MULTIVARIATE BEHAVIORAL RESEARCH
2021, VOL. 56, NO. 1, 120–149

A Tutorial on Estimating Time-Varying Vector Autoregressive Models

Jonas M. B. Haslbeck[a], Laura F. Bringmann[b], and Lourens J. Waldorp[a]

[a]Psychological Methods Group, University of Amsterdam; [b]Department of Psychometrics and Statistic, University of Groningen

ABSTRACT

Time series of individual subjects have become a common data type in psychological research. These data allow one to estimate models of within-subject dynamics, and thereby avoid the notorious problem of making within-subjects inferences from between-subjects data, and naturally address heterogeneity between subjects. A popular model for these data is the Vector Autoregressive (VAR) model, in which each variable is predicted by a linear function of all variables at previous time points. A key assumption of this model is that its parameters are constant (or stationary) across time. However, in many areas of psychological research time-varying parameters are plausible or even the subject of study. In this tutorial paper, we introduce methods to estimate time-varying VAR models based on splines and kernel-smoothing with/without regularization. We use simulations to evaluate the relative performance of all methods in scenarios typical in applied research, and discuss their strengths and weaknesses. Finally, we provide a step-by-step tutorial showing how to apply the discussed methods to an openly available time series of mood-related measurements.

KEYWORDS

VAR models; time-varying models; non-stationarity; time series analysis; intensive longitudinal data; ESM

1. Introduction

The ubiquity of mobile devices has led to a surge in time series (or intensive longitudinal) data sets from single individuals (e.g., Bak et al., 2016; Bringmann et al., 2013; Fisher et al., 2017; Groen et al., 2019; Hartmann et al., 2015; Kramer et al., 2014; Kroeze et al., 2016; Snippe et al., 2017; van der Krieke et al., 2017). This is an exciting development because these data allow one to model within-subject dynamics, which avoids the notorious problem of making within-subjects inferences from between-subjects data, and naturally addresses heterogeneity between subjects (Fisher et al., 2018; Molenaar, 2004). The ability to analyze within-subjects data therefore promises to be a major leap forward both for psychological research and applications in (clinical) practice.

A key assumption of all standard time series models is that all parameters of the data generating model are constant (or stationary) across the measured time period. This is called the *assumption of stationarity*.[1] While one often assumes constant parameters, changes of parameters over time are often plausible in psychological phenomena. As an example, take the repeated measurements of the variables *Depressed Mood*, *Anxiety* and *Worrying*, modeled by a time-varying first-order Vector Autoregressive (VAR) model shown in Figure 1. In week 1, there are no cross-lagged effects between any of the three variables. However, in week 2 we observe a cross-lagged effect from *Worrying* on *Mood*. A possible explanation could be a physical illness in week 2 that moderates the two cross-lagged effects. In week 3, we observe a cross-lagged effect from *Anxiety* on *Mood*. Again, this could be due to an unobserved moderator like a stressful period at work. The fourth visualization shows the average of the previous three models, which is the model one would obtain by estimating a stationary VAR model on the entire time series. In this situation, the stationary model is clearly inappropriate because it is different to the true model across *all* intervals of the time series.

Time-varying models are of central interest when studying psychological phenomena from a within-person perspective. For example, in the network approach to psychopathology, it is suggested that mental

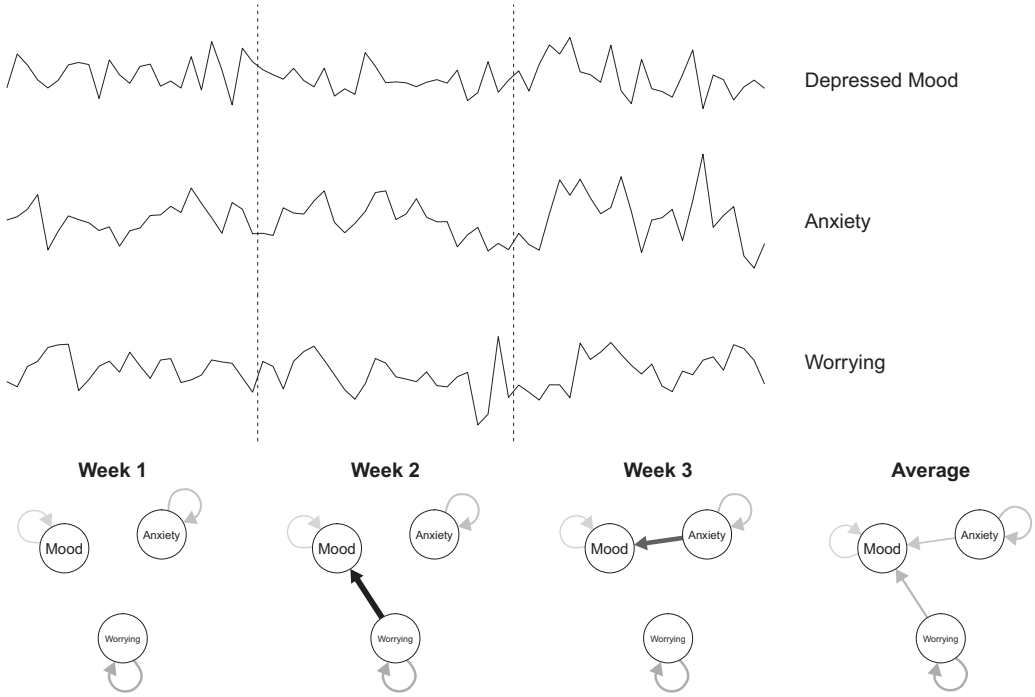

Figure 1. Upper panel: hypothetical repeated measurements of *Depressed Mood*, *Anxiety* and *Worrying*, generated from a time-varying lag 1 VAR model. Lower panel: the time-varying VAR-model generating the data shown in the upper panel. It consists of three models, one for each week. The fourth model (left to right) indicates the average of the three models, which is what one obtains when estimating a stationary VAR model on the entire time series.

disorders arise from causal interactions among symptoms (see also Borsboom & Cramer, 2013; Robinaugh et al., 2019; Schmittmann et al., 2013). This means that the interactions between symptoms are different for healthy and unhealthy individuals (Pe et al., 2015; van Borkulo et al., 2015) and that the interactions result in an individual change when she or he transitions from a healthy to an unhealthy state (or vice versa). Time-varying models are able to capture this change. Next to detecting these changes, they may also shed light on why those changes occurred. For example, one could correlate time-varying parameters with contextual factors such as elevated stress levels, social setting or major life events and thereby possibly uncover conditions and events that predict the onset of mental disorders. Time-varying models can also be used to study how parameters change in response to interventions. For example, in Section 4 we will fit a time-varying VAR model on ESM measurements during a double-blind medication reduction study (Wichers et al., 2016).

Time-varying models are also central to the idea of Early Warning Signals (EWS; Scheffer et al., 2009). For example, Wichers et al. (2016) suggested to anticipate phase-transitions between healthy and unhealthy states with EWS such as time-varying autocorrelation and variance (see also van de Leemput et al., 2014). Time-varying VAR models are an extension of these EWS to multivariate time-series. Anticipating the sensitive periods around phase transitions is interesting, because during those periods treatment may be more efficient (Olthof et al., 2019). This means that time-varying models could be used as a tool to monitor patients and determine periods during which treatment is most promising.

