparse VAR, and sparse DFM), and the fact that the SDFM can borrow information from across the series in a meaningful way, not only to aide interpretation of the consumption, but also to provide more accurate forecasts by harnessing the common component.

7 Discussion

In this paper, we have presented a novel method for performing estimation of sparse Dynamic Factor models via a regularised Expectation Maximisation algorithm. Our analysis of the related QML estimator provides support for its ability to recover structure in the factor loadings, up to permutation of columns, and scaling. To our knowledge this is the first time the QMLE approach has been studied for the sparse DFM model, and our analysis extends recent investigations by Despois and Doz (2022); Uematsu and Yamagata (2022) using sparse PCA based approaches. When factors are thought to be dependent, e.g. as in our VAR(1) construction, the QMLE approach appears particularly beneficial relative to SPCA. We also validate that simple BIC based hyperparameter tuning strategies appear to be able to provide reasonable calibration of sparsity in the high-dimensional setting.

There is much further work that can be considered for the class of sparse DFM models proposed here, for example looking at developing theoretical arguments on consistency, of both factor loadings, and the factor estimates themselves. In this paper, we opted for an empirical analysis of the EM algorithm, which we believe is more immediately useful for practitioners. Theoretical analysis of the proposed estimation routine is challenging for several reasons. First, one would need to decide whether to analyse the theoretical minimiser (QMLE), or the feasible estimate provided by the EM algorithm. Second, we need to consider the performance as a function of both n and p. For example, Proposition 2 from Barigozzi and Luciani (2022) gives theoretical results for the consistency of factor loadings for the regular unregularised QMLE and for a dense DFM model. A further line of work would be to generalise these results to the SDFM setting, for instance, can we show a result analogous to Theorem 1 in Bai and Li (2016), that shows the QMLE estimator

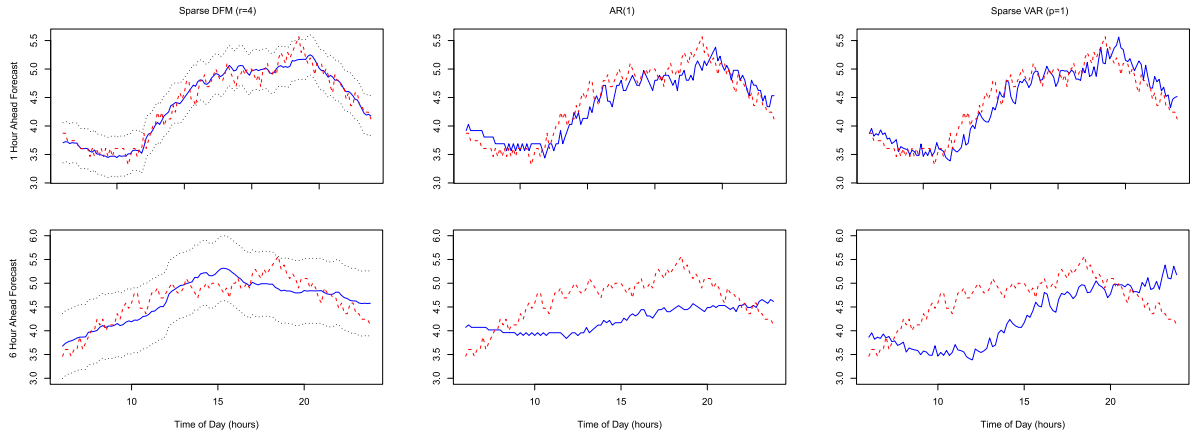

Fig. 12: Example of predicted consumption ($\sqrt{\text{kWh}}$) in one (accommodation) building on the campus. The top row represents 1 hour ahead forecasts based on an expanding window, whilst the bottom represents 6 hour ahead forecasts. The SDFM and SVAR are tuned on the 24 days of data prior to that presented in the figure. Confidence intervals for the SDFM are based on $1.96 \times [\hat{\mathbf{\Lambda}} \boldsymbol{P}_{t|n} \hat{\mathbf{\Lambda}}^\top + \hat{\mathbf{\Sigma}}_\epsilon]_{ii}^{1/2}$

.

