Sparse Bayesian Time-Varying Covariance Estimation in Many Dimensions

Gregor Kastner

Research Report Series
Report 129, September 2016

Institute for Statistics and Mathematics
http://statmath.wu.ac.at/
Sparse Bayesian Time-Varying Covariance Estimation in Many Dimensions

Gregor Kastner*
WU Vienna University of Economics and Business, Austria

First Version: July 16, 2016
This Version: September 18, 2016

Abstract

Dynamic covariance estimation for multivariate time series suffers from the curse of dimensionality. This renders parsimonious estimation methods essential for conducting reliable statistical inference. In this paper, the issue is addressed by modeling the underlying co-volatility dynamics of a time series vector through a lower dimensional collection of latent time-varying stochastic factors. Furthermore, we apply a Normal-Gamma prior to the elements of the factor loadings matrix. This hierarchical shrinkage prior effectively pulls the factor loadings of unimportant factors towards zero, thereby increasing parsimony even more. We apply the model to simulated data as well as daily log-returns of 300 S&P 500 stocks and demonstrate the effectiveness of the shrinkage prior to obtain sparse loadings matrices and more precise correlation estimates. Moreover, we investigate predictive performance and discuss different choices for the number of latent factors. Additionally to being a stand-alone tool, the algorithm is designed to act as a “plug and play” extension for other MCMC samplers; it is implemented in the R package factorstochvol.

JEL classification: C32; C51; C58

Keywords: dynamic conditional correlation, factor stochastic volatility, curse of dimensionality, shrinkage, predictive distribution

*Department of Finance, Accounting and Statistics, WU Vienna University of Economics and Business, Welthandelsplatz 1/D4/4, 1020 Vienna, Austria, +43 1 31336-5593, gregor.kastner@wu.ac.at
1 Introduction

The joint analysis of hundreds or thousands of time series exhibiting potentially time-varying variance-covariance structure has been on numerous research agendas for well over a decade. Suggested methods include Lopes et al. (2012) who treat the Cholesky-decomposed covariance matrix within the framework of Bayesian time-varying parameter models, Engle and Kelly (2012) who propose an estimator assuming that pairwise correlations are equal at every point in time, Pakel et al. (2014) who consider composite likelihood estimation, Gruber and West (2016) who use a decoupling-recoupling strategy to parallelize estimation (executed on graphical processors), Oh and Patton (forthcoming) who chose a copula-based approach to link separately estimated univariate models, among others. All these approaches have one thing in common; the objective to find a good balance between the necessary flexibility on the one hand and parameter parsimony on the other hand.

To tackle this issue, we use a factor stochastic volatility model. Key early references include Harvey et al. (1994), Pitt and Shephard (1999), and Aguilar and West (2000) which have later been picked up and extended by e.g. Philipov and Glickman (2006), Chib et al. (2006), Han (2006), Lopes and Carvalho (2007), Nakajima and West (2013), Zhou et al. (2014), Kastner et al. (2016), and Ishihara and Omori (forthcoming). While reducing the dimensionality of the problem at hand, models with many factors are still rather rich in parameters. Thus, we further propose to shrink unimportant elements of the factor loadings matrix to zero in an automatic way. This technique is inspired from high-dimensional regression problems where the number of parameters frequently exceeds the size of the data, an issue appearing in many applications. In particular, we adopt the approach brought forward by Griffin and Brown (2010) who propose to use a special prior structure – the Normal-Gamma prior – on the regression parameters (in our case the factor loadings matrix). This shrinkage prior is a generalization of the Bayesian Lasso (Park and Casella 2008) and has recently received attention in the econometrics literature, see e.g. Bitto and Frühwirth-Schnatter (2015).

Another major issue for such high-dimensional problems is the computational burden that goes along with statistical inference, in particular when joint modeling is attempted instead of multi-step approaches or rolling-window-type estimates. We propose to use a
Gibbs-type sampler which allows to jointly take into account both parameter as well as sampling uncertainty in a finite-sample setup through fully Bayesian inference, thereby allowing for inherent uncertainty quantification without relying on asymptotic approximations. Additionally, this approach allows for fully probabilistic in- and out-of-sample density predictions.

While not being the focus of this work, it is easy to extend the algorithm proposed by exploiting the modular nature of Markov chain Monte Carlo methods. Potential directions include more complex dynamics, or a combination with models for the mean structure of multivariate time series such as vector autoregressive (VAR) models.

2 Model Specification

Consider an \( m \)-variate zero-mean return vector \( \mathbf{y}_t = (y_{1t}, \ldots, y_{mt})' \) for time \( t = 1, \ldots, T \) whose conditional distribution is Gaussian, i.e.

\[
\mathbf{y}_t | \Sigma_t \sim N_m(\mathbf{0}, \Sigma_t).
\]

To reduce dimensionality, factor SV models utilize a decomposition of the \( m \times m \) covariance matrix \( \Sigma_t \) with \( m(m + 1)/2 \) free elements into a factor loadings matrix \( \Lambda \) of size \( m \times r \), an \( r \)-dimensional diagonal matrix \( V_t \) and an \( m \)-dimensional diagonal matrix \( U_t \) in the following fashion:

\[
\Sigma_t = \Lambda V_t \Lambda' + U_t.
\]

This reduces the number of free elements to \( mr + m + r \). Because \( r \) is typically chosen to be much smaller than \( m \), this specification constrains the parameter space substantially, thereby inducing parameter parsimony. For the paper at hand, \( \Lambda \) is considered to be time invariant whereas the elements of both \( V_t \) and \( U_t \) are allowed to evolve over time through parametric stochastic volatility models, i.e. \( V_t = \text{diag}(\exp(h_{1t}), \ldots, \exp(h_{mt})) \) and \( U_t = \text{diag}(\exp(h_{m+1,t}), \ldots, \exp(h_{m+r,t})) \) with

\[
\begin{align*}
    h_{it} &\sim N(\mu_i + \phi_i(h_{i,t-1} - \mu_i), \sigma_i^2), \quad i = 1, \ldots, m, \\
    h_{m+j,t} &\sim N(\phi_{m+j} h_{m+j,t-1}, \sigma_{m+j}^2), \quad j = 1, \ldots, r.
\end{align*}
\]
More specifically, $V_t$ contains the variances of underlying orthogonal factors $f_t \sim \mathcal{N}(0, V_t)$ that govern the contemporaneous dependence; $U_t$ describes the idiosyncratic (series-specific) variances.

2.1 Factor SV Model

This setup is commonly written in the following hierarchical form (e.g. Chib et al. 2006):

$$y_t | \Lambda, f_t, U_t \sim \mathcal{N}_m(\Lambda f_t, U_t), \quad f_t | V_t \sim \mathcal{N}_r(0, V_t),$$

pairwise independently for all points in time. To make further exposition clearer, let $y = (y_1 \cdots y_T)$ denote the $m \times T$ matrix of all observations, $f = (f_1 \cdots f_T)$ the $r \times T$ matrix of all latent factors and $h = (h'_1 \cdots h'_{m+r})'$ the $(T+1) \times (m+r)$ matrix of all $m+r$ log-variance processes $h_i = (h_{i0}, h_{i1}, \ldots, h_{iT})$, $i = 1, \ldots, m+r$. The vector $\theta_i = (\mu_i, \phi_i, \sigma_i)'$ is referred to as the vector of parameters where $\mu_i$ is the level, $\phi_i$ the persistence, and $\sigma_i^2$ the innovation variance of $h_i$. To denote specific rows and columns of matrices, we use the “dot” notation, i.e. $X_i \cdot$ refers to the $i$th row and $X \cdot j$ to the $j$th column of $X$.