In this tutorial paper we provide an introduction to how to estimate a time-varying version of the Vector Autoregressive (VAR) model, which is arguably the simplest multivariate time series model for temporal dependencies in continuous data, and is used in many of the papers cited above. We will focus on two sets of methods recently proposed by the authors to estimate such time-varying VAR models: Bringmann et al. (2018) presented a method based on splines using the Generalized Additive Modeling (GAM) framework, which estimates time-varying parameters by modeling them as a spline function of time; and Haslbeck and Waldorp (2018b) suggested a method based on penalized kernel-smoothing (KS), which estimates time-varying parameters by combining the estimates of several local models spanning the entire time series. While both methods are available to applied researchers, it is unclear how well they and their variants (with/without regularization or significance testing) perform in situations that are typical in applied research. We aim to improve this situation by making the following contributions:

1. We report the performance of GAM based methods with and without significance testing, and the performance of KS based methods with and without regularization in situations that are typical for Experience Sampling Method (ESM) studies.
2. We discuss the strengths and weaknesses of all methods and provide practical recommendations for applied researchers.
3. We compare time-varying methods to their corresponding stationary counterparts to address the question of how many observations are necessary to identify the time-varying nature of parameters.
4. We provide tutorials on how to estimate time-varying VAR models using both methods on an openly available intensive longitudinal dataset using the R-packages *mgm* and *tvvarGAM*.

The paper is structured as follows. In Section 2.1 we define time-varying VAR models, which are the focus of this paper. We next present two sets of methods to estimate such models: one method based on splines with and without significance testing (Section 2.2), and one method based on kernel estimation with and without regularization (Section 2.3). In Sections 3.1 and 3.2 we report two simulation studies that investigate the performance of these two models and their stationary counterparts. In Section 4 we provide a fully reproducible tutorial on how to estimate a time-varying VAR model from an openly available time series data set collected in ESM studies using the kernel smoothing method using the R-package *mgm* (we repeat the same tutorial with the GAM method in the appendix). Finally, in Section 5 we discuss possible future directions for research on time-varying VAR models.

2. Estimating time-varying VAR models

We first introduce the notation for the stationary first-order VAR model and its time-varying extension (Section 2.1) and then present the two methods for estimating time-varying VAR models: the GAM-based method (Section 2.2) and the penalized kernel-smoothing-based method (Section 2.3). We discuss implementations of related methods in Section 2.4.

2.1. Vector autoregressive (VAR) model

In the first-order Vector Autoregressive (VAR(1)) model, each variable at time point t is predicted by all variables (including itself) at time point $t-1$. Next to a set of intercept parameters, the VAR(1) model is comprised by autoregressive effects, which indicate how much a variable is predicted by itself at the previous time point, and cross-lagged effects, which indicate how much a variable is predicted by all other variables at the previous time point.

Formally, the variables $\mathbf{X}_t \in \mathbb{R}^p$ at time point $t \in \mathbb{Z}$ are modeled as a linear combination of the same variables at $t-1$

$$\begin{aligned}\mathbf{X}_t &= \boldsymbol{\beta}_0 + \boldsymbol{B}\mathbf{X}_{t-1} + \boldsymbol{\varepsilon} = \begin{bmatrix} X_{t,1} \\ \vdots \\ X_{t,p} \end{bmatrix} \\ &= \begin{bmatrix} \beta_{0,1} \\ \vdots \\ \beta_{0,p} \end{bmatrix} + \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{1,p} \\ \vdots & \ddots & \vdots \\ \beta_{p,1} & \cdots & \beta_{p,p} \end{bmatrix} \begin{bmatrix} X_{t-1,1} \\ \vdots \\ X_{t-1,p} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_p \end{bmatrix},\end{aligned} \tag{1}$$

where $\beta_{0,1}$ is the intercept of variable 1, $\beta_{1,1}$ is the autoregressive effect of $X_{t-1,1}$ on $X_{t,1}$, and $\beta_{p,1}$ is the cross-lagged effect of $X_{t-1,1}$ on $X_{t,p}$, and we assume that $\boldsymbol{\varepsilon} = \{\epsilon_1, \ldots, \epsilon_p\}$ are independent (across time points) samples drawn from a multivariate Gaussian distribution with variance-covariance matrix Σ. In this paper we do not model Σ, however, it can be obtained from the residuals of the model and used to estimate the inverse covariance matrix (see e.g., Epskamp et al., 2018).

Throughout the paper we deal with first-order VAR models in which all variables at time point t are a linear function of all variables at time point $t-1$. In the interest of brevity we will therefore refer to this first-order VAR model (or VAR(1) model) as a VAR model. More lags can be included by adding further parameter matrices and lagged variable vectors $\boldsymbol{X}_{t-k}$ (for a lag of k) to the model in Equation (1). Note that while we focus on VAR(1) models in the this paper, the presented methods can be used to estimate time-varying VAR models with any set of lags. For a detailed description of VAR models we refer the reader to Hamilton (1994).

In both the GAM and the KS method we estimate the model in (1) by predicting each of the variables $X_{t,i}$ for $i \in \{1, \ldots, p\}$ separately. Specifically, we model

$$\begin{aligned}X_{t,i} &= \beta_{0,i} + \boldsymbol{\beta_i}\mathbf{X}_{t-1} + \epsilon_i \\ &= \beta_{0,i} + \begin{bmatrix} \beta_{i,1} & \cdots & \beta_{i,p} \end{bmatrix} \begin{bmatrix} X_{t-1,1} \\ \vdots \\ X_{t-1,p} \end{bmatrix} + \epsilon_i,\end{aligned} \tag{2}$$

for all $i \in \{1, \ldots, p\}$, where $\boldsymbol{\beta_i}$ is the $1 \times p$ vector containing the lagged effects on $X_{t,i}$. After estimating the parameters in each equation, we combine all estimates to the VAR model in (1).

In order to turn the stationary VAR model in (1) into a time-varying VAR model, we introduce a time

index for the parameter matrices

$$\mathbf{X}_t = \boldsymbol{\beta}_{0,t} + \boldsymbol{B}_t \mathbf{X}_{t-1} + \boldsymbol{\varepsilon}. \quad (3)$$

This allows a different parameterization of the VAR model at *each time point* and thereby allows the model to vary across time. Throughout this paper we assume that the time-varying parameters are smooth deterministic functions of time. We define a smooth function as a function for which the first derivative exists everywhere. In the following two subsections we introduce two different ways to estimate such a time-varying VAR model.

The VAR model has often been discussed and visualized as a network model (Epskamp et al., 2018), and also here we will use both statistical and network/graph terminology. To avoid confusion between the two terminologies, we explicitly state how the terms in the two terminologies correspond to each other. From the statistical perspective there are two types of lagged effects between pairs of variables: autocorrelations (e.g., $X_{t-1} \rightarrow X_t$) and cross-lagged effects (e.g., $X_{t-1} \rightarrow Y_t$). In the network terminology variables are nodes, and lagged effects are represented by directed edges. An edge from a given node on itself is also called a self-loop, and represents autocorrelation effects. The value of lagged effects is represented in sign and the absolute value of the edge-weights of the directed edges. If an edge-weight between variables X_t and Y_{t-1} is nonzero, we say that the edge from X_t and Y_{t-1} is present. *Sparsity* refers to how strongly connected a network is: if many edges are present, sparsity is low; if only few edges are present, sparsity is high. On a node-level, sparsity is captured by the *indegree* (how many edges point toward a node) and *outdegree* (how many edges point away from a node). In statistical terminology indegree is the number of incoming lagged effects on variable X, and outdegree the number outgoing lagged effects from variable X.

2.2. The GAM method

In this section we explain how to estimate a time-varying VAR model using the Generalized Additive Model (GAM) framework, which allows for non-linear relationships between variables (see also Bringmann et al., 2018, 2017). We leverage the GAM framework for the estimation for time-varying models by using it to define each parameter as a function of time. Because GAMs are able to represent non-linear functions, this allows us to recover non-linear time-varying parameters. In what follows we illustrate how this approach works for the simplest possible example, a model consisting only of a time-varying intercept parameter, $y = \beta_{0,t} + \varepsilon$.

