Dynamic Factor Models

Seth Leonard
seth@ottoquant.com

OttoQuant
www.ottoquant.com

May 20, 2020

Beginning with vector autoregressions (VARs) introduced by Sims (1980), multivariate methods for forecasting time series data have become standard practice in economics among academia, central banks, and research institutions. Since then multivariate predictive modeling has evolved to encompass a number of challenges presented by real world data. Watson and Engle (1983) introduced a simple algorithm for estimating dynamic factor models (DFMs) by maximum likelihood. By modeling the driving process behind (multivariate) observed data as latent (unobserved), DFMs are able to incorporate missing observations without falsified imputed data, and allow the modeling of noisy observations due to measurement error. Mariano and Murasawa (2003) present a DFM framework for mixed frequency data, allowing practitioners to incorporate, for example, monthly and quarterly data without having to aggregate observations to the lowest frequency in the data. Giannone et al. (2005) and Giannone et al. (2008) pioneered applications of DFMs to nowcasting, with a specific emphasis on using the real time data flow to update predictions as soon as any new information becomes available. This application is particularly appropriate as it mirrors applications of Kalman (1960)'s original work (on which DFMs are based): real time estimates of Apollo spacecraft positions based on noisy and intermittent measurement data. Since then work on state space models, models that break complicated processes into two parts: observation or measurement and prediction or transition, has proliferated. Bańbura et al. (2015) use a state space framework for conditional forecasts using VARs, and McCracken and McGillicuddy (2019) explore a wide range of approaches to con-

ditional forecasting both within and outside of state space methods. Conditional forecasting, or quantitative scenario analysis, has a wide range of applications, from GDP forecasts conditional on monetary or fiscal policy scenarios to stress testing financial institutions.

Though we have already derived the Kalman filter and smoother, these notes begin by adding one more tool which will make practical implementation of DFMs much more efficient: the Durbin-Koopman disturbance smoother. We will then introduce several methods for estimating DFMs: estimation by principal components, by maximum likelihood following Watson and Engle (1983), and Bayesian estimation.

1 The Durbin-Koopman Disturbance Smoother

1.1 The Standard Filter and Smoother

Though we have already derived the Kalman filter and smoother, it is worth restating those results here. We begin with the measurement equation

$$y_t = Hx_t + \varepsilon_t \tag{1}$$

and the transition equation

$$z_t = Az_{t-1} + e_t \tag{2}$$

where ε_t and e_t are normally distributed error terms with the covariance matrix

$$Cov \begin{bmatrix} e_t \\ \varepsilon_t \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$$

In the above y_t are noisy observations, z_t stacked factors $z_t = \begin{bmatrix} x_t & x_{t-1} & \dots & x_{t-p} \end{bmatrix}$ with p lags, and n_t predetermined, exogenous variables.

Given the parameters H, A, Q, and R, estimates of factors or estimates of missing

series in y_t derive from the Kalman filter and smoother. Our Kalman filter is

$$\begin{aligned} z_{t|t-1} &= Az_{t-1|t-1} \\ P_{t|t-1} &= AP_{t-1|t-1}A' + Q \\ y_{t|t-1} &= \tilde{H}z_{t|t-1} \\ S_t &= \tilde{H}P_{t|t-1}\tilde{H}' + R \\ C_t &= P_{t|t-1}\tilde{H}' \\ z_{t|t} &= z_{t|t-1} + C_t S_t^{-1}(y_{t|t} - y_{t|t-1}) \\ P_{t|t} &= P_{t|t-1} - C_t S_t^{-1} C_t' \end{aligned}$$

and the Kalman smoother is

$$\begin{aligned} z_{t|T} &= z_{t|t} + g_t(x_{t+1|T} - z_{t+1|t}) \\ P_{t|T} &= P_{t|t} - g_t(P_{t+1|t} - P_{t+1|T})g_t' \end{aligned} \tag{3}$$

where $g_t = P_{t|t}A'P_{t+1|t}^{-1}$. $\tilde{H}$ in the above incorporates a helper matrix J to extract, in the simplest example, contemporaneous factors from z_t. That is, $J = \begin{bmatrix} I_m & 0 & 0 & \dots \end{bmatrix}$ and $\tilde{H} = HJ$. In the above notation $x_{t|t-1}$ refers to our estimate of x_t given observations through period $t-1$ while $x_{t|t}$ is our estimate of x_t given observations through t, and $x_{t|T}$ is our estimate of x_t conditional on all available data through period T. Note that in the above $K_t = C_t S_t^{-1}$ is the Kalman gain, $\nu_t = y_{t|t} - y_{t|t-1}$ is the prediction error, and thus $K_t\nu_t$ is our forecast update.

Smoothing here requires we store the following matrices (or vectors) from the filter at every period t: $P_{t|t}$, $P_{t+1|t}$, $z_{t|t}$, and $z_{t+1|t}$. Additionally, we must invert the matrix $P_{t+1|t}$ at each iteration of the smoother. This can be computationally burdensome for even moderately sized models. Consider an example with four factors and five lags. In this case we will be inverting the 20×20 matrix $P_{t+1|t}$ in each time period. For mixed frequency models inverting $P_{t+1|t}$ can become prohibitive. For a daily/weekly/monthly model with three factors and three lags at a monthly (30 day) frequency $P_{t+1|t}$ will be a 270×270 matrix. The Durban-Koopman disturbance smoother allows us to avoid these cumbersome calculations resulting in much faster and more efficient model estimation.