The proportion of variance explained through the common factors for each component series,

$$C_{it} = 1 - \frac{U_{ii,t}}{\Sigma_{ii,t}}, \quad i = 1, \ldots, m,$$

are referred to as the communalities. Here, $U_{ii,t}$ and $\Sigma_{ii,t}$ denote the $i$th diagonal element of $U_t$ and $\Sigma_t$, respectively. Because by construction $0 \leq U_{ii,t} \leq \Sigma_{ii,t}$, the communality for each component series and for all points in time lies between zero and one. The joint (overall) communality $C_t$ is simply defined as the arithmetic mean over all series;

$$C_t = m^{-1} \sum_{i=1}^m C_{it}.$$

2.2 Prior Distributions

The usual prior for each element of the factor loadings matrix is a zero-mean normal distribution, i.e. $\Lambda_{ij} \sim \mathcal{N}(0, \tau_{ij}^2)$ independently for each $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, r\}$, where $\tau_{ij}^2 \equiv \tau^2$ is a constant specified a priori. A natural relaxation of the assumption
of static prior variance is to model this variance as a random variable and to specify a second level of hierarchy by placing a hyperprior on \( \tau^2_{ij} \). Following Griffin and Brown (2010) we do so by modeling \( \tau^2_{ij} \) with a Gamma distribution,

\[
\Lambda_{ij} | \tau^2_{ij} \sim \mathcal{N}(0, \tau^2_{ij}) , \quad \tau^2_{ij} | \lambda^2_i \sim \mathcal{G}(a_i, a_i \lambda^2_i / 2). \tag{3}
\]

Integrating out \( \tau^2_{ij} \) yields a density for \( \Lambda_{ij} | \lambda^2_i \) of the form

\[
p(\Lambda_{ij} | \lambda^2_i) \propto |\Lambda_{ij}|^{a_i-1/2}K_{a_i-1/2}(\sqrt{a_i} \lambda_i | \Lambda_{ij}|),
\]

where \( K \) is the modified Bessel function of the third kind. This implies that the conditional variance of \( \Lambda_{ij} | \lambda^2_i \) is \( 2/\lambda^2_i \); the excess kurtosis of \( \Lambda_{ij} \) is \( 3/a_i \). For the remainder of this paper, we let \( \lambda^2_i \sim \mathcal{G}(c_i, d_i) \) where \( c_i \) and \( d_i \) are fixed hyperhyperparameters. Instead of placing yet another hyperprior on \( a_i \), we rather fix \( a_i \) and treat it as a structural parameter. Choosing \( a_i \) small enforces strong shrinkage towards zero, while choosing \( a_i \) large imposes little shrinkage. For more elaborate discussions on Bayesian shrinkage in general and the effect of \( a_i \) specifically, see Griffin and Brown (2010) and Polson and Scott (2011). Note that the Bayesian Lasso prior (Park and Casella 2008) arises as a special case when \( a_i = 1 \).

One can see prior (3) as row-wise shrinkage with element-wise adaption in the sense that all variances in row \( i \) can be thought of as “random effects” from the same underlying distribution. In other words, each series has high and a priori independent mass not to load on any factors and thus can be thought of as series-specific shrinkage.

Analogously, it turns out to be fruitful to also consider column-wise shrinkage with element-wise adaption, i.e.

\[
\Lambda_{ij} | \tau^2_{ij} \sim \mathcal{N}(0, \tau^2_{ij}) , \quad \tau^2_{ij} | \lambda^2_j \sim \mathcal{G}(a_j, a_j \lambda^2_j / 2), \tag{4}
\]

with the corresponding prior \( \lambda^2_j \sim \mathcal{G}(c_j, d_j) \). This means that each factor has high and a priori independent mass not to be loaded on by any series and thus can be thought of as factor-specific shrinkage.

For the \( m \) idiosyncratic volatilities, the initial state \( h_{i0} \) is distributed according to the stationary distribution of the AR(1) process (1), i.e. \( h_{i0} | \mu_i, \phi_i, \sigma_i \sim \mathcal{N}(\mu_i, \sigma_i^2 / (1 - \phi_i^2)) \),
For the $r$ latent volatilities we use a slightly modified priors which do not depend on the persistence, i.e. $h_{m+j,0| \sigma_{m+j}} \sim \mathcal{N}(0, B_{m+j} \sigma^2_{m+j})$, $j = 1, \ldots, r$. If $B_{m+j}$ is chosen small, these priors turn out to induce more stable marginal posteriors for extremely persistent latent factor volatilities where $\phi_{m+j}$ is close to 1.

Priors for the univariate SV parameters are set as follows: $p(\mu_i, \phi_i, \sigma_i) = p(\mu_i)p(\phi_i)p(\sigma_i)$, where the level $\mu_i \in \mathbb{R}$ is equipped with the usual normal prior $\mu_i \sim \mathcal{N}(b_\mu, B_\mu)$, the persistence parameter $\phi_i \in (-1, 1)$ is implied by $(\phi_i + 1)/2 \sim \mathcal{B}(a_0, b_0)$ as in Kim et al. (1998), and the volatility of volatility parameter $\sigma_i \in \mathbb{R}^+$ is chosen according to $\sigma^2_i \sim B_\sigma \chi^2_1 \equiv G(1/2, 1/(2B_\sigma))$ as in Kastner and Frühwirth-Schnatter (2014).

### 2.3 A Remark about Identification

Identifying loadings for latent factor models is a long-standing issue that goes back to at least Anderson and Rubin (1956) who discuss identification of factor loadings. Even though this problem is alleviated somewhat in the case when factors are allowed to exhibit conditional heteroskedasticity (Sentana and Fiorentini 2001; Rigobon 2003), most authors have chosen an upper triangular constraint of the loadings matrix with unit diagonal elements, thereby introducing dependence on the ordering of the data. However, when estimation of the actual factor loadings is not the primary concern (but rather a means to estimate and predict the covariance structure), this issue is less striking because a unique identification of the loadings matrix is not necessary.\(^1\) This allows to leave the factor loadings matrix to be completely unrestricted, often alleviating MCMC convergence.

### 3 Statistical Inference

There are a number of methods to estimate factor stochastic volatility models such as quasi-maximum likelihood (e.g. Harvey et al. 1994), simulated maximum likelihood (e.g. Liesenfeld and Richard 2006; Jungbacker and Koopman 2006), and Bayesian MCMC simulation (e.g. Pitt and Shephard 1999; Aguilar and West 2000; Chib et al. 2006; Han 2006). For high dimensional problems of this kind, Bayesian MCMC estimation is a very

\(^1\)Note that the conditional covariance matrix $\Sigma_t = \Lambda \Lambda' + U_t$ only involves a rotation-invariant transformation of $\Lambda$. 
efficient estimation method because it allows to draw from the high dimensional joint posterior by drawing from lower dimensional conditional posteriors.