Panel (a) of Figure 2 shows that the values of y are varying over time, so the intercept will have to be time-varying as well, if the intercept-only model is supposed to fit the data well. This is achieved by summing the following five basis functions

$$\hat{\beta}_{0,t} = \hat{\alpha}_1 R_1(t) + \hat{\alpha}_2 R_2(t) + \hat{\alpha}_3 R_3(t) + \hat{\alpha}_4 R_4(t) + \hat{\alpha}_5 R_5(t), \quad (4)$$

which are displayed in panels (b)–(f) in Figure 2. Panel (g) overlays all used basis functions, and panel (h) displays the estimate of the final smooth function $\hat{\beta}_{0,t}$, which is obtained by adding up the weighted basis functions ($\hat{\alpha}$) (see panel (g) and (h) of Figure 2). The optimal regression weights are estimated using standard linear regression techniques. The same rationale is applied to every time-varying parameter in the model.

There are several different spline bases such as cubic, P-splines, B-splines, and thin plate splines. The advantage of thin plate splines, which is the basis used here, is that one does not have to specify knot locations, resulting therefore in fewer subjective decisions that need to be made by the researcher (Wood, 2006). The basis functions in Figure 2 exemplify the thin plate spline basis. In the figure, panels (b)-(f) show that each additional basis function (R) increases the nonlinearity of the final smooth function. This is reflected in the fact that every extra basis function is more "wiggly" than the previous basis functions. For example, the last basis function in panel (f) is "wigglier" than the first basis function in panel (b). The spline functions used here are smooth up to the second derivative. Thus, a key assumption of the GAM method is that all true time-varying parameter functions are smooth as well. This assumption is also called the assumption of *local stationarity*, because smoothness implies that the parameter values that are close in time are very similar, and therefore locally stationary. This would be violated by, for example, a step function, where the GAM method would provide incorrect estimates around a "jump" (but would still provide good estimates for the two constant parts).

As the number of basis functions determines the nonlinearity of the smooth function (e.g., $\hat{\beta}_{0,t}$), a key problem is how to choose the optimal number of basis functions. The final curve should be flexible enough to be able to recover the true model, but not too flexible as this may lead to overfitting (Andersen,

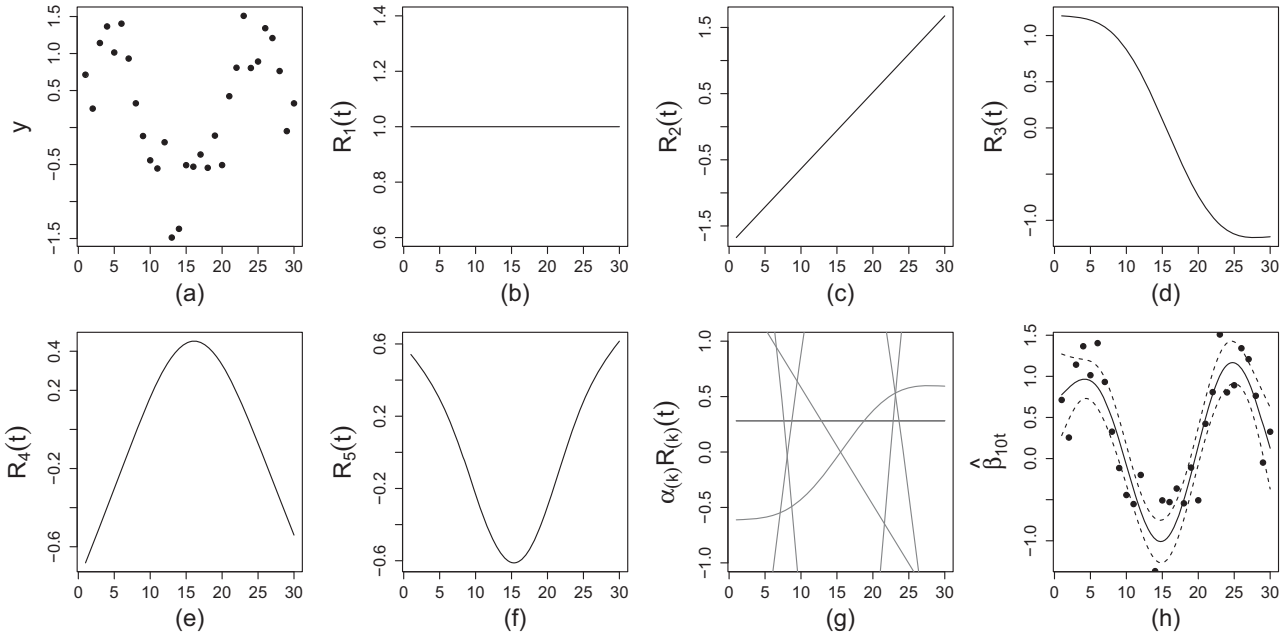

Figure 2. An example of the basis function for a time-varying parameter $\hat{\beta}_{0,t}$. In panel (a) the data are shown. In panel (b)–(f) the estimated 5 basis functions are given and panel (g) shows the weighted basis functions. In the last panel (h) the final smooth function is illustrated with credible intervals around the smooth function.

2009; Keele, 2008). The method used here to find the optimal number of basis functions is penalized likelihood estimation (Wood, 2006). Instead of trying to select the optimal number of basis functions directly, one can simply start by including more basis functions than would be normally expected, and then adjust for too much wiggliness with a *wiggliness penalty* (Wood, 2006).

Thus, the problem of selecting the right number of basis functions is reduced to selecting the right wiggliness penalty. This is achieved using generalized cross-validation (Golub et al., 1979), where the penalty parameter with the lowest Generalized Cross-Validation (GCV) value is expected to provide a good bias-variance trade-off. Specifically, the penalization decreases the influence of the basis functions (R) by reducing the values of their regression coefficients ($\hat{\alpha}$). Therefore, smoothness is imposed on the curve both through the choice of the number of basis functions and the final level of penalization on these basis functions.

To estimate time-varying VAR models with the GAM method, we use the *tvvarGAM* package in *R* (Bringmann, Haslbeck, & Tendeiro, 2020), which is a wrapper around the *mgcv* package (Wood, 2006). As the wiggliness penalty is automatically determined, the user only needs to specify a large enough number of basis functions. The default settings are the thin plate regression spline basis and 10 basis functions, which although an arbitrary number, is often sufficient (see the simulation results in Bringmann et al., 2017). The minimum number is in most models three basis functions. In general, it is recommended to increase the number of basis functions if it is close to the effective degrees of freedom (edf) selected by the model. The effective degrees of freedom is a measure of nonlinearity. A linear function has an edf of one, and higher edf values indicate wigglier smooth functions (Shadish et al., 2014).

The GAM function in the *mgcv* package outputs the final smooth function, the GCV value and the edf. Furthermore, the uncertainty about the smooth function is estimated with 95% Bayesian credible intervals (Wood, 2006). In the remainder of this manuscript we refer to this method as the GAM method. We refer to a variant of the GAM method, in which we set those parameters to zero whose 95% Bayesian credible interval overlaps with zero, with GAM(st), for "significance thresholded." With GLM we refer to the standard unregularized VAR estimator.