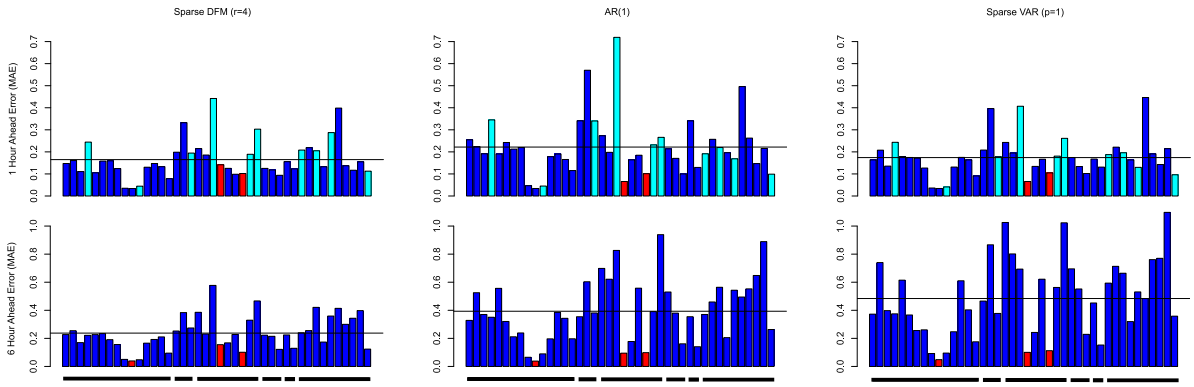

Fig. 13: Forecast errors (MAE) for each building for (top) 1 hour ahead forecast, and (bottom) 6 hour ahead forecast. Performance evaluated on one hold out day ($144 - h$ data points). Each bar is colored according to which method performs best for that building. Blue: SDFM, red: AR(1), navy: SVAR. The solid black line indicates average performance across all buildings, the grouping of buildings is indicated via the dashed line under the plots.

.

of the loadings is equivalent to the OLS estimator applied to the true factors? These kind of approaches could potentially enable a formal comparison of the sparse PCA based approaches and our QMLE approach.

On a more methodological front, one could consider extending the regularisation strategy presented here to look at different types of sparsity assumption, or indeed to encode other forms of prior. Two potential extensions could be to relax the assumption that the factor loadings remain constant over time, or adopt a group-lasso type regularisation on the loadings. The latter would enable users to associate factors with pre-defined sub-sets of the observed series, but still in a somewhat data-driven manner. For instance, in the energy application we could consider grouping the series via type of building and encouraging sparsity at this grouped level, rather than at the building level. This could be particularly useful

if we consider the application to smart meters at the sub-building, e.g. floor-by-floor, or room-by-room level. One of the benefits of the ADMM optimisation routine developed here is that it easily extended to these settings. Further work can also consider jointly tuning the level of sparsity alongside the number of factors to be estimated, e.g., via a joint BIC measure. In general, we reflect that the optimal choice of number of factors (or rank) is an open question, other information criteria, or indeed rank regularisation methods (Bai and Ng, 2019) could be investigated here.

A final contribution of our work is to demonstrate the application of SDFM on a real-world dataset, namely the interpretation and prediction of smart meter data. Traditionally, application of DFM based models has been within the economic statistics community, however, there is no reason they should not find much broader utility. The application to modelling energy consumption in a heterogeneous environment is novel in itself, and serves to raise awareness of how the DFM can help provide an exploratory tool for complex high-dimensional time series. In this case, not only is the sparse DFM beneficial for interpreting consumption patterns, identifying distinctive profiles of buildings that qualitatively align with our intuition, e.g. based on type of use, but also in forecasting consumption ahead of time. With the latter, the DFM can borrow from buildings with similar consumption profiles to better predict consumption peaks/dips further ahead in time. Further applications of our proposed SDFM estimator can be found in our software paper (Mosley et al, 2023), that also provides guidance on how to implement the methods in R. In particular, the paper demonstrates an application of the DFM to predicting trade-in-goods flows showing that assuming sparse factors can improve forecast performance relative to the DFM, and that the structure of the loadings can be substantially altered as a function of α.