3.1 MCMC Estimation

One substantial advantage of MCMC methods over other ways to learn about the posterior distribution is that it constitutes a modular approach due to the conditional nature of the sampling steps. For the model at hand, this means that conditionally on the matrix of variances $\boldsymbol{\tau} = (\tau_{ij})_{1 \leq i \leq m; 1 \leq j \leq r}$, the sampling steps of Kastner et al. (2016) can be executed with minor modifications. For sampling $\boldsymbol{\tau}$, we follow Griffin and Brown (2010).

**Algorithm 1** (MCMC sampling steps for the Factor SV model).

1. For factors and idiosyncratic variances, obtain $m$ conditionally independent draws of the idiosyncratic log-volatilities from $h_i\mid y_i, \Lambda_i, f, \mu_i, \phi_i, \sigma_i$ and their parameters from $\mu_i, \phi_i, \sigma_i\mid y_i, \Lambda_i, f, h_i$ for $i = 1, \ldots, m$. Similarly, perform $r$ updates for the factor log-volatilities from $h_{m+j}\mid f_{m+j}, \phi_{m+j}, \sigma_{m+j}$ and their parameters from $\phi_{m+j}, \sigma_{m+j}\mid f_{m+j}, h_{m+j}$ for $j = 1, \ldots, r$. This amounts to $m + r$ univariate SV-updates.\(^2\)

2a. Row-wise shrinkage only: For $i = 1, \ldots, m$, sample from

$$\lambda_i^2 \mid \tau_i \sim \mathcal{G} \left( c_i + a_i \tilde{r}, d_i + \frac{a_i}{2} \sum_{j=1}^{\tilde{r}} \tau_{ij}^2 \right),$$

where $\tilde{r} = \min(i, r)$ if the loadings matrix is restricted to have zeros above the diagonal and $\tilde{r} = r$ in the case of an unrestricted loadings matrix. For $i = 1, \ldots, m$ and $j = 1, \ldots, \tilde{r}$, draw from $\tau_{ij}^2 \mid \lambda_i, \Lambda_{ij} \sim \text{GIG}(a_i - \frac{1}{2}, a_i \lambda_i^2, \Lambda_{ij}^2)$.\(^3\)

\(^2\)We use the C-level interface to `update` in the R (R Core Team 2016) package `stochvol` (Kastner 2016) which provides a single sweep of the draws implementing an efficient AWOL (Kastner and Frühwirth-Schnatter 2014) sampler with interweaving (Yu and Meng 2011).

\(^3\)The Generalized Inverse Gaussian distribution $\text{GIG}(m, k, l)$ has a density proportional to $x^{m-1} \exp \left\{ -\frac{1}{2}(kx + l/x) \right\}$. Sampling can easily be done using the R package `GIGrvg` (Leydold and Hörmann 2015) which contains a very stable implementation of the algorithm described in Hörmann and Leydold (2013) particularly designed for the varying parameter case. Be aware that `GIGrvg` uses a slightly different parameterization; $\text{GIG}(m, k, l) \equiv \text{GIG}_{\text{GIGrvg}}(\lambda, \chi, \psi)$ where $\lambda = m$, $\chi = l$, $\psi = k$. 
2b. Column-wise shrinkage only: For \( j = 1, \ldots, r \), sample from

\[
\lambda_j^2 | \tau_j \sim \mathcal{G} \left( c_j + a_j (m - \tilde{j} + 1), d_j + \frac{a_j}{2} \sum_{i=\tilde{j}}^{m} \tau_{ij}^2 \right),
\]

where \( \tilde{j} = j \) if the loadings matrix is restricted to have zeros above the diagonal and \( \tilde{j} = 1 \) otherwise. For \( j = 1, \ldots, r \) and \( i = \tilde{j}, \ldots, r \), draw from \( \tau_{ij}^2 | \lambda_j, \Lambda_{ij} \sim \text{GIG}(a_j - \frac{1}{2}, a_j \lambda_j^2, \Lambda_{ij}^2) \).^{3}

3. Letting \( \Psi_i = \text{diag}(\tau_{i1}^{-2}, \tau_{i2}^{-2}, \ldots, \tau_{i\tilde{r}}^{-2}) \), draw \( \Lambda_i' | f, y_i, h_i, \Psi_i, \sim \mathcal{N}_\tilde{r}(b_{iT}, B_{iT}) \) with \( B_{iT} = (X_i'X_i + \Psi_i)^{-1} \) and \( b_{iT} = B_{iT}X_i'y_i \). Hereby, \( \tilde{y}_i = (y_{i1}e^{-h_{i1}/2}, \ldots, y_{iT}e^{-h_{iT}/2})' \) denotes the \( i \)th normalized observation vector and

\[
X_i = \begin{bmatrix}
    f_{11}e^{-h_{i1}/2} & \cdots & f_{\tilde{r}1}e^{-h_{i1}/2} \\
    \vdots & \ddots & \vdots \\
    f_{1T}e^{-h_{iT}/2} & \cdots & f_{\tilde{r}T}e^{-h_{iT}/2}
\end{bmatrix}
\]

is the \( T \times \tilde{r} \) design matrix. This constitutes a standard Bayesian regression update.

3*. When inference on the factor loadings matrix is sought, optionally redraw \( \Lambda \) using deep interweaving (Kastner et al. 2016) to speed up mixing. This step is of less importance if one is interested in the (predictive) covariance matrix only.

4. Sampling the factors conditionally on the loadings is again a standard Bayesian regression update and proceeds exactly as in Kastner et al. (2016).

3.2 Computational Aspects

High-dimensional models, in particular models with many latent variables, pose a non-negligible computational challenge to those aiming for efficient MCMC implementations. This is mainly due to the intrinsically iterative nature of MCMC in the sense that posterior draws are generated conditionally on older draws. Thus, they cannot be parallelized straightforwardly and call for compiled and optimized programming languages to avoid the cost of code interpretation at every iteration. Moreover, memory access needs to be optimized, as large amounts of latent variable draws must be stored either temporarily (if required only for the next conditional draws) or more permanently (if required for
Table 1: Asymptotic computational cost of the sampling steps with respect to the time series length $T$, the number of time series $m$ and the number of factors $r$.

direct posterior inference). In this paper, we tackle the computational burden by using high-performance C and C++ code, interfaced to R via 
RcppArmadillo (Eddelbuettel and Sanderson 2014). Additionally to providing an interface between R and C++, 
RcppArmadillo also accommodates fast linear algebra routines by means of the Armadillo library (Sanderson 2010).

The approximate computational burden as a function of the time series length, the dimensionality of the data as well as the number of factors is summarized in Table 1. All sampling steps are of linear complexity with the exception of sampling the factor loadings as well as the factors which are of quadratic complexity in $r$ as they both involve the inversion of an $r \times r$ matrix, implying the overall asymptotic cost of $O(Tmr^2)$. The algorithm is linear in both $m$ and $T$ and because the number of factors $r$ is typically small, the quadratic asymptotic complexity in $r$ is of little relevance. In fact, measurements of computing time indicate almost linear growth in $r$ for all practically relevant values for $r$.

4 Generally speaking, most of the computation time is spent sampling the latent volatilities, followed by the factor and factor loadings matrix. Only in models with very many factors, sampling factors and loadings becomes a substantial burden. The computational cost of shrinkage with the Normal-Gamma prior is practically negligible for all model sizes.