After the model is estimated, it is informative to check if the smooth functions were significantly different from zero (at some point over the whole time range), and if each smooth function had enough basis functions. Significance can be examined using the p-values of each specific smooth function, which indicates whether the smooth function is significantly different from zero. To see whether there are enough basis functions, the edf of each smooth function can be examined. The edf value should be well below the

Time	$X_{t,1}$	$X_{t-1,1}$	...	$X_{t,p}$	$w_{t_e=3}$	$w_{t_e^*=3}$
1	0.03	−0.97	...	−0.08	0.61	0.00
2	1.15	−1.07	...	−0.56	0.88	0.14
3	0.11	0.63	...	1.09	1.00	1.00
4	−1.08	0.13	...	1.88	0.88	0.14
5	−0.93	1.00	...	−0.29	0.61	0.00
6	−1.08	0.17	...	−1.36	0.32	0.00
7	0.27	−1.72	...	−1.13	0.14	0.00
8	0.03	−1.26	...	−0.97	0.04	0.00
9	−1.29	−1.05	...	−0.10	0.01	0.00
10	−0.07	−0.04	1.05	−0.12	0.00	0.00

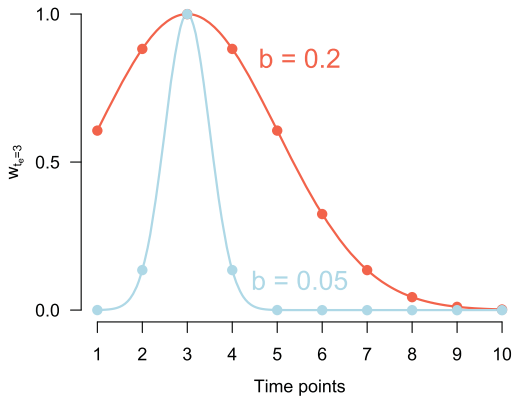

Figure 3. Illustration of the weights defined to estimate the model at time point t_e = 3. Left panel: a kernel function defines a weight for each time point in the time series. Right panel: the weights shown together with the VAR design matrix constructed to predict $X_{t,1}$.

maximum possible edf or the number of basis functions for the smooth function (or term) of interest (in our case 10, see Wood, 2006). When the edf turns out to be too high, the model should be refitted with a larger (e.g., double) number of basis functions.

2.3. The kernel-smoothing method

In the kernel-smoothing method one obtains time-varying parameters by estimating and combining a sequence of local models at different time points across the time series. A local model is estimated by weighting all observations depending on how close they are to the time point at which the local model is estimated. In Figure 3 we show an example in which a single local model is estimated at time point t_e = 3. We do this by giving the time points close to t_e a high weight and time points far away from t_e a very small or zero weight. If we estimate models like this on a sequence of equally spaced estimation points across the whole time series and take all estimates together, we obtain a time-varying model.

To define the weight at each time point we use a Gaussian kernel function $\mathcal{N}(\mu = \mathsf{t_e}, \sigma^2 = \mathsf{b}^2)$ to define a weight for each time point in the time series

$$w_{j,t_e} = \frac{1}{\sqrt{2\pi \mathsf{b}^2}} \exp\left\{-\frac{(\mathsf{j}-\mathsf{t_e})^2}{2\mathsf{b}^2}\right\}, \qquad (5)$$

where $j \in \{1, 2, ..., n\}$, which is the local constant or Nadaraya-Watson estimator (Fan & Gijbels, 1996).

For the example shown in Figure 3 this means that the time point t_e = 3 gets the highest weight, and if the distance to t_e increases, the weight becomes exponentially smaller. The same idea is represented in the data matrix in the right panel of Figure 3: each time point in the multivariate time series is associated with a weight defined by the kernel function. The smaller we choose the bandwidth b of the kernel function, the lower the number of observations we combine in order to estimate the model at t_e: when using a kernel with bandwidth b = 0.2 (red curve), we combine more observations than when using the kernel with b = 0.05 (blue curve). The smaller the bandwidth the larger the sensitivity to detect changes in parameters over time. However, a small bandwidth means that less data is used and therefore the estimates are less reliable (e.g., only three time points when b = 0.05; see right panel of Figure 3).

Since we combine observations close in time to be able to estimate a local model, we have to assume that the models close in time are also similar. This is equivalent to assuming that the true time-varying parameter functions are smooth, or locally stationary. Thus, the key assumption of the kernel-smoothing approach is the same as in the spline approach. For the kernel-smoothing method, we need the additional assumption that the chosen bandwidth is small enough to capture the time-varying nature of the true model. For example, if the parameters of the true model vary widely over time, but the bandwidth is so large that at any estimation point almost the entire time series is used for estimation, it is impossible to recover the true time-varying function.

The weights w_{j,t_e} defined in (5) enter the loss function of the ℓ_1-regularized regression problem we use to estimate each of the p time-varying versions of the model in (2)

$$\hat{\boldsymbol{\beta}}_{t_e} = \arg_{\boldsymbol{\beta}_{t_e}, \beta_{0,t_e}} \min\left\{\frac{1}{\mathsf{n}}\sum_{\mathsf{j}=2}^{\mathsf{n}} \mathsf{w}_{\mathsf{j},\mathsf{t_e}}(\mathsf{X}_{\mathsf{i},\mathsf{j}} - \beta_{0,\mathsf{t_e}} - \boldsymbol{\beta}_{t_e}\boldsymbol{X}_{j-1})^2 + \lambda_{\mathsf{i}}||\boldsymbol{\beta}_{\mathsf{t}}||_1\right\}, \qquad (6)$$

where $X_{i,j}$ is the j^{th} time point of the i^{th} variable in the design matrix, $||\boldsymbol{\beta}_{t_e}||_1 = \sum_{i=1}^{p} \sqrt{\beta_{i,t_e}^2}$ is the ℓ_1-norm of $\boldsymbol{\beta}_{t_e}$, and λ_i is a parameter controlling the strength of the penalty. Note that the indices i and t_e are fixed in (6) because we estimate the time-varying VAR model equation by equation, separately for each estimation point t_e.

For each of the p regressions, we select the λ_i that minimizes the out-of-sample deviance in 10-fold cross validation (Friedman et al., 2010). In order to select an appropriate bandwidth b, we choose the $\hat{b}$ that minimizes the out of sample deviance *across* the p regressions in a time stratified cross validation scheme (for details see Section 3.1.2). We choose a constant bandwidth for all regressions so we have a constant bandwidth for estimating the whole VAR model. Otherwise the sensitivity to detect time-varying parameters and the trade-off between false positives and false negatives differs between parameters, which is undesirable.

In ℓ_1-penalized (LASSO) regression the squared loss is minimized together with the ℓ_1-norm of the parameter vector. This leads to a trade-off between fitting the data (minimizing squared loss) and keeping the size of the fitted parameters small (minimizing ℓ_1-norm). Minimizing both together leads to small estimates being set to exactly zero, which is convenient for interpretation. When using ℓ_1-penalized regression, we assume that the true model is sparse, which means that only a small number of parameters in the true model are nonzero. If this assumption is violated, the largest true parameters will still be present, but small true parameters will be incorrectly set to zero. However, if we keep the number of parameters constant and let $n \to \infty$, ℓ_1-regularized regression also recovers the true model if the true model is not sparse. For an excellent treatment on ℓ_1-regularized regression see Hastie et al. (2015).

As noted above, the larger the bandwidth b, the more data is used to estimate the model around a particular estimation point. Indeed, the data used for estimation is proportional to the area under the kernel function or the sum of the weights $N_{\text{util}} = \sum_{j=1}^{n} w_{j,t_e}$. Notice that N_{util} is smaller at the beginning and end of the time series than in the center, because the kernel function is truncated. This necessarily leads to a smaller sensitivity to detect effects at the beginning and the end of the time series. For a more detailed description of the kernel smoothing approach see Haslbeck and Waldorp (2018b). In the remainder of this manuscript we refer to this method as KS(L1). With GLM(L1) we refer to the stationary ℓ_1-penalized estimator.