To conclude, we remark that adding sparsity in the DFM framework seems to be feasible in the sense that we can construct estimators that can reliably recover this structure. As shown in our experiments, the QMLE approach we propose here compares favourably with more simplistic sparse PCA approaches, especially in the setting where there is dependence between the factors and in the high-dimensional setting. Overall the sparse DFM provides a useful alternative to other high-dimensional time series models, for both predictive and inferential tasks. An additional benefit of the EM approach is its ability to readily handle arbitrary patterns of missing data, an issue often faced in the analysis of high-dimensional time series.

Supplementary information. Code to replicate the smart meter example presented in this paper can be found on GitHub (github.com/alexgibberd/dfmEnergyExample). The sparse DFM package used to implement the EM algorithm can be found on CRAN or via GitHub (github.com/mosleyl/sparseDFM). We refer the reader to Mosley et al (2023) for further details on how to use the package.

Acknowledgments. AG and TTC acknowledge funding from the EPSRC grant EP/T025964/1. AG and LM were supported by the ESRC grant ES/V006339/1. LM acknowledges support from the STOR-i Centre for Doctoral Training and the Office for National Statistics.

References

Bai J (2003) Inferential theory for factor models of large dimensions. Econometrica 71(1):135–171

Bai J, 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

Bai J, Ng S (2002) Determining the number of factors in approximate factor models. Econometrica 70(1):191–221

Bai J, Ng S (2008) Forecasting economic time series using targeted predictors. Journal of Econometrics 146:304–317

Bai J, Ng S (2013) Principal components estimation and identification of static factors. Journal of Econometrics 176:18–29

Bai J, Ng S (2019) Rank regularized estimation of approximate factor models. Journal of Econometrics 212:78–96

Bańbura M, Modugno M (2014) Maximum likelihood estimation of factor models on datasets

with arbitrary pattern of missing data. Journal of Applied Econometrics 29(1):133–160

Banbura M, Giannone D, Reichlin L (2010) Nowcasting. ECB Working Paper

Barigozzi M, Luciani M (2022) Quasi maximum likelihood estimation and inference of large approximate dynamic factor models via the EM algorithm. arXiv Preprint arXiv:191003821

Boyd S, Parikh N, Chu E, et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning 3(1):1–122

Carroll JB (1953) An analytical solution for approximating simple structure in factor analysis. Psychometrika 18(1):23–38

Croux C, Exterkate P (2011) Sparse and robust factor modelling. Tinbergen Institute Discussion Paper TI 122/4

De Jong P, Mackinnon MJ (1988) Covariances for smoothed estimates in state space models. Biometrika 75(3):601–602

Despois T, Doz C (2022) Identifying and interpreting the factors in factor models via sparsity: Different approaches. HAL Id: halshs-02235543v3

Doz C, Fuleky P (2020) Dynamic factor models. In: Macroeconomic Forecasting in the Era of Big Data. Springer, p 27–64

Doz C, Giannone D, Reichlin L (2011) A two-step estimator for large approximate dynamic factor models based on Kalman filtering. Journal of Econometrics 164(1):188–205

Doz C, Giannone D, Reichlin L (2012) A quasi-maximum likelihood approach for large, approximate dynamic factor models. Review of Economics and Statistics 94(4):1014–1024

Durbin J, Koopman SJ (2012) Time Series Analysis by State Space Methods. Oxford University Press

Fisher AJ (2015) Toward a dynamic model of psychological assessment: Implications for personalized care. Journal of Consulting and Clinical Psychology 83(4):825

Foroni C, Marcellino M (2014) A comparison of mixed frequency approaches for nowcasting Euro area macroeconomic aggregates. International Journal of Forecasting 30(3):554–568

Freyaldenhoven S (2023) Identification through sparsity in factor models: the l1-rotation criterion. Federal Reserve Bank Philadelphia (Working Paper)