To a certain extent, computation can further be sped up by computing the individual steps of Algorithm 1 in parallel. In practice, however, doing so is only useful in shared memory environments (e.g. through multithreading/multiprocessing) as the increased communication overhead in distributed memory environments easily outweighs the speed gains.

$^4$On a single CPU core, estimating a 15-factor model takes about twice as long as estimating a 1-factor model; going to 20 factors takes about three times as long.
3.3 Prediction

Given draws of the joint posterior distribution of parameters and latent variables, it is straightforward to predict future covariances and consequently also future observations. This gives rise to the predictive density, defined as

\[ p(y_{t+1}|y_{[1:t]}) = \int_K p(y_{t+1}|y_{[1:t]}', \kappa) \times p(\kappa|y_{[1:t]}) \, d\kappa, \]  

(5)

where \( \kappa \) denotes the vector of all unobservables, i.e. parameters and latent variables. The superscript \( o \) in \( y_{[1:t]}' \) denotes \textit{ex post} realizations (observations) for the set of points in time \{1, \ldots, t\} of the \textit{ex ante} random values \( y_{[1:t]} = (y_1 \cdots y_t) \), see also Geweke and Amisano (2010) and Dawid and Musio (2015). The integration space \( K \) simply stands for the space of the possible values for \( \kappa \). Because (5) is the integral of the likelihood function where the values of \( \kappa \) are weighted according to their posterior distribution, it can be seen as the forecast density for an unknown value \( y_{t+1} \) after accounting for the uncertainty about \( \kappa \), given the history \( y_{[1:t]}' \).

As with most quantities of interest in Bayesian analysis, computing the predictive density is difficult because it constitutes an extremely high-dimensional integral which cannot be solved analytically. However, it may be approximated at a given “future” point \( y' \) through Monte Carlo integration,

\[ p(y' \mid y'_{[1:t]}) \approx \frac{1}{K} \sum_{k=1}^{K} p(y' \mid y'_{[1:t]}, \kappa^{(k)}_{[1:t]}), \]  

(6)

where \( \kappa^{(k)}_{[1:t]} \) denotes the \( k \)-th draw from the posterior distribution up to time \( t \). If (6) is evaluated at \( y' = y'_{t+1} \), it is commonly referred to as the (one-step-ahead) predictive likelihood at time \( t + 1 \), denoted \( PL_{t+1} \). Also, draws from (5) can straightforwardly be obtained by generating values \( y^{(k)}_{t+1} \) from the distribution given through the (in our case multivariate Gaussian) density \( p(y_{t+1}|y_{[1:t]}', \kappa^{(k)}_{[1:t]}) \).

Algorithm 2 (Predictive density and likelihood evaluation at time \( t + 1 \)).

1. Reduce the dataset to the training set \( y_{[1:t]}' = (y_1' \cdots y_t') \).

2. Run the posterior sampler using data from the training set only to obtain \( K \) pos-
terior draws $\kappa^{(k)}_{[1:t]}$, $k = 1, \ldots, K$.

3. For each $i \in \{1, \ldots, m\}$, simulate $K$ univariate draws from the conditional distribution $h_{i,t+1,[1:t]}|\mathbf{y}''_{t+1,[1:t]}, \kappa_{[1:t]}$ by drawing $h_{i,t+1,[1:t]}^{(k)}$ from a normal distribution with mean $\mu_{i,[1:t]}^{(k)} + \phi_{i,[1:t]}^{(k)} (h_{i,t,[1:t]}^{(k)} - \mu_{i,[1:t]}^{(k)})$ and standard deviation $\sigma_{i,[1:t]}^{(k)}$ for $k = 1, \ldots, K$.

4. For each $j \in \{1, \ldots, r\}$, simulate $K$ univariate draws from the conditional distribution $h_{m+j,t+1,[1:t]}|\mathbf{y}''_{t+1,[1:t]}, \kappa_{[1:t]}$ by drawing $h_{m+j,t+1,[1:t]}^{(k)}$ from a normal distribution with mean $\phi_{m+j,[1:t]}^{(k)} h_{m+j,t,[1:t]}^{(k)}$ and standard deviation $\sigma_{m+j,[1:t]}^{(k)}$ for $k = 1, \ldots, K$.

Obtain $K$ draws from the predictive distribution of the $j$th common latent factor $f_{m+j,t+1,[1:t]}$ by drawing from a normal distribution with mean 0 and variance $\exp h_{m+j,t+1,[1:t]}^{(k)}$ for $k = 1, \ldots, K$.

5. To obtain $PL_{t+1}$, average over $k = 1, \ldots, K$ densities of

$$
\mathcal{N}_m \left( \Lambda_{[1:t]}^{(k)} f_{t+1,[1:t]}^{(k)}, U_{t+1,[1:t]}^{(k)} \right),
$$

evaluated at $\mathbf{y}''_{t+1}$, where $U_{t+1,[1:t]}^{(k)} = \text{diag} \left( \exp h_{1,t+1,[1:t]}^{(k)}, \ldots, \exp h_{m,t+1,[1:t]}^{(k)} \right)$. Note that because the covariance matrix is a diagonal matrix, this method only requires univariate Gaussian evaluations and is thus computationally efficient. Nevertheless, because evaluation is done conditionally on realized values of $f_{t+1,[1:t]}$, it is extremely unstable in many dimensions. Moreover, since the numerical inaccuracy increases with an increasing number of factors $r$, this approach can lead to systematic undervaluation of $PL_{t+1}$ for larger $r$.

5*. Alternatively, to obtain $PL_{t+1}$, average over $k = 1, \ldots, K$ densities of

$$
\mathcal{N}_m \left( 0, \Sigma_{t+1,[1:t]}^{(k)} \right),
$$

evaluated at $\mathbf{y}''_{t+1}$, where $\Sigma_{t+1,[1:t]}^{(k)} = \Lambda_{[1:t]}^{(k)} V_{t+1,[1:t]}^{(k)} (\Lambda_{[1:t]}^{(k)})' + U_{t+1,[1:t]}^{(k)}$ with $V_{t+1,[1:t]}^{(k)} = \text{diag} \left( \exp h_{m+1,t+1,[1:t]}^{(k)}, \ldots, \exp h_{m+r,t+1,[1:t]}^{(k)} \right)$. Because this evaluation directly targets the marginal predictive density with respect to $f_{t+1,[1:t]}$, it is numerically more stable, irrespectively of the number of factors $r$. However, it requires a full $m$-variate Gaussian density evaluation for each $k$ and is thus computationally much
more expensive.\footnote{Note that because $\Sigma_t$ has rank $r$, using the Woodbury matrix identity,}

Because the marginal likelihood $ML = p(y^o) = \prod_{t=1}^{T} p(y^o_t | y^o_{1:t-1})$ can be decomposed into a product of $T$ predictive likelihoods, $PL_t$ is a disaggregated measure for evaluating model evidence and Algorithm 2 provides a direct way of computing it. Note however that these computations are exceptionally time-consuming because they require a complete model fit for each point in time. On the other hand, they can be executed in parallel in e.g. distributed computing environments such as clusters of workstations.