2.4. Related methods

Several implementations of related models are available as free software packages. The R-package *earlywarnings* (Dakos & Lahti, 2013) implements the estimation of a time-varying AR model using a moving window approach. The R-package *MARSS* (Holmes et al., 2012; 2013) implements the estimation of (time-varying) state-space models, of which the time-varying VAR model is a special case. While the state-space model framework is very powerful due to its generality, it requires the user to specify the way parameters are allowed to vary over time, for which often no prior theory exists in practice (Belsley & Kuti, 1973; Tarvainen et al., 2004). In parallel efforts Casas and Fernandez-Casal (2018) developed the R-package *tvReg*, which estimates time-varying AR and VAR models, as well as IRF, LM and SURE models, using kernel smoothing similar to the kernel smoothing approach described in the present paper, however does not offer ℓ_1-regularization. Furthermore, the R-package *bvarsv* (Krueger, 2015) allows one to estimate time-varying VAR models in a Bayesian framework.

The R-package *dynr* (Ou et al., 2019) provides an implementation for estimating regime switching discrete time VAR models, and the R-package *tsDyn* (Fabio Di Narzo et al., 2009) allows to estimate the regime switching Threshold VAR model (Hamaker et al., 2010; Tong & Lim, 1980). These two methods estimate time-varying models that switch between piece-wise constant regimes, which is different to the methods presented in this paper, which assume that parameters change smoothly over time.

Another interesting way to modeling time-varying parameters is by using the fused lasso (Hastie et al., 2015). However, to our best knowledge this method is currently only implemented for the estimation of Gaussian Graphical Models: Monti (2014) provide a Python implementation of the SINGLE algorithm (Monti et al., 2014), and (Gibbert, 2017) provides a Python implementation of the (group) fused-lasso based method as presented in Gibberd and Nelson (2017).

3. Evaluating performance via simulation

In this section we use two simulation studies to evaluate the performance of the above introduced methods in scenarios that are typical in applied research. In the first simulation (Section 3.1) we generate time-varying VAR models based on a random graph with fixed sparsity, which is a natural choice in the absence of any knowledge about the structure of VAR models in a given application. This simulation allows us to

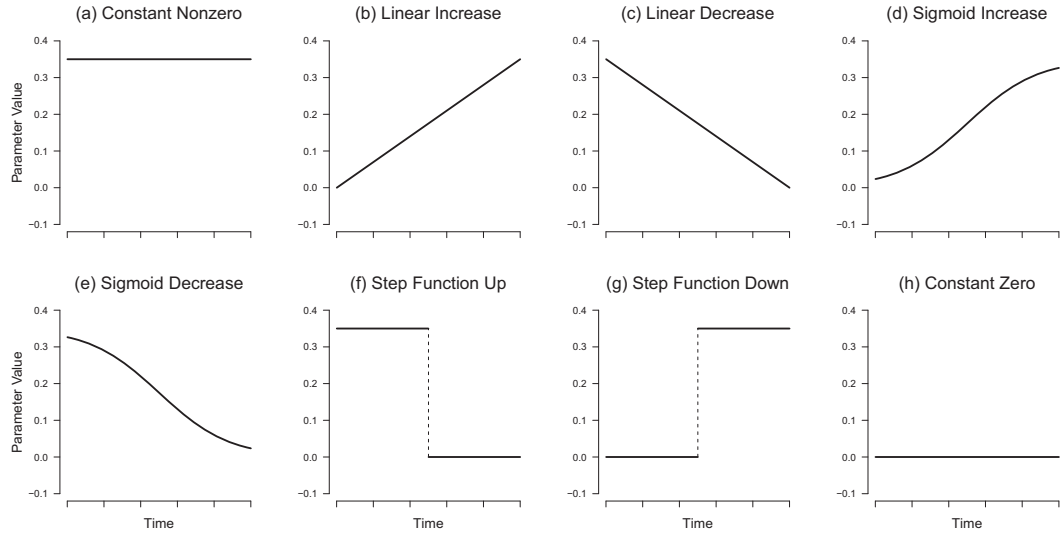

Figure 4. The eight types of time-varying parameters used in the simulation study: (a) constant nonzero, (b) linear increase, (c) linear decrease, (d) sigmoid increase, (e) sigmoid decrease, (f) step function up, (g) step function down and (h) constant zero.

get a rough overview of the performance of all methods and their strengths and weaknesses. In the second simulation (Section 3.2), we generate time-varying VAR models in which we vary the level of sparsity. This simulation allows us to discuss the strengths and weaknesses of all methods in more detail, specifically, we can discuss in which situations methods with/without regularization or thresholding perform better. Finally, in Section 3.3 we discuss the combined results of both simulations, and provide recommendations for applied researchers.

3.1. Simulation A: random graph

In this simulation we evaluate the performance of all methods in estimating time-varying VAR models that are generated based on a random graph. We first describe how we generate these time-varying VAR models (Section 3.1.1), discuss details about the estimation methods (Section 3.1.2), report the results (Section 3.1.3), and provide a preliminary discussion (Section 3.1.4).

3.1.1. Data generation

We generated time-varying VAR models by first selecting the structure of a stationary VAR model and then turning this stationary VAR model into a time-varying one. Specifically, we used the following procedure to specify whether a parameter in the time-varying VAR(1) model is nonzero: we choose all our VAR models to have $p = 10$ variables, which is roughly the number of variables measured in typical ESM studies. We start out with an empty $p \times p$ VAR parameter matrix. In this matrix we set all p autocorrelations to be nonzero, since autocorrelations are expected to be present for most phenomena and are observed in essentially any application (e.g., aan het Rot, Hogenelst, & Schoevers, 2012; Snippe et al., 2017; Wigman et al., 2015). Next, we randomly set 26 of the $p \times p - p = 90$ off-diagonal elements (the cross-lagged effects) to be present. This corresponds to an edge probability of $P(\text{edge}) \approx 0.29$.[2] This approach returns an initial $p \times p$ matrix with ones in the diagonal and zeros and ones in the off-diagonal.

In a second step we use the structure of this VAR model to generate a time-varying VAR model. Specifically, we randomly assign to each of the nonzero parameters one of the sequences (a)–(g) in Figure 4. If an edge is absent in the initial matrix, all entries of the parameter sequence are set to zero (panel (h) in Figure 4). Note that only the time-varying parameter functions (a–e) and (h) in Figure 4 are smooth functions of time. Therefore, the two methods presented in this paper are only consistent estimators for those types of time-varying parameters. They cannot be consistent estimators for the step-functions (f) and (g), however, we included them to investigate how closely the methods studied in this paper can approximate the step function.

The maximum parameter size of time-varying parameters is set to $\theta = 0.35$ (see Figure 4). The noise is drawn from a multivariate Gaussian with variances

[2]We set a fixed number of elements to nonzero instead of using draws with $P(\text{edge}) = 0.2$, because we resample the VAR matrix until it represents a stable VAR model (the absolute value of all eigenvalues is smaller than 1). By fixing the number of nonzero elements we avoid biasing $P(\text{edge})$ through this resampling process. Thus, none of the VAR matrices in any iteration and at any time point has an eigenvalue with absolute value greater than 1.

$\sigma^2 = \sqrt{0.10}$ and all covariances being equal to zero. Hence the signal/noise ratio used in our setup is $\mathrm{S/N} = \frac{0.35}{0.10} = 3.50$. All intercepts are set to zero and the covariances between the noise processes assigned to each variable are zero.