1.2 Durban Koopman (2001) Filtering and Smoothing

Sticking with the above notation, Durbin and Koopman (2001) write the Kalman filter as

$$\nu_t = y_t - \tilde{H} x_{t|t-1} \tag{4}$$

$$S_t = \tilde{H} P_{t|t-1} \tilde{H}' + R \tag{5}$$

$$K_t = A P_{t|t-1} \tilde{H}' S_t^{-1} \tag{6}$$

$$z_{t+1|t} = A z_{t|t} + K_t \nu_t \tag{7}$$

$$L_t = A - K_t \tilde{H} \tag{8}$$

$$P_{t+1|t} = A P_{t|t} L_t' + Q \tag{9}$$

Note that in this notation the Kalman gain in (7) updates the forecast for *next period* factors; factoring A out of equation (8) yields the same forecast update as the standard Kalman filter. Similarly, multiplying out the elements of (9) yields $P_{t+1|t} = A P_{t|t-1} A' - A C_t S_t^{-1} C_t' A' + Q$. For the above notation, the disturbance smoother is

$$r_t = \tilde{H}' S_t^{-1} \nu_t + L_t' r_{t+1} \tag{10}$$

with $r_T = 0$ from which we can recover the vector of errors (disturbances) as

$$\begin{bmatrix} \varepsilon_{t|T} \\ e_{t|T} \end{bmatrix} = \begin{bmatrix} R S_t^{-1} & -R K_t' \\ 0 & Q \end{bmatrix} \begin{bmatrix} \nu_t \\ r_t \end{bmatrix} \tag{11}$$

Note that in order to recover smoothed factors we must iterate r_t back to r_0. Equipped with $e_{t|T}$ we can recover smoothed estimates of the factors by initializing our smoothed factors as

$$z_1 = z_{1|0} + P_{1|0} r_0$$

or, since we typically set z_0 to zero simply $z_1 = P_{1|0} r_0$, and iterating forward again using

$$z_{t|T} = A z_{t-1|T} + e_{t|T}$$

In this case we must store the matrices (or vectors) ν_t, H, S_t^{-1}, L_t, $z_{t|t-1}$, and potentially K_t. Since filtering requires solving $\tilde{H} S_t^{-1}$ disturbance smoothing requires no additional matrix inversions and will thus be much more computationally efficient.

2 Estimation via Principal Components

By far the simplest and fastest way to estimate the factor model outlined in equations (1) and (2) is by principal components, though this approach has several major drawbacks. Perhaps most importantly, principal components solves a static problem. That is, when using principal components we find factors by looking at the measurement equation in isolation ignoring the inter-temporal correlations implied by the transition equation. Additional problems with principal components estimates are (1) data must be square and complete, a large drawback when using a methodology particularly adept at handling missing and noisy observations and (2) principal components limits our ability to fix parameters or apply prior beliefs to parameter estimates. Despite these drawbacks, when we have a data set without missing observations principal components provides a very quick and easy way to estimate a factor model. We begin by examining reduced rank regressions before moving on to principal components as a special case. Once we have our principal component estimates of factors we can estimate the transition equation by OLS and simply plug these results into the Kalman filter and, if desired, smoother to obtain our final factor estimates.

2.1 Reduced Rank Regression

Suppose we observe a $k \times t$ set of data Y that we wish to explain using a $s \times t$ set of data X, but that the number of series in X (denoted by s) is very large and that we think the relationship between X and Y can in fact be summarized by the data in an $m \times t$ matrix γX. That is, we would like to reduce the rank of X before estimating its relationship with Y. Assuming we want to model a linear relationship, we can write this problem succinctly as

$$Y = B\gamma X + \epsilon \tag{12}$$

where both B and γ are parameter matrices to estimate. If we want the least squares estimates of B and γ then we need to minimize

$$\min \left\{ tr\big(\underbrace{(Y - B\gamma X)}_{\epsilon} \underbrace{(Y - B\gamma X)'}_{\epsilon'} \big) \right\} \tag{13}$$

where tr denotes the trace of the covariance matrix for ϵ. Note that equation (12) will be observationally equivalent for

$$Y = \underbrace{B\theta}_{B^a} \underbrace{\theta^{-1}\gamma}_{\gamma^a} X + \epsilon$$

where B^a and γ^a are some alternative B and γ, thus we need to impose some sort of normalization on γ. It is convenient to impose

$$\gamma XX'\gamma' = I_m \tag{14}$$

From a simple OLS model we already know that given γ our estimate for B that minimizes our loss function (13) will be

$$B = YX'\gamma'(\gamma XX'\gamma')^{-1}$$

which, given our normalization (14) is just $YX'\gamma'$. Plugging this into our loss function we have

$$\min \Big\{ tr\big(YY' - 2YX'\gamma'\gamma XY' + YX'\gamma'\gamma XX'\gamma'\gamma XY'\big) \Big\}$$

which, given (14) is just

$$\min \Big\{ tr\big(YY' - YX'\gamma'\gamma XY'\big) \Big\} \tag{15}$$

Since the first term does not include γ we can re-write this minimization problem as a constrained maximization, using the property of a matrix trace that $tr(ABC) = tr(CAB) = tr(BCA)$ so that $tr(YX'\gamma'\gamma XY') = tr(\gamma XY'YX'\gamma')$,

$$\mathcal{L} = tr\big(\gamma XY'YX'\gamma' + \Lambda\big(I_m - \gamma XX'\gamma'\big)\big)$$

where the constraint comes from our normalization (14) and Λ is a diagonal matrix of Lagrange multipliers. The first order condition for this problem (see the appendix for useful matrix derivatives) is

$$2\gamma XY'YX' - 2\Lambda\gamma XX' = 0$$

or

$$\gamma XY'YX'(XX')^{-1} - \Lambda\gamma = 0$$

so that the solution for γ is the left eigenvectors of $XY'YX'(XX')^{-1}$. Since left eigenvectors are a bit annoying (Matlab automatically computes right eigenvectors) we can simply take the transpose of this result

$$(XX')^{-1}XY'YX'\gamma' - \gamma'\Lambda = 0 \tag{16}$$

so that γ' is given by the (right) eigenvectors of $(XX')^{-1}XY'YX'$ associated with the largest m eigenvalues, the diagonal elements of Λ (Λ is a diagonal matrix). Recall that the Lagrange multiplier is a measure of how strongly our constraint binds.[1] We chose the eigenvectors associated with the largest eigenvalues because those vectors give the factors γX which will have the largest impact on our objective function $tr\big(YX'\gamma'\gamma XY'\big)$. Once we have γ we can simply compute $B = YX'\gamma'$.

[1] Specifically Λ is the partial derivative of our objective function with respect to the constraint, in this case I_k.