For comparing competing models $A$ and $B$ between time points $t_1$ and $t_2$, we consider cumulative predictive Bayes factors defined through

$$BF_{t_1,t_2}(A, B) = \frac{p_A(y^o_{t_1+1:t_2} | y^o_{1:t_1})}{p_B(y^o_{t_1+1:t_2} | y^o_{1:t_1})} = \prod_{t=t_1+1}^{t_2} \frac{PL_t(A)}{PL_t(B)}, \quad (9)$$

where $PL_t(A)$ and $PL_t(B)$ denote the predictive likelihood of model $A$ and $B$ at time $t$, respectively. When the cumulative predictive Bayes factor is greater than 1 (or its logarithm greater than 0) at a given point in time, there is evidence in favor of model $A$, and vice versa. Thereby, data up to time $t_1$ is regarded as prior information; out-of-sample evaluation starts at time $t_1 + 1$.

4 Simulation Study

The aim of this section is to apply the model to a simulated data set in order to illustrate the shrinkage properties of the Normal-Gamma prior for the factor loadings matrix elements. For this purpose, we simulate data from a two factor model for $m = 10$ time series of length $T = 1000$. For estimation, however, an overfitting model with three latent factors is employed. The nonzero parameter values used for simulation are picked randomly and are indicated as black circles in Figure 1; some loadings are set to zero, substantially speeds up the repetitive evaluation of the multivariate Gaussian distribution.
indicated by black dots. We set \( \Lambda_{ij} \) to zero if \( j > i \) for simulation and estimation.

In the following, we compare five specific prior settings. Setting 1, the usual standard normal prior with variance \( \tau^2_{ij} \equiv \tau^2 = 1 \), constituting the benchmark; setting 2, the row-wise Bayesian Lasso, where \( a_i = 1 \) for all \( i \); setting 3, the column-wise Bayesian Lasso, where \( a_j = 1 \) for all \( j \); setting 4, the Normal-Gamma prior with row-wise shrinkage, where \( a_i = 0.1 \) for all \( i \); setting 5, the Normal-Gamma prior with column-wise shrinkage, where \( a_j = 0.1 \) for all \( j \). Throughout this section, prior hyperparameters are chosen as follows: \( b_\mu = 0, B_\mu = 1000, B_\sigma = 1, B_{m+j} = 1 \) for \( j = 1, \ldots, r \). The prior hyperparameters for the persistence of the latent log variances are chosen as \( a_0 = 10, b_0 = 2.5 \) for the idiosyncratic volatilities and \( a_0 = 2.5, b_0 = 2.5 \) for the factor volatilities; note that the parameters of the superfluous factor are only identified through the prior. The shrinkage hyperhyperparameters are set as in Belmonte et al. (2014), i.e. \( c_i = c_j = d_i = d_j = 0.001 \) for all applicable \( i \) and \( j \). The algorithm is run for 25,000 iterations and the first 5,000 draws are discarded as burn-in. Using the \( R \) package \texttt{factorstochvol}\textsuperscript{6}, this takes about 2 minutes on a single core of a standard desktop computer.

Figure 1 shows smoothed kernel density estimates of posterior loadings under the different prior assumptions. The signs of the loadings have not been identified so that a multimodal posterior distribution hints at a “significant” loading whereas a unimodal posterior hints at a zero loading, see also Frühwirth-Schnatter and Wagner (2010).

It stands out that very little shrinkage (if any) is induced by the standard Gaussian prior. The other priors, however, impose considerably tighter posteriors. For the nonzero loadings on factor one, the row-wise Bayesian Lasso exhibits the strongest pull towards zero. Little difference between the various shrinkage priors can be spotted for the nonzero loadings on factor two.

Turning towards the zero loadings, strongest shrinkage is imposed by both variants of the Normal-Gamma prior, followed by the Bayesian Lasso variants and the standard Gaussian prior. This is particularly striking for the loadings on the superfluous third factor. The difference between row- and column-wise shrinkage for the Lasso variants can most clearly be seen in row nine and column three, respectively; row-wise Lasso captures the “zero-row” nine better, while column-wise Lasso captures the “zero-column” three

\textsuperscript{6}This package is in preparation to be published on CRAN, the Comprehensive \( R \) Archive Network.
Figure 1: Kernel density estimates of posterior factor loadings under different priors. The standard Gaussian prior (setting 1) in red solid strokes, the row-wise Lasso prior (setting 2) in blue long-dashed strokes, the column-wise Lasso prior (setting 3) in green short-dashed strokes, the row-wise Normal-Gamma prior (setting 4) in purple dotted strokes, the column-wise Normal-Gamma prior (setting 5) in orange dashed-dotted strokes. The vertical axis is capped at 30.
better. Because of the increased element-wise shrinkage of the Normal-Gamma prior, the difference between the row-wise and the column-wise variant are minimal. Note that under the Normal-Gamma priors, even after sign-identification, the posterior of series four’s loading on factor one, \( p(\Lambda_{4,1}|\mathbf{y}) \), is bimodal.

In the context of covariance modeling, however, factor loadings can be viewed upon as a mere means to parsimony, not the actual quantity of interest. Thus, Figure 2 displays the posterior estimate of the covariance matrix at the first and the last point of the sample \( (t = 1 \text{ and } t = T = 1000, \text{ respectively}) \) along with data generating values. The part below the diagonal depicts posterior estimates in setting 1 and the part above the diagonal depicts posterior estimates in setting 4. Eye-balling shows that the effect of different priors on posterior location and posterior variance is rather small in general; notice, however, that insignificant correlations depict less posterior uncertainty under the Normal-Gamma prior (setting 4). This stands out in particular for component series 9 which is uncorrelated with the other series.

![Figure 2: Estimated posterior correlation matrix at time \( t = 1 \) (left) and \( t = 1000 \) (right). The lower triangular parts stem from a model using the standard Gaussian prior (setting 1); the upper triangular parts stem from a model using the row-wise Normal-Gamma prior with \( a_i = 0.1 \) (setting 4). The inner and outer diameters of the discs are determined by posterior mean minus/plus two posterior standard deviations. Data generating values are indicated in black.](image)

To illustrate further, we also display selected estimated correlations over time in Figure 3. The top panel shows a posterior interval estimate (mean plus/minus two standard deviations) for the correlation of series 1 and series 2 (which is nonzero) under all four prior
settings; the bottom panel depicts that interval estimate for the correlation of series 9 and 10 (which is zero). While the relative differences between the settings in the nonzero correlation case are relatively small, the zero correlation case is picked up substantially better when shrinkage priors are used; posterior means are closer to zero and the posterior interval estimate is tighter.

![Figure 3](image)

**Figure 3**: “True” (gray, solid) and estimated posterior correlations between series 1 and series 2 (top) as well as series 9 and series 10 (bottom). To illustrate posterior uncertainty, estimates are depicted as point wise symmetrical intervals around the posterior mean, the width is 4 posterior standard deviations.

For a more comprehensive understanding of the shrinkage effect, relative RMSEs (root mean squared errors, averaged over time) between the true and the estimated pairwise correlations are depicted in Table 2. The part above the diagonal represents the relative performance of the row-wise Lasso prior (setting 2) with respect to the baseline prior (setting 1), the part below the diagonal represents the relative performance of the row-wise Normal-Gamma prior (setting 4) with respect to the row-wise Lasso prior (setting 2). Clearly, gains are highest for series nine which is by construction completely uncorrelated to the other series.