Using these time-varying VAR model, we generate 12 independent time series with lengths $n = \{20, 30, 36, 69, 103, 155, 234, 352, 530, 798, 1201, 1808\}$. We chose these values because they cover the large majority of scenarios applied researchers typically encounter. Each of these time-varying models covers the full time period $[0, 1]$ and is parameterized by a $p \times p \times n$ parameter array $B_{i,j,t}$. For example, the $B_{1,2,310}$ indicates the cross-lagged effect from variable 2 on variable 1 at the 310th measurement point, which occurs at time point $310/530 \approx 0.59$, if there are in total 530 measurements. Importantly, in this setting increasing n does not mean that the time period between the first and the last measurement of the time series becomes larger. Instead, we mean by a larger n that more evenly spaced measurements are available in the same time period. This means that the larger n, the smaller the time interval between two adjacent measurements. That is, the data density in the measured time period increases with n, which is required to consistently estimate time-varying parameters (Robinson, 1989). This makes sense intuitively: if the goal is to estimate the time-varying parameters of an individual in January, then one needs sufficient measurements in January, and it does not help to add additional measurements from February.

We run 100 iterations of this design and report the mean absolute error over iterations. These mean errors serve as an approximation of the expected population level errors.

3.1.2. Estimation

To estimate time-varying VAR models via the GAM method we use the implementation in the R-package *tvvarGAM* (Bringmann et al., 2017) version 0.1.0, which is a wrapper around the *mgcv* package (version 1.8-22). The tuning parameter of the spline method is the number of basis functions used in the GAM. Previous simulations have shown that 10 basis functions give good estimates of time-varying parameters (Bringmann et al., 2018). To ensure that the model is identified, for a given number of basis functions k and variables p, we require at least $n_{\min} > k(p + 1)$ observations. In our simulation, we used this constraint to select the maximum number of basis functions possible given n and p, but we do not use less than 3 or more than 10 basis functions. That is, the selected number of basis functions k_s is defined as

$$k_s = \max\left\{3, \min\left\{\max\left\{\mathsf{k}; \mathsf{k} > \frac{\mathsf{n}}{\mathsf{p}+1}\right\}, 10\right\}\right\}. \quad (7)$$

If k_s satisfies the above constraint, the time-varying VAR model can be estimated with the spline-based method. With this constraint the model cannot be estimated for $n = \{20, 30\}$. We therefore do not report results for GAM and GAM(st) for these sample sizes.

In principle it would be possible to combine ℓ_1-regularization with the GAM-method, similarly as in the KS-method. However, an implementation of such a method is currently not available and we therefore cannot include it in our simulation.

We estimated the time-varying VAR model via the KS and KS(L1) methods using the R-package *mgm* (Haslbeck & Waldorp, 2018b) version 1.2-2. As discussed in Section 2.3, these methods require the specification of a bandwidth parameter. Therefore, the first step of applying these methods is to select an appropriate bandwidth parameter by searching the candidate sequence $\boldsymbol{b} = \{0.01, 0.045, 0.08, 0.115, 0.185, 0.22, 0.225, 0.29, 0.325, 0.430, 0.465, 0.5\}$. For $n \leq 69$ we omit the first 5 values in $\boldsymbol{b}$, and for $n > 69$ we omit the last 5 values. We did this to save computational cost because for small n, small b are never selected, and analogously for large n, large b values are never selected. To select an appropriate bandwidth parameter we use a cross-validation-like scheme, which repeatedly divides the time series in a training and a test set, and in each repetition fits time-varying VAR models using the bandwidths in b, and evaluates the prediction error on the test set for each bandwidth. More concretely, we define a test set S_{test} by selecting $|S_{\text{test}}| = \lceil (0.2\mathsf{n})^{2/3} \rceil$ time points stratified equally across the whole time series. Next, we estimate a time-varying VAR model for each variable p at each time point in S_{test} and predict the p values at that time point. After that we compute for each b the $|S_{\text{test}}| \times p$ absolute prediction errors and take the arithmetic mean. Next, we select the bandwidth $\hat{b}$ that minimizes this mean prediction error. Finally, we estimate the model on the full data using $\hat{b}$ and $\hat{\lambda}$ at 20 equally spaced time points, where we select an appropriate penalty parameter $\hat{\lambda}_i$ with 10-fold cross-validation for each of the p variables (for more details see Haslbeck & Waldorp, 2018b).

We also investigate the performance of the kernel-smoothing method without ℓ_1-regularization. We refer to this method as KS. In order to compare the ℓ_1-regularized time-varying VAR estimator to a stationary ℓ_1-regularized VAR estimator, we also

estimate the latter using the *mgm* package. We call this estimator GLM(L1).

Both time-varying estimation methods are consistent if the following assumptions are met; (a) the data is generated by a time-varying VAR model as specified in Equation (2), (b) all parameters are smooth functions of time, (c) with the eigenvalues of the VAR matrix being within the unit circle at all time points, (d) and the error covariance matrix is diagonal. We also fit a standard stationary VAR model using linear regression to get the unbiased stationary counter-part of the GAM methods. Specifically for the KS-method, it is additionally required that we consider small enough candidate bandwidth values. We do this by using the sequence $\boldsymbol{b}$ specified above.

3.1.3. Results

We first report the performance of the GLM, GLM(L1), KS, KS(L1), GAM and GAM(st) methods in estimating different time-varying parameters by evaluating the estimation error *averaged across time*. Next, we zoom in on the performance *across time*, for the constant and the linear increasing parameter function, and finally examine the performance in structure recovery of all methods.

3.1.3.1. Absolute error averaged over time. Figure 5 displays the absolute estimation error, averaged over time points, iterations, and time-varying parameter functions of the same type, as a function of sample size n. Since the linear increase/decrease, sigmoid increase/decrease, and step function increase/decrease are symmetric, we collapsed them into single categories to report estimation error. The absolute error on the y-axis can be interpreted as follows: let's say we are in the scenario with $n = 155$ observations and estimate the constant function in Figure 5 (a) with the stationary ℓ_1-regularized regression GLM(L1). Then the *expected* average (across the time series) error of the constant function is ±0.09.

Figure 5 (a) shows that, for all methods, the absolute error in estimating the constant nonzero function is large for small n and seems to converge to zero as n increases. The GLM method has a lower estimation error than its ℓ_1-regularized counterpart, GLM(L1). Similarly, the KS method outperforms the KS(L1) method. The stationary GLM method also outperforms all time-varying methods, which makes sense because the true parameter function is not time-varying.

For the linearly increasing/decreasing time-varying parameter in Figure 5 (b), the picture is more complex. For very small n from 20 to 46 the regularized methods GLM(L1) and KS(L1) perform best. This makes sense because, for such small n, the estimates of all other methods suffer from huge variance. For sample sizes from 46 to 155 the unregularized methods perform better: now the bias of the regularized methods outweighs the reduction in variance. From sample sizes between 155 and 352 the time-varying methods start to outperform the two stationary methods. Interestingly, until around $n = 530$ the KS methods outperforms all other time-varying methods. For $n > 530$ all time-varying methods perform roughly equally. Overall, the error of all time-varying methods seem to converge to zero, as we would expect from a consistent estimator. The error of the stationary methods converges to ≈0.088 which is the error resulting from approximating the time-varying function with the optimal constant function $y(t) = \frac{0.35}{2}$. Since the sigmoid increase/decrease functions in panel (c) are very similar to the linear increase/decrease functions, we obtain qualitatively the same results as in the linear case.

In the case of the step function we again see a similar qualitative picture, however here the time-varying methods outperform the stationary methods already at a sample size of around $n = 69$. The reason is that the step function is more time-varying in the sense that here the best constant function is a worse approximation than in the linear and the sigmoid case. Another difference is that the GAM(st) method seems to outperform all other methods by a small margin if the sample size is large.

Finally, the absolute error for estimating the constant zero function is lowest for the regularized methods and the thresholded GAM method. This is what one expect since these methods bias estimates toward zero, and the true parameter function is zero across the whole time period.