2.2 Principal Components as a Special Case

Principal components is a special case of reduced rank regressions in which $X = Y$. That is, γY is a collection of $m < k$ series that we believe is sufficient to explain Y. If $X = Y$ then equation (16) simply becomes

$$YY'\gamma' - \gamma'\Lambda = 0 \tag{17}$$

in which case the solution for γ' is the (right) eigenvectors of YY' associated with the largest m eigenvalues. We can actually derive principal components from a slightly more general problem. Suppose we want to minimize the loss function

$$\min\{tr\big((Y - \gamma F)(Y - \gamma F)'\big)\} \tag{18}$$

where F can be any set of explanatory variables. Thus, unlike the problem in section 2.1, we are not restricting what the set of explanatory variables can be (previously it was X). We again need to impose a normalization of γ since $Y = \gamma\theta\theta^{-1}F + \epsilon$ will be observationally equivalent to the model in (18). In this case the normalization

$$I_m = \gamma'\gamma \tag{19}$$

is convenient. Minimizing first over F, the solution for the factors given γ is

$$F = (\gamma'\gamma)^{-1}\gamma'Y$$

or, using (19),

$$F = \gamma'Y$$

Plugging this back into our minimization problem, equation (18), we have

$$\min\{tr\big(YY' - 2\gamma\gamma'YY' + \gamma\gamma'YY'\gamma\gamma'\big)\}$$

Using the property of a matrix trace that $tr(ABC) = tr(CBA) = tr(ACB)$ and our normalization, equation (19) this result becomes

$$\min\{tr\big(YY' - Y'\gamma\gamma'Y\big)\}$$

which, again using our normalization, we can write as the constrained maximization problem

$$\mathcal{L} = tr\big(\gamma'YY'\gamma + \Lambda(I - \gamma'\gamma)\big)$$

with first order condition

$$YY'\gamma - \gamma\Lambda = 0$$

thus γ is the eigenvectors of YY' associated with the m largest eigenvalues (note that this γ is the transpose of the previous one we derived for reduced rank regressions). It is fairly simple in this context to calculate how much each principal component, in this case each row of γY, reduces our original loss function, equation (18). If we scale our loss function by $1/T$ and assume Y has zero mean our loss function is simply the trace of the covariance matrix of Y, denoted $tr(\Sigma_{YY})$. We can re-write this trace using the eigendecomposition of Σ_{YY} as[2]

$$tr\big(\Gamma\Lambda\Gamma'\big) = tr(\Gamma'\Gamma\Lambda) = tr(\Lambda)$$

where again we have used the properties of a matrix trace. Thus the trace of Σ_{YY} is the sum of its eigenvalues. With one principal component, that associated with the largest eigenvalue denoted λ_1, our scaled loss function is

$$\begin{aligned}
& tr\Big(\tfrac{1}{T}(Y-\gamma\gamma'Y)(Y-\gamma\gamma'Y)'\Big) \\
= \; & tr\Big(\tfrac{1}{T}(YY'-2\gamma\gamma'YY'+\gamma\gamma'YY'\gamma'\gamma\Big) \\
= \; & tr\Big(\tfrac{1}{T}(YY'-\gamma\gamma'YY')\Big) \\
= \; & tr\Big(\tfrac{1}{T}(YY'-\gamma'YY'\gamma)\Big) \\
= \; & tr\Big(\Gamma\Lambda\Gamma'-\gamma'\Gamma\Lambda\Gamma'\gamma\Big) \\
= \; & tr\left(\Lambda-\begin{bmatrix}\lambda_1 & 0 & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots\end{bmatrix}\right)
\end{aligned}$$

where the last equality comes from the fact that $\gamma'\Gamma$ is a matrix of zeros with a one in the upper left corner, as is $\Gamma'\gamma$. Thus the first principal component reduces our scaled loss function by λ_1. Similarly, the first two principal components will reduce our scaled loss function by $\lambda_1+\lambda_2$. The usual interpretation of this result is that the first m principal components explain $\frac{\sum_{i=1}^{m}\lambda_i}{\sum_{i=1}^{k}\lambda_i}$ of the variance of Y, where by variance of Y we mean the sum of the diagonal of the covariance matrix.

[2]From our first order condition for m factors using the scaled loss function, $\Sigma_{YY}\gamma-\gamma\Lambda$, we can write the system of equations for the complete set of factors when $m=k$ as $\Sigma_{YY}\Gamma=\Gamma\Lambda$ where Λ is a diagonal matrix of the eigenvalues of Σ_{YY} and the columns of Γ are the associated eigenvectors. Thus $\Sigma_{YY}=\Gamma\Lambda\Gamma'$ where we maintain our normalization $\Gamma'\Gamma=I_k$, and since Γ is now square it is also true that $\Gamma\Gamma'=I_k$

2.3 Summing Up

Beginning with our transition equation

$$Y_t = Hx_t + \varepsilon_t$$

if our set of k covariance stationary observables Y_t does not have missing observations then we can estimate the $k \times m$ matrix of loadings H by principal components as the (right) eigenvectors associated with the m largest eigenvalues of Σ_{YY} where Σ_{YY} is the (typically scaled to have ones on the diagonal) covariance matrix for Y_t, that is, $\Sigma_{YY} = (Y - \bar{Y})'(Y - \bar{Y})'$. Due to the normalizing assumption $I_m = H'H$ in equation (19) our estimate of the factors x_t is

$$H'Y_t = \hat{x}_t$$

and residuals are given by

$$\hat{\varepsilon}_t = Y_t - H\hat{x}_t$$

allowing us to estimate R. Typically we will use only the diagonal elements of R as Doz et al. (2012) show that this model, the approximate factor model, even if misspecified still consistently estimates factors. Using the factors $\hat{x}_t$ we can estimate the transition equation by OLS for parameters A and Q. These are nearly all we will need to run the Kalman filter and, if desired, smoother. The only thing that remains are initial values x_0 and P_0. Because x_0 is not known, it is good practice to initiate the filter with diffuse values, that is, specifying a large initial factor variance P_0. For example, we might begin our filter with $x_0 = 0_m$ and $P_0 = 10^5 I_m$.[3]

3 Maximum Likelihood Estimation via Watson and Engle (1983)

Maximum likelihood estimation of state space models using numerical techniques such as the function `optim()` in R or `fminsearch()` in Matlab provide an easy to program method for estimating model parameters. To use a numerical method we can simply write a function that returns the log likelihood and search for parameters that maximize it. The great drawback of numerical methods is that

[3] In the case of more than one lag p in the transition equation we would have $x_0 = 0_{m \times p}$ and $P_0 = 10^5 I_{m \times p}$.

they are computationally intensive and thus slow to converge and unfeasible for larger models. Additionally, numerical approaches may get stuck at local optima, thereby missing the global optimum of a problem, particularly when the number of parameters is large. Watson and Engle (1983) provide an alternative approach that overcomes these shortcomings of numerical routines.