Additionally, geometric averages of these performance indicators are displayed in the first
row (setting 2 vs. baseline) and in the last row (setting 4 vs. baseline). They can be seen as the average relative performance of one specific series’ correlation estimates with all other series.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1.07</td>
<td>1.02</td>
<td>1.05</td>
<td>1.08</td>
<td>1.06</td>
<td>1.17</td>
<td>1.02</td>
<td>1.25</td>
<td>1.09</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.01</td>
<td>1.00</td>
<td>1.08</td>
<td>1.00</td>
<td>1.00</td>
<td>1.28</td>
<td>0.99</td>
<td>1.27</td>
<td>1.03</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
<td>1.01</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>1.16</td>
<td>1.03</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.99</td>
<td>1.00</td>
<td>1.10</td>
<td>1.00</td>
<td>1.00</td>
<td>1.27</td>
<td>1.00</td>
<td>1.09</td>
<td>1.04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.39</td>
<td>1.01</td>
<td>1.71</td>
<td>1.07</td>
<td>1.07</td>
<td>1.01</td>
<td>1.01</td>
<td>1.30</td>
<td>1.09</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.55</td>
<td>1.00</td>
<td>1.23</td>
<td>0.99</td>
<td>1.21</td>
<td>1.04</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.45</td>
<td>0.99</td>
<td>1.10</td>
<td>0.99</td>
<td>1.33</td>
<td>1.05</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.48</td>
<td>1.00</td>
<td>3.91</td>
<td>1.00</td>
<td>3.91</td>
<td>5.61</td>
<td>1.01</td>
<td>1.47</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.00</td>
<td>1.00</td>
<td>1.06</td>
<td>0.98</td>
<td>1.00</td>
<td>1.01</td>
<td>1.20</td>
<td>1.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2.63</td>
<td>1.97</td>
<td>2.53</td>
<td>2.09</td>
<td>2.58</td>
<td>2.44</td>
<td>1.66</td>
<td>2.11</td>
<td>1.26</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.96</td>
<td>0.99</td>
<td>1.02</td>
<td>1.67</td>
<td>0.99</td>
<td>0.96</td>
<td>3.95</td>
<td>1.05</td>
<td>2.17</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Relative RMSEs of pairwise correlations. Above the diagonal: Row-wise Lasso \( (a_i = 1) \) vs. benchmark standard Gaussian prior with geometric means (first row). Below the diagonal: Row-wise Normal-Gamma \( (a_i = 0.1) \) vs. row-wise Lasso prior \( (a_i = 1) \) with geometric means (last row). Numbers greater than one mean that the former prior performs better than the latter. Hyperhyperparameters are set to \( c_i = d_i = 0.001 \).

To illustrate the fact that extreme choices of \( c_i \) and \( d_i \) are crucial for the shrinkage effect of the Bayesian Lasso, Table 3 collects relative RMSEs for moderate hyperparameter choices \( c_i = d_i = 1 \). Note that the performance of the Bayesian Lasso deteriorates substantially while performance of the Normal-Gamma prior is relatively robust with regard to these choices. This indicates that the shrinkage effect of the Bayesian Lasso is strongly dependent on the particular choice of these hyperparameters (governing row-wise shrinkage), while the Normal-Gamma can adapt better through increased element-wise shrinkage.

An overall comparison of the errors under different priors is provided in Table 4 which lists RMSEs and MAEs for all prior settings, averaged over the non-trivial correlation matrix entries as well as time. Note again that results under the Lasso prior are sensitive to the particular choices of the global shrinkage hyperhyperparameters as well as the choice of row- or column-wise shrinkage, which is hardly the case for the Norma-Gamma prior. Interestingly, the performance gains achieved through shrinkage prior usage are higher when absolute errors are considered; this is coherent with the extremely high kurtosis of
Table 3: Relative RMSEs of pairwise correlations. Above the diagonal: Row-wise Lasso ($a_i = 1$) vs. benchmark standard Gaussian prior with geometric means (first row). Below the diagonal: Row-wise Normal-Gamma ($a_i = 0.1$) vs. row-wise Lasso prior ($a_i = 1$) with geometric means (last row). Numbers greater than one mean that the former prior performs better than the latter. Hyperhyperparameters are set to $c_i = d_i = 1$.

Normal-Gamma-type priors which, while placing most mass around zero, allow for large values.

Table 4: Different error measures ($\times 10^{-2}$) of posterior mean correlation estimates under various priors, averaged over $m = 10$ series. The column titled “GSH” contains the values of the global shrinkage hyperhyperparameters $c_i = c_j = d_i = d_j$.

To conclude, we briefly examine predictive performance by investigating cumulative log predictive Bayes factors. Thereby, the first 1000 points in time are treated as prior information, then 1-day- and 10-days-ahead predictive likelihoods are recursively evaluated until $t = 1500$. Table 5 displays the sum of these values for the respective models, standardized so that the baseline model 2-factor model with Gaussian prior has value zero. This way, numbers greater than zero can be interpreted as evidence in favor of the respective model, whereas numbers smaller than zero can be seen as evidence against (cf.
Jeffreys 1948).

<table>
<thead>
<tr>
<th></th>
<th>no fac</th>
<th>1 fac</th>
<th>2 fac</th>
<th>3 fac</th>
<th>4 fac</th>
<th>5 fac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Gaussian</td>
<td>-845.80</td>
<td>-316.41</td>
<td>-2.81</td>
<td>-5.83</td>
<td>-8.38</td>
<td></td>
</tr>
<tr>
<td>Row-wise NG (c_i = 1)</td>
<td>-845.80</td>
<td>-317.75</td>
<td>0.27</td>
<td>-0.44</td>
<td>-1.74</td>
<td>-2.34</td>
</tr>
<tr>
<td>Col-wise NG (c_j = 1)</td>
<td>-845.80</td>
<td>-317.45</td>
<td>0.38</td>
<td>-0.77</td>
<td>-1.61</td>
<td>-2.06</td>
</tr>
<tr>
<td>Row-wise NG (c_i = 0.01)</td>
<td>-845.80</td>
<td>-317.71</td>
<td>0.40</td>
<td>-0.29</td>
<td>-1.36</td>
<td>-1.97</td>
</tr>
<tr>
<td>Col-wise NG (c_j = 0.01)</td>
<td>-845.80</td>
<td>-317.52</td>
<td>1.04</td>
<td>-0.12</td>
<td>-0.88</td>
<td>-2.07</td>
</tr>
<tr>
<td>Standard Gaussian</td>
<td>-787.41</td>
<td>-255.92</td>
<td>-4.98</td>
<td>-7.75</td>
<td>-12.67</td>
<td></td>
</tr>
<tr>
<td>Row-wise NG (c_i = 1)</td>
<td>-787.41</td>
<td>-256.85</td>
<td>2.90</td>
<td>1.73</td>
<td>0.87</td>
<td>-0.05</td>
</tr>
<tr>
<td>Col-wise NG (c_j = 1)</td>
<td>-787.41</td>
<td>-256.44</td>
<td>3.14</td>
<td>1.17</td>
<td>0.90</td>
<td>0.09</td>
</tr>
<tr>
<td>Row-wise NG (c_i = 0.01)</td>
<td>-787.41</td>
<td>-256.81</td>
<td>3.18</td>
<td>1.98</td>
<td>1.28</td>
<td>0.53</td>
</tr>
<tr>
<td>Col-wise NG (c_j = 0.01)</td>
<td>-787.41</td>
<td>-255.90</td>
<td>2.85</td>
<td>1.61</td>
<td>0.84</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 5: Estimated log Bayes factors at \( t = 1500 \) against the 2-factor model using a standard Gaussian prior, where data up to \( t = 1000 \) is treated as the training sample. Lines 1 to 5 correspond to cumulative 1-day-ahead Bayes factors, lines 6 to 10 correspond to 10-days-ahead predictive Bayes factors.