In Figure 5 we reported the mean population errors of the six compared methods in various scenarios. These mean errors allow one to judge whether the *expected* error of one method will be larger than the one of another method. However, it is also interesting to inspect the population sampling variance around these mean errors. This allows one to gauge with which probability one method will be better than another for a given sample. We show a version of Figure 5 that includes the 25% and 95% quantiles of the absolute error in Appendix A.

3.1.3.2. Absolute error over time for constant and linear increasing function. To investigate the behavior of the different methods in estimating parameters across the time interval, Figure 6 displays the mean absolute error for each estimation point (spanning the

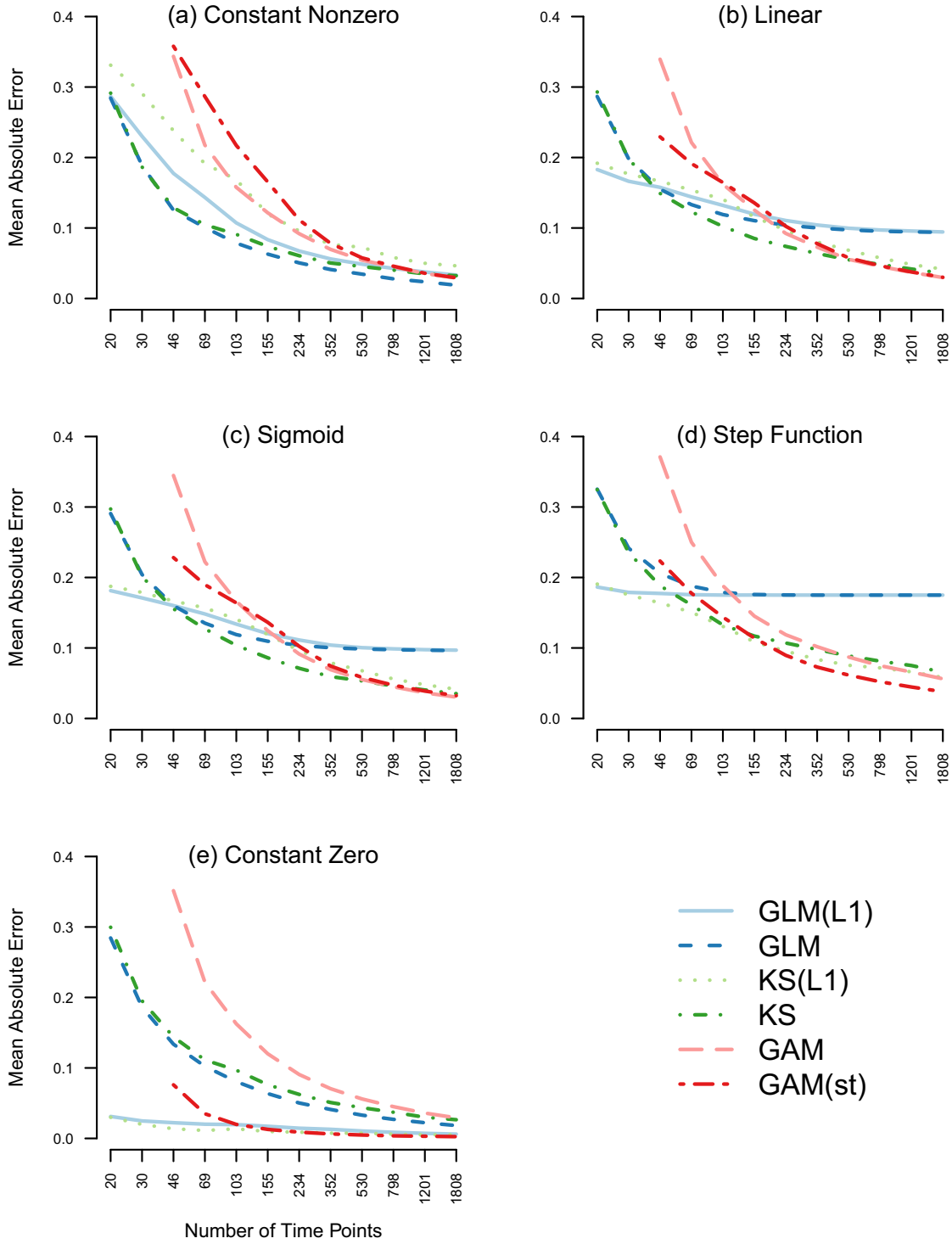

Figure 5. The five panels show the mean absolute estimation error averaged over the same type, time points, and iterations as a function of the number of observations n on a log scale. We report the error of six estimation methods: stationary unregularized regression (blue), stationary ℓ_1-regularized regression (light blue), time-varying regression via kernel-smoothing (green), time-varying ℓ_1-regularized regression via kernel-smoothing (light green), time-varying regression via GAM (pink), and time-varying regression via GAM with thresholding at 95% CI (red). Some data points are missing because the respective models are not identified in that situation (see Section 3.1.2).

full period of the time series) for the constant nonzero function and the linear increasing function for $n = \{103, 530, 1803\}$. Note that these results were already shown in aggregate form in Figure 5: for instance, the average (across time) of estimates of the stationary ℓ_1-regularized method in Figure 6 (a) corresponds to the single data point in Figure 5 (a) of the same method at $n = 103$.