3.1 The Algorithm

Watson and Engle (1983) propose an iterative scheme that describes an expectation-maximization (EM) algorithm for estimating state space models. The insight Watson and Engle (1983) for the purpose of estimating the model described in equations (1) and (2) is that, though we do not observe the true factors, we can still calculate the necessary moment matrices for A, H, Q, and R using an appropriate adjustment and, moreover, we can use the standard Kalman smoother to calculate this adjustment. I begin by re-stating the model for clarity. The measurement equation for k observables and m factors is

$$y_t = Hx_t + \varepsilon_t$$

and the transition equation is

$$x_t = Bz_{t-1} + e_t$$

where z_t is defined as

$$z_t = \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-p+1} \end{bmatrix}$$

and p denotes the number of lags in the transition equation. Then the moment matrices we need to estimate are, for B: $E(z_{t-1}z_{t-1}')$ and $E(x_t z_{t-1}')$; for H: $E(x_t x_t')$ and $E(y_t x_t')$; for Q: $E(e_t e_t')$; and for R: $E(\varepsilon_t \varepsilon_t')$. Proper estimation of the above moment matrices allows us to then compute maximum likelihood estimates of the model parameters.

In brief, the approach works by noting that, where $x_{t|T}$, $P_{t|T}$, $C_{t|T}$, and so on are the values calculated at the j^{th} iteration of the algorithm (I suppress the superscript j to keep the notation cleaner and use $z_{t|T}$ to denote our estimate of the factors z_t given observations $1:T$)

$$E(z_{t-1}z_{t-1}') = E\Big((z_{t-1|T} + (z_{t-1} - z_{t-1|T}))(z_{t-1|T} + (z_{t-1} - z_{t-1|T}))'\Big) \quad (20)$$

which we can estimate as

$$\frac{1}{T}\Big[\sum_t z_{t-1|T} z'_{t-1|T} + \sum_t P_{t-1|T}\Big]$$

$$E(x_t z'_{t-1}) = E\Big((x_{t|T} + (x_t - x_{t|T}))(z_{t-1|T} + (z_{t-1} - z_{t-1|T}))'\Big) \tag{21}$$

which we can estimate as

$$\frac{1}{T}\Big[\sum_t (x_{t|T}(z_{t-1|T})') + \sum_t C_{t|T}\Big]$$

giving the moment matrices for B;

$$E(x_t x'_t) = E\Big((x_{t|T} + (x_t - x_{t|T}))(x_{t|T} + (x_t - x_{t|T}))'\Big) \tag{22}$$

which we estimate as

$$\frac{1}{T}\Big[\sum_t x_{t|T} x'_{t|T} + \sum_t P^x_{t|T}\Big]$$

$$\begin{aligned} E(y_t x'_t) &= E\Big(y_t\big(x_{t|T} + (x_t - x_{t|T})\big)'\Big) \\ &= E(y_t x'_{t|T}) \end{aligned} \tag{23}$$

giving the moment matrices for H; and for the covariance matrices Q and R

$$e_t = (x_{t|T} - B z_{t-1|T}) + \big((x_t - x_{t|T}) - B(z_{t-1} - z_{t-1|T})\big) \tag{24}$$

so that we estimate $E(e_t e'_t)$ as

$$\frac{1}{T}\Big[\sum_t v_{t|T} v_{t|T} + \sum_t P^x_{t|T} \sum_t B P_{t-1|T} B' + \sum_t B C_{t|T} + \sum_t C'_{t|T} B'\Big]$$

and

$$\varepsilon_t = y_t - H x_{t|T} - H(x_t - x_{t|T}) \tag{25}$$

so that we estimate $E(\varepsilon_t \varepsilon'_t)$ as

$$\frac{1}{T}\Big[\sum_t \varepsilon_{t|T} \varepsilon_{t|T} + \sum_t H P^x_{t|T} H'\Big]$$

As suggested by Watson and Engle (1983) we can calculate the correlations $E\big((x_t - x_{t|T})(z_t - z_{t-1|T})'\big)$ by including an extra lag of x_t in the state vector of the Kalman filter and smoother. For example, if we wanted to estimate a model with three lags we would write the transition equation as

$$\begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix} = \begin{bmatrix} B_1 & B_2 & B_3 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \\ x_{t-4} \end{bmatrix} + \begin{bmatrix} v_t \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Denote the variance of this augmented vector of states in period t as

$$P_{t|T}^{WE} = \begin{bmatrix} \sigma_{t,t} & \sigma_{t,t-1} & \sigma_{t,t-2} & \sigma_{t,t-3} \\ \sigma_{t-1,t} & \sigma_{t-1,t-1} & \sigma_{t-1,t-2} & \sigma_{t-1,t-3} \\ \sigma_{t-2,t} & \sigma_{t-2,t-1} & \sigma_{t-2,t-2} & \sigma_{t-2,t-3} \\ \sigma_{t-3,t} & \sigma_{t-3,t-1} & \sigma_{t-3,t-2} & \sigma_{t-3,t-3} \end{bmatrix} \tag{26}$$

For this three lag example $P_{t|T}$ is the upper left[4] 3×3 submatrix of $P_{t|T}^{WE}$ in equation (26); $C_{t|T}$ refers to the upper right 1×3 submatrix of $P_{t|T}^{WE}$, and $P_{t|T}^{x}$ refers to element $\sigma_{t,t}$. This gives all the results needed to estimate the moment matrices above. The EM algorithm proceeds by alternately estimating the unobserved factors via the Kalman filter and smoother and estimating the parameters of the model as outlined above until the log likelihood function converges.