Not very surprisingly, log Bayes factors are highest for the 2-factor model; within this class, models imposing stronger shrinkage perform slightly better, in particular when considering the longer 10-day horizon. Underfitting models predict very poorly both on the short and the longer run, and overfitting models are also somewhat worse. Note that shrinkage safeguards against overfitting, at least to a certain extent.

The above findings are rather consistent if dimensionality is increased; in analogy to above, Table 6 reports the overall RMSEs and MAEs for 4950 pairwise correlations, resulting from \( m = 100 \) component series at \( T = 1000 \) points in time. The factor loadings have again been generated randomly with about 50% of the loadings being equal to zero. For estimation, no restriction on the factor loadings matrix was imposed.

5 Application to S&P 500 Data

In this section we apply the SV factor model to stock prices listed in the Standard & Poor’s 500 index. We only consider firms which have been continuously included from November 1994 until December 2013, resulting in \( m = 300 \) stock prices on 5001 days, ranging from 11/1/1994 to 12/31/2013. Data was obtained from Bloomberg Terminal in January 2014. Instead of considering raw prices, we investigate \( T = 5000 \) percentage log-returns which we demean a priori. The first 10 members of this collection are displayed.
Table 6: Different error measures ($\times 10^{-2}$) of posterior mean correlation estimates under various priors, averaged over $m = 100$ series. The column titled “GSH” contains the values of the global shrinkage hyperhyperparameters $c_i = c_j = d_i = d_j$.

The presentation consists of two parts. First, we exemplify inference using a multivariate stochastic volatility model and discuss the outcome. Second, we perform out-of-sample predictive evaluation and compare different models. To facilitate interpretation of the results discussed in this section, we consider the GICS\footnote{Global Industry Classification Standard, retrieved from https://en.wikipedia.org/w/index.php?title=List_of_S%26P_500_companies&oldid=589980759 on April 11, 2016.} classification into 10 sectors listed in Table 7.

<table>
<thead>
<tr>
<th>GICS sector</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer Discretionary</td>
<td>45</td>
</tr>
<tr>
<td>Consumer Staples</td>
<td>28</td>
</tr>
<tr>
<td>Energy</td>
<td>23</td>
</tr>
<tr>
<td>Financials</td>
<td>54</td>
</tr>
<tr>
<td>Health Care</td>
<td>30</td>
</tr>
<tr>
<td>Industrials</td>
<td>42</td>
</tr>
<tr>
<td>Information Technology</td>
<td>27</td>
</tr>
<tr>
<td>Materials</td>
<td>23</td>
</tr>
<tr>
<td>Telecommunications Services</td>
<td>3</td>
</tr>
<tr>
<td>Utilities</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 7: GICS sectors and the amount of members within the S&P 500 data set.

5.1 A four-factor model for 300 S&P 500 members

To keep graphical representation feasible, we only focus on the latest 2000 returns of our data set, i.e. 5/3/2006 to 12/31/2013. This time frame is chosen to include both the 2008
Figure 4: The first 20 components of $m = 300$ demeaned stock price percentage log-returns listed in the S&P 500 index.
financial crisis as well as the period before and thereafter. Furthermore, we restrict our discussion to a four-factor model. This choice is somewhat arbitrary and is mainly made for the sake of convenience. Comparison of predictive performance for varying number of factors and different time frames will be discussed in Section 5.2.

We run our sampler with the Normal-Gamma prior for 110 000 draws and discard the first 10 000 draws as burn-in. Of the remaining 100 000 draws every 10th draw is kept, resulting in 10 000 draws used for posterior inference. Hyper- and hyperhyperparameters are set as follows: $a_i \equiv a = 0.1$, $c_i \equiv c = 1$, $d_i \equiv d = 1$, $b_\mu = 0$, $B_\mu = 100$, $a_0 = 20$, $b_0 = 1.5$, $B_\sigma = 1$, $B_{m+j} = 1$ for $j = 1, \ldots, r$. To prevent factor switching, we set all elements above the diagonal to zero. The leading series are chosen manually after a preliminary unidentified run such that series with high loadings on that particular factor (but low loadings on the other factors) become leaders. Note that this intervention (which introduces an order dependency) is only necessary for interpreting the factor loadings matrix but not for covariance estimation or prediction.

To illustrate the substantial degree of volatility co-movement, mean posterior variances are displayed in Figure 5. This picture resembles one where all series are modeled with independent univariate stochastic volatility models. Clear spikes can be spotted during the financial crisis in late 2008 but also in early 2010 and late 2011.

![Figure 5: Mean posterior variances $E(\text{diag}(\Sigma_t) | y)$ for $t = 1, \ldots, T$ (logarithmic scale).](image)

![Figure 6: Posterior mean of the joint communalities $C_t$ (bold line) along with mean plus/minus two posterior standard deviations (light lines).](image)

This picture is mirrored (to a certain extent) in Figure 6, displaying the joint commu-
nality $C_t$, i.e. the average proportion of variances explained through the common factors. In particular during the financial crisis, the first half of 2010 and late 2011 the joint communality reaches high values of 0.7 and more.

Median posterior factor loadings are visualized in Figure 7. In the top panel it can be seen that all series significantly load on the first factor which consequently could be interpreted to represent the “overall state of the economy”. Highly loading elements include United States Steel Corp. (X) and Cliffs Natural Resources Inc. (CLF), both of which belong to the sector Materials and both of which have been dropped from the S&P 500 index in 2014 due to market capitalization changes. Cummins Inc. (CMI, Industrials) and PulteGroup, Inc. (PHM, Consumer Discretionary) rank third and fourth. Companies in sectors Consumer Staples, Utilities and Health Care tend to load comparably low on this factor.

Investigating the second factor, it stands out that due to the use of the Normal-Gamma prior a considerable amount of loadings are shrunk towards zero. Main drivers are all in the sector Utilities. Also, companies in sectors Consumer Staples, Health Care and (to a certain extent) Financials load positively here. Both the loadings on factor 3 as well as the loadings on factor 4 are substantially shrunk towards zero. Notable exceptions are Energy and Materials companies for factor 3 and Financials for factor 4.

The corresponding factor log variances are displayed in Figure 8. Even though some similarities can be spotted, each process exhibits specific characteristics. Note for instance the sharp increase of volatility in early 2010 which is mainly visible for the “overall” factor 1; the Utilities (factor 2) specific pre-crisis volatility peak during early 2008; the generally smooth volatility behavior of the Energy and Materials driven factor 3; the comparably “nervous” volatility evolution of factor 4 governed by the Financials.