Geweke J (1977) The dynamic factor analysis of economic time series. In: Latent Variables in Socio-Economic Models. North-Holland

Giannone D, Reichlin L, Small D (2008) Nowcasting: The real-time informational content of macroeconomic data. Journal of Monetary Economics 55(4):665–676

Grassi S, Proietti T, Frale C, et al (2015) EuroMInd-C: A disaggregate monthly indicator of economic activity for the Euro area and member countries. International Journal of Forecasting 31(3):712–738

Harvey A (1996) Intervention analysis with control groups. International Statistical Review/Revue Internationale de Statistique 64(3):313–328

Hu Y, Chi EC, Allen GI (2016) ADMM algorithmic regularization paths for sparse statistical machine learning. In: Splitting Methods in Communication, Imaging, Science, and Engineering. Springer International Publishing, p 433–459

Jennrich RI, Sampson P (1966) Rotation for simple loadings. Psychometrika 31(3):313–323

Kaiser HF (1958) The varimax criterion for analytic rotation in factor analysis. Psychometrika 23(3):187–200

Koopman SJ, Durbin J (2000) Fast filtering and smoothing for multivariate state space models. Journal of Time Series Analysis 21(3):281–296

Kristensen JT (2017) Diffusion indexes with sparse loadings. Journal of Business & Economic Statistics 35(3):434–451

Lee D, Baldick R (2016) Load and wind power scenario generation through the generalized dynamic factor model. IEEE Transactions on Power Systems 32(1):400–410

Lin T, Ma S, Zhang S (2015) On the global linear convergence of the ADMM with Multi-Block variables. SIAM Journal on Optimization 25(3):1478–1497

Liu X, Wallin G, Chen Y, et al (2023) Rotation to sparse loadings using lp losses and related inference problems. Psychometrika 88:527–553

Luciani M (2015) Monetary policy and the housing market: A structural factor analysis. Journal of Applied Econometrics 30(2):199–218

Magnus JR, Neudecker H (2019) Matrix Differential Calculus with Applications in Statistics and Econometrics. Wiley

Mariano RS, Murasawa Y (2010) A coincident index, common factors, and monthly real GDP. Oxford Bulletin of Economics and Statistics 72(1):27–46

Molenaar P (1985) A dynamic factor model for the analysis of multivariate time series. Psychometrika 50(2):181–202

Mosley L, Chan TS, Gibberd A (2023) sparseDFM: An R package to estimate dynamic factor models with sparse loadings. arXiv Preprint arXiv:230314125

Poncela P, Ruiz E, Miranda K (2021) Factor extraction using Kalman filter and smoothing: This is not just another survey. International Journal of Forecasting 37(4):1399–1425

Rohe K, Zeng M (2020) Vintage factor analysis with varimax performs statistical inference. arXiv Preprint arXiv:200405387

Sargent TJ, Sims CA, et al (1977) Business cycle modeling without pretending to have too much a priori economic theory. New Methods in Business Cycle Research 1:145–168

Shumway RH, Stoffer DS (1982) An approach to time series smoothing and forecasting using the EM algorithm. Journal of Time Series Analysis 3(4):253–264

Stock JH, Watson M (2011) Dynamic factor models. Oxford Handbooks Online

Stock JH, Watson MW (2002) Forecasting using principal components from a large number of predictors. Journal of the American Statistical Association 97(460):1167–1179

Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58(1):267–288

Uematsu Y, Yamagata T (2022) Estimation of sparsity-induced weak factor models. Journal of Business and Economic Statistics 41:213–227

Watson MW, Engle RF (1983) Alternative algorithms for the estimation of dynamic factor, mimic and varying coefficient regression models. Journal of Econometrics 23(3):385–400

Wu H, Chan S, Tsui K, et al (2013) A new recursive dynamic factor analysis for point and interval forecast of electricity price. IEEE Transactions on Power Systems 28(3):2352–2365

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. Journal of Computational and Graphical Statistics 15(2):265–286