4 Practical Issues

4.1 Identification

An important practical issue for either maximum likelihood or Bayesian estimation of factor models is identifying the model. The issue arises as the factors x_t are never observed. Thus the model

$$\begin{aligned} y_t &= Hx_t + \varepsilon_t \\ x_t &= Bx_{t-1} + e_t \end{aligned} \tag{27}$$

is equivalent to the model

$$\begin{aligned} y_t &= \underbrace{H\theta^{-1}}_{\mathsf{H}} \underbrace{\theta x_t}_{\mathsf{x_t}} + \varepsilon_t \\ \underbrace{\theta x_t}_{\mathsf{x_t}} &= \underbrace{\theta B \theta^{-1}}_{\mathsf{B}} \underbrace{\theta x_{t-1}}_{\mathsf{x_{t-1}}} + \underbrace{\theta e_t}_{\mathsf{e_t}} \end{aligned}$$

[4]Of course each element $\sigma_{i,j}$ is itself a matrix so in fact $P_{t|T}$ will have dimension $3m \times 3m$.

Indeed, factor estimates and the likelihood function for the model in (27) would be identical to those for the model

$$
\begin{aligned}
y_t &= \mathsf{H}\mathsf{x}_\mathsf{t} + \varepsilon_t \\
\mathsf{x}_\mathsf{t} &= \mathsf{B}\mathsf{x}_{\mathsf{t}-1} + \mathsf{e}_\mathsf{t}
\end{aligned} \tag{28}
$$

Identification may be important for interpreting our model.[5] It is, for maximum likelihood and Bayesian estimation, also important for estimation.[6] Identification requirements for maximum likelihood estimation are less stringent than for Bayesian estimation. Bayesian estimation by simulation relies on the assumption we are drawing factors and parameters from the same distribution at each iteration. On the other hand, because our convergence criteria for maximum likelihood is in the likelihood function, parameter estimates can drift between the models in (27) and (28) as the likelihood function for the two will be identical. What is important is that we scale the model at each iteration of the algorithm; otherwise our parameter estimates may become arbitrarily large or small thereby breaking the algorithm. There is no one way to scale or identify our model; for a discussion of several identification possibilities see Stock and Watson (2016). To scale the model we could enforce $H'H = I$ as with principal components or even $diag(H'H) = I$. Perhaps the simplest identification technique, however, it set the top $m \times m$ submatrix of H, where m is the number of factors, to be an identity matrix (this is called "naming factors" identification). In this case our model will not just be scaled, but be fully identified (thus we could and will use it for Baysian estimation as well). This strategy has the added advantage of giving our model a simple interpretation: this first factor describes filtered movements of the first variable in common with the entire data set, the second factor describes filtered movements of the second variable in common with the entire data set, and so on.

5 Bayesian Estimation by Simulation

Bayesian estimation by simulation is a third approach to estimating factor models which for many applications will be the best option. There are of course theoretical reasons why one might prefer Bayesian statistics generally — humans typically never encounter a situation in life in which they have no prior beliefs, even if these beliefs are strongly biased. Specifically to our purpose of estimating factor models,

[5] See Stock and Watson (2016) for a more comprehensive discussion of interpretation and identification issues.

[6] For principal component estimation identification is dealt with by the normalization for H in equation (19).

using prior beliefs may have practical applications. For example, we may believe unobserved factors to transition smoothly from one period to the next despite noisy observations. We could incorporate this belief into our model by biasing the parameters of the transition equation B towards zero and/or increasing the value of ν_0, the degrees of freedom in our inverse-Wishart prior for the variance of shocks in the transition equation.

6 Sampling

Our results for the Kalman filter, multivariate normal posteriors, and the Durbin Koopan disturbance smoother provide the tools we will use to estimate our model. Estimation begins with an initial guess for parameters. Given this initial guess, we would like to draw factors from our model[7]

$$y_t = Hx_t + \varepsilon_t \tag{29}$$

$$x_t = Bx_{t-1} + e_t \tag{30}$$

where

$$\begin{bmatrix} e_t \\ \varepsilon_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \right)$$

One possible approach is to

1. Use the standard Kalman filter to obtain an estimate of $x_{T|T}$ and $P_{T|T}$ describing our posterior distribution for factors in the final period.
2. Draw $\tilde{x}_T \sim \mathcal{N}\big(x_{T|T}, P_{T|T}\big)$

[7] I use the case of one lag in the transition equation for explanatory purposes here as the notation is simple. For more lags our model becomes

$$y_t = \tilde{H} z_t + \varepsilon_t$$

$$z_t = A z_{t-1} + e_t$$

where typically we will have $\tilde{H} = \begin{bmatrix} H & 0 & 0 & \dots \end{bmatrix}$, $z_t = \begin{bmatrix} x_t & x_{t-1} & \dots & x_{t-p+1} \end{bmatrix}$ where p is the number of lags, and A is the companion form of B in the more familiar vector autoregression $x_t = Bz_t + e_t$. Note also that I am somewhat lose with the definition of e_t as when we have more than one lag this vector includes zeros as "shocks" to lagged variables in z_t to make dimensions agree.