To conclude, we show three examples of the mean posterior correlation matrix $\Sigma_t$ in Figure 9. The series are grouped according to the alphabetically ordered industry sectors (and simply sorted according to their ticker symbol therein). A video displaying the mean correlation matrix for all points in time can currently be downloaded from http://statmath.wu.ac.at/~kastner/corvid.mp4.

Considering the last trading day in 2006, obviously highly correlated clusters appear within Energy and Utilities, to a certain extent also within Financials, Industrials and
Figure 7: Median loadings on the first two factors (top) and the last two factors (bottom) of a 4-factor model applied to $m = 300$ demeaned stock price log-returns listed in the S&P 500 index. Shading: Sectors according to the Global Industry Classification Standard.
Figure 8: Latent factor log variances $h_{m+j}$, $j = 1, \ldots, 4$ (top to bottom). Bold line indicates the posterior mean; light lines indicate mean ± 2 standard deviations.

Figure 9: Posterior mean of the time-varying correlation matrix $E(\Sigma|y)$, exemplified for $t \in \{173, 696, 1218\}$ which corresponds to the last trading day in 2006, 2008, 2010, respectively. The matrix has been rearranged to reflect the different GICS sectors in alphabetical order, cf. Table 7.
Materials. Not very surprisingly, there appears to be low correlation between companies in the sectors Consumer Discretionary/Staples and Energy but higher correlation between Energy, Industrials and Materials. Looking at the last trading day of 2008, the overall picture changes radically. Higher correlation can be spotted throughout, both within sectors but also between sectors. There are only few companies who show little and practically no companies that show no correlation with others. Another two years later, we again see a different overall picture. Lower correlations throughout become apparent with moderate correlations remaining within the sectors Energy, Utilities, and in particular Financials.

5.2 Prediction

Even for univariate volatility models, evaluating in- or out-of-sample fit is not straightforward because the quantity of interest (the conditional standard deviation) is not directly observable. While in lower dimensions this issue can be circumvented to a certain extent using intraday data and computing realized measures of volatility, the difficulty becomes more striking when the dimension increases. Thus, we focus on iteratively predicting the observation density out-of-sample which is then evaluated at the actually observed values. Because this approach involves re-estimating the model for each point in time, it is computationally costly but can be parallelized in a trivial fashion on multi-core computers.

For the S&P 500 data set, we begin by using the first 3000 data points (until 5/2/2006) to estimate the one-day-ahead predictive likelihood for day 3001 as well as the ten-day-ahead predictive likelihood for day 3010. In a separate estimation procedure, the first 1001 data points (until 5/3/2006) are used to estimate the one-day-ahead predictive likelihood for day 3002 and the corresponding ten-day-ahead predictive likelihood for day 3011, etc. This procedure is repeated for 500 days, yielding the log predictive Bayes factors displayed in Figure 10.

We use a no-factor model as the baseline; this corresponds to 300 individual stochastic volatility models fitted to each component series separately. For each date, values greater than zero mean that the model outperforms the baseline model up to that point in time. Competitors of the no-factor SV model are \( r \)-factor SV models with \( r = 1, \ldots, 15 \) under
Figure 10: Cumulative sums of one-day-ahead log predictive Bayes factors over series-specific univariate SV models. Solid lines indicate values under the standard normal prior, dashed lines indicate values obtained under the Normal-Gamma prior.

the usual standard normal prior and under the Normal-Gamma prior with \( a_i \equiv 0.1 \) and \( c_i \equiv d_i \equiv 1 \) employing row-wise-shrinkage. All other parameters are kept identical, i.e. \( b_\mu = 0, B_\mu = 100, a_0 = 20, b_0 = 2.5 \) (idiosyncratic persistences), \( a_0 = 2.5, b_0 = 2.5 \) (factor persistences), \( B_\sigma = 0.1, B_{m+j} = 0.1 \) for \( j = 1, \ldots, r \). Note that because the object of interest in this exercise does not require the factor loadings matrix to be identified, no a priori restrictions are placed on \( \Lambda \). This alleviates the problem of arranging the data in any particular order before running the sampler.

While joint models with \( r > 0 \) outperform the marginal (no-factor) model for all points in time, 2/27/2007 particularly stands out. This day corresponds to the burst of the Chinese stock bubble of 2007, a major crash in Chinese stock markets causing severe drops in markets worldwide.\(^8\) It appears that joint modeling of stock prices is particularly important on days of extreme events, where conditional correlations are often higher.

Overall log predictive likelihoods for the entire period (corresponding to the rightmost values depicted in Figure 10) are displayed in Figure 11. Gains in predictive power are substantial up to around eight factors, with little difference for the two priors. After this

\(^8\)The pronounced jump in log Bayes factors over the no-factor model is visible in all sectors with the exception of “Energy”.
point, the benefit of adding even more factors becomes less pronounced; the effect of the priors however becomes stronger. Again, while differences in scores are minor for models with fewer factors, the benefit of shrinkage grows when $r$ gets larger. E.g., the estimated 1-day and 10-days log Bayes factors of the 15-factor model in favor of the NG prior over the standard normal prior amount to 336 and 628, respectively.

![Figure 11: Estimated 1-day- and 10-days-ahead log predictive likelihoods, accumulated up to $t = 3500$, for models with 0, 1, . . . , 15 factors. Data until $t = 3000$ is treated as training data.](image)

6 Conclusion and Outlook

The aim of this paper was to present an efficient and parsimonious method of estimating high-dimensional time-varying covariance matrices through factor stochastic volatility models. We did so by proposing an efficient Bayesian MCMC algorithm that incorporates parsimony by modeling the covariance structure through common latent factors which themselves follow univariate SV processes. Moreover, we added additional sparsity by utilizing a hierarchical shrinkage prior, the Normal-Gamma prior, on the factor loadings. We showed the effectiveness of our approach through simulation studies and illustrated the effect of different choices of the “shrinkage parameter” $a$. We applied the algorithm to a high-dimensional data set consisting of stock returns of 300 S&P 500 members.
and conducted an out-of-sample predictive study to compare different prior settings and investigate the choice of the number of factors.

Because the algorithm scales linearly in both the series length $T$ as well as the number of component series $m$, applying it to even higher dimensions is straightforward. We have experimented with simulated data in thousands of dimensions for thousands of points in time and successfully recaptured the time-varying covariance matrix. Further research could also be directed towards incorporating prior knowledge into building the hierarchical structure of the Normal-Gamma prior, e.g. by choosing the global shrinkage parameters according to industry sectors.

**Acknowledgments**

Variants of this paper were presented at the 6th European Seminar on Bayesian Econometrics (ESOBE), the 2015 NBER-NSF Time Series Conference, the 2nd Vienna Workshop on High-Dimensional Time Series in Macroeconomics and Finance, the 2015 International Work-Conference on Time Series Analysis, and the 2016 ISBA World Meeting. The author thanks all participants, in particular Manfred Deistler, Sylvia Frühwirth-Schnatter, John Geweke, Hedibert Lopes, Ruey Tsay, Herman van Dijk, Mike West, and the ESOBE-discussant Mark Jensen for crucially valuable comments and suggestions.
References


Yu, Y. and X.-L. Meng (2011). “To center or not to center: that is not the question—An ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency”.