3. Iterate backwards one period using the standard Kalman smoother. That is, calculate

$$\begin{aligned} x_{T-1|T} &= x_{T-1|T-1} + g_{T-1}(\tilde{x}_T - x_{T|T-1}) \\ P_{T-1|T} &= P_{T-1|T-1} - g_{T-1}(P_{T|T-1} - P_{T|T})g'_{T-1} \end{aligned}$$

4. Draw $\tilde{x}_{T-1} \sim \mathcal{N}\big(x_{T-1|T}, P_{T-1|T}\big)$

5. Repeat these iterations back to the initial period to generate draws of $\tilde{x}_t$ for every period.

This approach is simple. However, as

$$g_t = P_{t|t}A'P^{-1}_{t+1|t}$$

we must invert $P_{t+1|t}$ at each step of the iteration. For a steady state model, that is, one in which the variance of factors does not change, this is not a big drawback as we can calculate g once (it will be the same in every period for a steady state model) and proceed. If the factor variance is not constant (this will always be the case if y_t has missing observations), calculating $P^{-1}_{t+1|t}$ becomes computationally burdensome, particularly in models with many lags as P has dimension $m \times p$ where m is the number of factors and p the number of lags. For this more general case, Durbin and Koopman (2012) propose a much more computationally efficient method for drawing factors. Their key result for normally distributed data, presented earlier in Durbin and Koopman (2001), is a method for drawing from the conditional distribution $f(x|y)$ when doing so directly is burdensome but drawing from the joint normal distribution $f(x, y)$ and calculating the expected values $E(x|y)$ is computationally more efficient. To borrow their notation, we can draw x^+, y^+ from $f(x, y)$, calculate $\hat{x} = E(x|y)$ (the filtered and smoothed factors from our true observations y), and calculate $\hat{x}^+ = E(x^+|y^+)$ (the filtered and smoothed observations from our simulated observations y^+). Then the draw from our desired distribution $f(x|y)$ is $\tilde{x} = \hat{x} + x^+ - \hat{x}^+$. The values $\hat{x}$ are the posterior means for the factors conditional on the data y, and $x^+ - \hat{x}^+$ are the simulated zero mean shocks. For our purposes the authors point out that this approach can be made even more efficient as follows:

Algorithm 1

1. Begin with a draw for x_0, ε_t at every period t, and e_t at every period t and iterate equations (29) and (30) forward to obtain a draw for factors x_t^+ and a draw for observations y_t^+.

2. Obtain $y_t^* = y_t - y_t^+$.

3. Obtain $\hat{x}_t^*$ by filtering and smoothing y_t^* using the disturbance smoother described in section 1.

4. We can then calculate our simulated factors as

$$\tilde{x}_t = \hat{x}_t^* + x_t^+$$

This excellent algorithm provides a quick, efficient, and easy to code method for sampling factors. Once we have a draw for factors, we can then draw parameters based on these simulated factors. Because we will be using an exact factor model, that is, we will treat the covariance matrix for shocks to observations R as diagonal, we can estimate each diagonal element of R and each row of H using a normal-inverse gamma conjugate prior.

Because our draw for R_{jj}, the j^{th} diagonal of the covariance matrix for shocks to the observation equation and thus the variance of shocks to the j^{th} observed series, is conditional on the observations and factors only, while our draw for H_j, the j^{th} row of H, is conditional on R_{jj}, we will begin by drawing variances. Specifically, for series j we draw R_{jj} from

$$f(R_{jj}|\tilde{X}, y) \sim \mathcal{IG}\Big((y - X\beta_T)'(y - X\beta_T) + (\beta_T - h_0)'\Lambda_0(\beta_T - h_0) + s_0, T + \nu_0\Big) \tag{31}$$

In the above $\beta_T = (\tilde{X}'X + \Lambda_0)^{-1}(\tilde{X}'y_j + \Lambda_0 h_0)$. s_0 is our prior scale parameter — our prior for the variance of shocks when multiplied by ν_0. ν_0 is our prior "degrees of freedom", that is, ν_0 determines how aggressively we shrink R_{jj}. Λ_0 is a diagonal matrix determining tightness on our prior for H_j, denoted h_0. Typically Λ_0 will be an identity matrix times a (scalar) tightness parameter, with larger values corresponding to a tighter prior, and h_0 will be zero for all rows of H, though we could use different priors for each row if desired. Once we have a draw for R_{jj}, we draw row j of H from

$$f(H_j|R_{jj}, \tilde{X}, y_j) \sim \mathcal{N}\Big(\Lambda_T^{-1}(\tilde{X}'y_j + \Lambda_0 h_0), R_{jj}\Lambda_T^{-1}\Big) \tag{32}$$

where $\Lambda_T = (\tilde{X}'X + \Lambda_0)$. Note that the posterior mean for H_j is just β_T as defined above.

Our draws for Q (the covariance of shocks to the observation equation) and B are nearly identical except that, because Q contains off-diagonal elements, we will use a normal-inverse Wishard conjugate prior. Beginning again with covariances, we draw Q from

(33)

$$f(Q|\tilde{X}, Y) \sim \mathcal{IW}\Big((Y - \tilde{X}B_T)'(Y - \tilde{X}B_T) + (B_T - B_0)'\Lambda_0(B_T - B_0) + V_0, \nu_0 + T\Big)$$

where $B_T = (\tilde{X}'\tilde{X} + \Lambda_0)^{-1}(X'Y + \Lambda_0 B_0)$. Again, Λ_0 determines the tightness on our prior for B, denoted B_0. Typically we will use a zero prior for B, though using a random walk prior is also popular. V_0 is our prior scale parameter (our prior for the covariance to shocks when multiplied by ν_0) and ν_0 determines how aggressively we shrink Q. In most programming languages, including the C++ and R code associated with this book, we will need to use the vectorized form of our posterior for B. Thus we draw $\beta = vec(B')$ from

$$f(\beta|Q, \tilde{X}, Y) \sim \mathcal{N}\Big(vec\big(\big[\Lambda_T^{-1}(\tilde{X}'Y + \Lambda B_0)\big]'\big), \Lambda_T^{-1} \otimes \Sigma\Big) \tag{34}$$

where $\Lambda_T = \tilde{X}'\tilde{X} + \Lambda_0$ and our posterior mean (in matrix format) is just B_T as defined above.

Once we have a draw for parameters we can again draw factors. We continue these iterations until our posterior distributions converge to a stationary distribution, and then draw a sufficient number of parameters to suit our needs, that is, estimating posterior means and variances of the parameters themselves. This whole process is re-stated in the following algorithm:

Algorithm 2

1. Begin with a guess for parameters.
2. Given parameters, sample factors following Algorithm 1.
3. Given factors obtained in step (2), draw:
   (a) R_{jj} from the distribution in (31) and H_j from the distribution in (32) for every series j.
   (b) Q from the distribution in (33) and B from the distribution in (34).

4. Repeat steps 2 and 3 until distributions converge to stationary distributions.

5. Once distributions converge repeat steps 2 and 3 storing draws for R, H, Q, and B at each iteration.

The question of when the distribution has converged is not straight forward — see Robert and Casella (2010) for a discussion of convergence criteria. Our typical approach will be two simply use trace plots to examine convergence visually.

6.1 Identification

Identifying factors ensures that we are in fact drawing from the same distributions in each step of Algorithm 2. Otherwise, our posterior means would be meaningless. In practice, sampling under-identified models will typically cause parameters or factors to explode, that is, become arbitrary large and thus break our iterations. The issue of identification is described in section 4.1. The simplest, though not the only, solution to identification of Bayesian dynamic factor models is what Stock and Watson (2011) call naming factors identification, that is, setting the top $m \times m$ submatrix of H (where m is the number of factors) to be the identify matrix. Employing this identification scheme in our sampling iterations requires that we create a temporary $m \times m$ H matrix for the first m series in Y, which I will call M. Based on our draw for factors $\tilde{X}$, draw the elements of R_{jj}, $j \in (1, m)$ and M for the first m series from distributions (31) and (32) respectively. To enforce that the top $m \times m$ submatrix of H is I_m we then use M to normalize our draw for the factors $\tilde{X}$. This turns out to be quite simple. Using the observation equation, our normalization is

$$y_{t,1:m} = \underbrace{MM^{-1}}_{I_m} \underbrace{M\tilde{x}_{t,old}}_{\tilde{x}_{t,new}} + \varepsilon_t$$

Thus for a model with one lag, our normalization is just[8]

$$\tilde{X}_{new} = \tilde{X}_{old} M$$

Because we have not yet estimated the transition equation (we do that after normalizing factors) or any other elements of H and R, this is all that is needed to

[8]We postmultiply by M here as X_{old} is the matrix of simulated factors with time indexed by rows

identify our model. For a model with p lags where

$$z_t = \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-p+1} \end{bmatrix}$$

This normalization becomes

$$\tilde{Z}_{new} = \tilde{Z}_{old}(I_p \otimes M) \tag{35}$$

7 An Example: Bayesian vs. ML Estimation for Simulated Data

As a first example I simulate ten series ($k = 10$) from a factor model with observation equation

$$y_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ .5 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0.5 & -1 \\ 0 & .5 & 1 \end{bmatrix} x_t + \varepsilon_t \tag{36}$$

and transition equation

$$x_t = \begin{bmatrix} 0.4 & 0 & 0 & 0.3 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0.3 & 0.2 & 0 & 0.2 & 0.1 & 0 & 0.1 & 0 \\ 0 & 0.2 & 0.4 & 0 & 0 & 0.2 & 0 & 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix} + e_t \tag{37}$$

where

$$\varepsilon_t \sim \mathcal{N}\big(0, I_{10}\big)$$

and

$$e_t \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix}\right)$$

	$T = 50$	$T = 100$	$T = 200$
Bayes	0.42	0.34	0.30
ML	0.44	0.36	0.33

Table 1: MSE for Bayesian vs. ML Estimation

Note that the first observed series y_t^1 is just the first factor x_t^1 and that this factor is independent of the other two factors.

My objective in this section will be to estimate the "true" data $y_t^{true} = y_t - \varepsilon_t$ or simply $y_t^{true} = Hx_t$ based on the observed series y_t. Note that this true series is never observed. In addition to the noise ε_t I drop observations from the first three series. Thus for these series I have five observations followed by five missing values, then five observations and so on. Simulations proceed by

1. Simulating data according to equations (29) and (30)
2. Dropping blocks of five observations from the first three series
3. Estimating the model by simulation as outlined in section 5 and by maximum likelihood as outlined in section 3
4. Calculating the mean squared error $1/T \sum(\hat{y}_t - y_t^{true})^2$ for all observed series and averaging over the 10 series for 1000 repetitions

Table 1 presents the results when I estimate a model with three factors and two lags. As is evident from the table Bayesian estimation yields slightly better results in terms of mean squared error. However, this comes at the cost of longer run time as we are simulating entire distributions, not simply estimating a few parameters. Note, however, that Bayesian estimation may be faster for models with large numbers of lags, such as mixed frequency models, as inverting the covariance matrix for predicted factors $P_{t+1|t}$ becomes increasingly burdensome as p gets large.

8 An Introduction to Mixed Frequency Models

Because state space models are so apt and handling missing data they are particularly well suited to mixed frequency data sets in which, for example, a quarterly variable will not be observed for two out of three months.

Suppose first that our model is in log levels and, as a concrete example, that frequencies are either monthly or quarterly. Denote y_t^q the log of a quarterly observation in month t and y_t^m the log of a monthly observation in month t. Then

$$e^{y_t^m} = e^{y_t^m} + e^{y_{t-1}^m} + e^{y_{t-2}^m} \tag{38}$$

The difficulty lies in the fact that equation (1) is linear in the log variables while equation (38) is not; to overcome this issue simply take a linear approximation of (38) yielding

$$y_t^m = \frac{1}{3}(y_t^m + y_{t-1}^m + y_{t-2}^m) \tag{39}$$

Plugging equation (1) into the above yields the linear state space structure:

$$y_t^q = \frac{1}{3}Hx_t + \frac{1}{3}Hx_{t-1} + \frac{1}{3}Hx_{t-2} + \varepsilon_t \tag{40}$$

Note that this requires that the model include at least three lags of factors, although one need not estimate coefficients on factors with more than one lag in the transition equation.

To put the model into log differences we begin with equation (39) and note that what we observe, Δy_t^q, is

$$\begin{aligned} y_t^q - y_{t-3}^q &= \frac{1}{3}(y_t^m - y_{t-3}^m) + \frac{1}{3}(y_{t-1}^m - y_{t-4}^m) + \frac{1}{3}(y_{t-2}^m - y_{t-5}^m) \\ &= \frac{1}{3}\Delta y_t^w + \frac{2}{3}\Delta y_{t-1}^w + \Delta y_{t-2}^w + \frac{2}{3}\Delta y_{t-3}^w + \frac{1}{3}\Delta y_{t-4}^w \end{aligned} \tag{41}$$

This is the result presented in Mariano and Murasawa (2003). Unlike the levels case, we now need to include at least four lags of the factors.

Estimates of H and R now must take into account the structure of the data in (39) for level data or (41) for differenced data. A simple way to proceed is to define a new helper matrix J_q for low frequency data. Maintaining the monthly/quarterly

structure and the three lag model by way of example, the helper matrix for quarterly data, assuming we are dealing with data in levels, will be

$$J_q = \begin{bmatrix} \frac{1}{3}I_m & \frac{1}{3}I_m & \frac{1}{3}I_m \end{bmatrix} \tag{42}$$

or, for quarterly variables in first differences

$$J_q = \begin{bmatrix} \frac{1}{3}I_m & \frac{2}{3}I_m & I_m & \frac{2}{3}I_m & \frac{1}{3}I_m \end{bmatrix} \tag{43}$$

The difficulty with a more general structure, monthly/daily data for example, is two fold. First, the number of high frequency periods (days) in a low frequency period (months) gets large. For differenced data our helper matrix for monthly variables, assuming 31 days in the month, becomes the $m \times 61$ matrix

$$J_q = \begin{bmatrix} \frac{1}{31}I_m & \frac{2}{31}I_m & \dots & \frac{2}{31}I_m & \frac{1}{31}I_m \end{bmatrix} \tag{44}$$

Thus the vector of factors z_t must include at least 61 lags. Second, the number of days in the month is not constant. Nor is the number of days in February constant from one year to the next. Thus our helper matrix J_q must change depending on the number of days in the current month. Though conceptually simple, these facts can become computationally difficult. To illustrate the point, our daily/monthly data requires 61 lags in z_t. If our model includes a modest three factors, than z_t has 183 elements, a huge number for a literature in which there or typically only a few latent factors.

The structure of the transition equation is an additional consideration when the number of high frequency periods (days) in a low frequency period (months) is large. The frequency for factors is the highest frequency in the model — in our daily/monthly example factors are thus daily. However, assuming we are interested in nowcasting or forecasting a low frequency (monthly) variable we would like to include, in our monthly-daily example, at least a month of lags in the transition equation. This raises the problem of over-parameterization: with three factors and 30 lags we will be estimating 90 parameters in the transition equation excluding the covariances in Q. Our solution is to again introduce a helper matrix which, for the transition equation, we call J_B. We then specify the number of lags at each frequency in the model. For a daily/weekly/monthly model we we then have p_d daily lags, p_w weekly lags, and p_m monthly lags (in the transition equation we use the term month to refer to 30 days regardless of the actual number of days in the month). For a model with two factors and $p_d = 1$, $p_w = 1$, and $p_m = 1$ our transition equation equation can be described by

$$B = \begin{bmatrix} b_{11}^d & b_{12}^d & b_{11}^w & b_{12}^w & b_{11}^m & b_{12}^m \\ b_{21}^d & b_{22}^d & b_{21}^w & b_{22}^w & b_{21}^m & b_{22}^m \end{bmatrix}$$

$$J_B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & \dots \\ 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \dots \\ \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & \dots \\ 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \dots \end{bmatrix}$$

where b_{21}^w refers to the multiplier of factor 1 on factor 2 at a weekly frequency, and finally

$$X_{t+1} = BJ_BZ_t + e_t$$

Equipped with the (sparse) helper matrices described above our model fits into the smoothing and filtering algorithms described in section 1. The only complication is that $\tilde{H}_t = HJ_t$ changes from one month to the next as the number of days in a month changes. This does not present any theoretical difficulties but requires careful coding when writing estimation routines.

9 Further Reading

Durbin and Koopman (2012) provide the definitive reference for practical implementation of dynamic factor models. Watson and Engle (1983) is a very readable paper and remains one of the best resources for estimating DFMs by maximum likelihood (section 3 draws directly from this paper). Other papers that may be useful to the beginning practitioner include Stock and Watson (2011), Mariano and Murasawa (2003), and Giannone et al. (2008).

References

Bańbura, M., Giannone, D., and Lenza, M., 2015. Conditional forecasts and scenario analysis with vector autoregressions for large cross-sections. *International Journal of Forecasting*, 31(3):739–756.

Doz, C., Giannone, D., and 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. and Koopman, S., April 2001. An efficient and simple simulation smoother for state space time series analysis. Computing in Economics and Finance 2001 52, Society for Computational Economics.

Durbin, J. and Koopman, S. J., 2012. *Time Series Analysis by State Space Methods*. Oxford University Press.

Giannone, D., Reichlin, L., and Sala, L., 2005. Monetary Policy in Real Time. In Gertler, M. and Rogoff, K., editors, *NBER Macroeconomics Annual 2004*, volume 19. MIT Press.

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

Kalman, R., 01 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45.

Mariano, R. S. and Murasawa, Y., 2003. A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics*, 18(4): 427–443.

McCracken, M. W. and McGillicuddy, J. T., March 2019. An empirical investigation of direct and iterated multistep conditional forecasts. *Journal of Applied Econometrics*, 34(2):181–204.

Robert, C. P. and Casella, G., 2010. *Monte Carlo Statistical Methods*. Springer Publishing Company, Incorporated.

Sims, C. A., January 1980. Macroeconomics and Reality. *Econometrica*, 48(1): 1–48.

Stock, J. H. and Watson, M. W., 2011. Dynamic Factor Models. In Clements, M. P. and Hendry, D. F., editors, *The Oxford Handbook of Economic Forecasting*. Oxford, Oxford Handbooks.

Stock, J. H. and Watson, M. W., 2016. Factor Models and Structural Vector Autoregressions in Macroeconomics. In Taylor, J. B. and Uhlig, H., editors, *Handbook of Macroeconomics*, volume 2A, pages 415–525. North-Holland.

Watson, M. W. and Engle, R. F., 1983. Alternative Algorithms for the Estimation of Dynamic Factor, Mimic and Varying Coefficient Regression Models. *Journal of Econometrics*, 23(3):385–400.