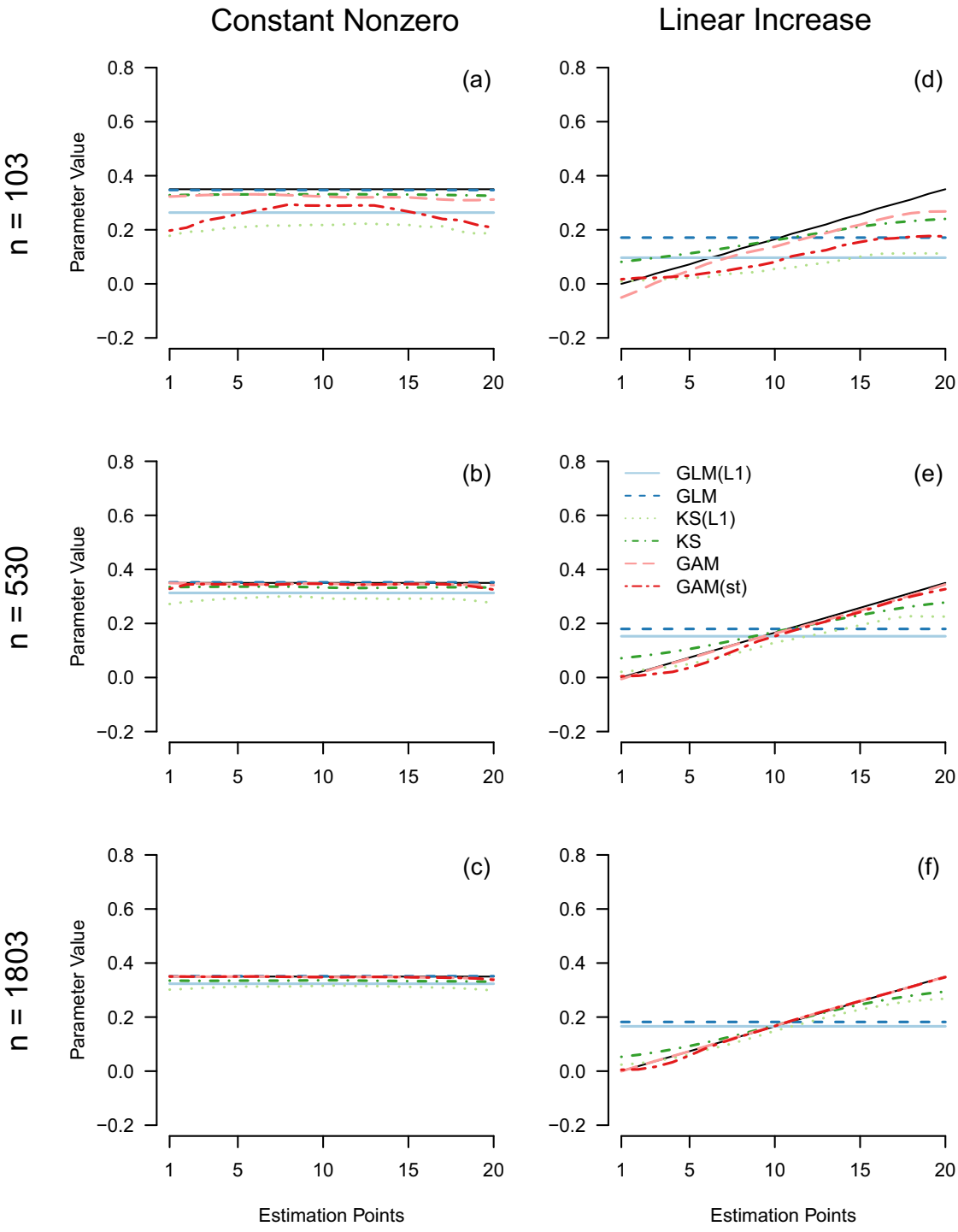

Figure 6. Mean and standard deviations of estimates for the constant parameter (left column), and the linear increasing parameter (right column), for $n = 103$ (top row), $n = 530$ (second row) and $n = 1803$ (bottom row) averaged over iterations, separately for the five estimation methods: stationary ℓ_1-regularized regression (red), unregularized regression (blue), time-varying ℓ_1-regularized regression via kernel-smoothing (green), time-varying regression via GAM (pink), and time-varying regression via GAM with thresholding at 95% CI (orange).

Panel (a) of Figure 6 shows the average parameter estimates of each method for the constant function with $n = 103$ observations. In line with the aggregate results in Figure 5, the stationary methods outperform the time-varying methods, and the unregularized methods outperform the regularized methods. We also see that the KS(L1) and the GAM(st) methods are biased downwards at the beginning and the end of the time series. The reason is that less data is available at these points, which results in stronger bias toward zero (KS(L1)) and more estimates being thresholded to zero. When increasing n, all methods become

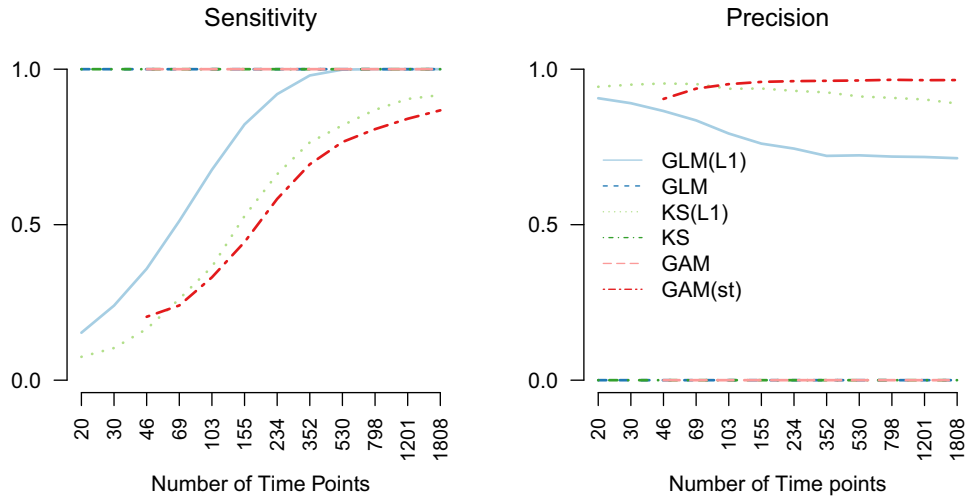

Figure 7. Sensitivity and precision for the five estimation methods across all edge-types for different variations of *n*. The lines for the unthresholded GAM(st) method and the stationary GAM method overlap completely, since they do not return estimates that are exactly zero. Some data points are missing because the respective models are not identified in that situation (see Section 3.1.2).

better at approximating the constant nonzero function. This is what we would expect from the results in Figure 5, which suggested that the absolute error of all methods converges to zero as *n* grows.

In the case of the linear increase with $n = 103$ (d), we see that the time-varying methods follow the form of the true time-varying parameter, however, some deviations exists. With larger *n*, the time-varying methods recover the linearly increasing time-varying parameter with increasing accuracy. In contrast, the stationary methods converge to the best-fitting constant function. We also see that the average estimates of the regularized methods are closer to zero than the estimates of the unregularized methods. However, note that, similar to panel (e) in Figure 5, the regularized methods would perform better in recovering the constant zero function.

Here we only presented the mean estimates of each method, which displays the bias of the different methods as a function of sample size. However, it is equally important to consider the variance around estimates. We display this variance in Figure 12 in Appendix B. This figure shows that — as expected — the variance is very large for small *n*, but approaches 0 when *n* becomes large.

3.1.3.3. Performance in structure recovery. In some situations the main interest may be to recover the *structure* of the VAR model, that is, we would like to know which parameters in the VAR parameter matrix are nonzero. We use two measures to quantify the performance of structure recovery. Sensitivity, the probability that a parameter that is nonzero in the true model is estimated to be nonzero; and precision, the probability that a nonzero estimate is nonzero in the true model. While higher values are better for both sensitivity and precision, different estimation algorithms typically offer different trade-offs between the two. Figure 7 shows this trade-off for the five estimation methods:

The unregularized stationary GLM method, the unregularized KS method, and the unthresholded time-varying GAM method have a sensitivity of 1 and a precision of 0 for all *n*. This is trivially the case because these methods return nonzero estimates with probability 1, which leads to a sensitivity of 1 and a precision of 0. Consequently, these methods are unsuitable for structure estimation. For all remaining methods, sensitivity seems to approach 1 when increasing *n*, while GLM(L1) has the highest sensitivity followed by KS(L1) and GAM(st). As expected, the precision of these methods is stacked up in reverse.

3.1.4. Discussion

The first simulation showed how the different methods perform in recovering a VAR model with $p = 10$ variables based on a random graph, with linear, sigmoid, step and constant parameter functions, with sample sizes that cover most applications in psychology. The compared methods differ in the dimensions stationary vs. time-varying methods, unregularized vs. regularized methods, and GAM- vs. KS-based methods. Since all these dimensions interact with each other and with the type of time-varying

parameter function they aim to recover, we discuss these interactions separately for each parameter function.

3.1.4.1. Constant nonzero function. In the case of the constant nonzero function the stationary and unregularized GLM performed best, followed by the unregularized time-varying KS method. It makes sense that GLM performs best, because the true parameter function in this case is nonzero and constant across time. The KS method performs similarly especially for small n, because the bandwidth selection will select a very high bandwidth, which leads to a weighting that is almost equal for all time points, which leads to estimates that are very similar to the ones of the GLM method. The next best method is the stationary regularized GLM(L1) method. This is because the regularization decreases performance if the true parameter function is nonzero, however, it uses the correct assumption that the true parameter function is constant across time. Since the ability to estimate time-varying parameters is no advantage when estimating the constant nonzero function, the KS(L1) method performs worse than the GLM(L1) method. Interestingly, the unregularized GAM function performs much worse than the unregularized KS method. The significance-thresholded GAM(st) method performs worse than the GAM method, because if the true parameter function is nonzero, thresholding it to zero can only increase estimation error.

3.1.4.2. Linear and sigmoid functions. The results for the linear increasing/decreasing function are similar to the constant nonzero function, except that all time-varying methods have a lower absolute error than the stationary methods from $n > 234$. The KS method is already better from $n > 46$. A difference to the constant nonzero function is that the two regularized methods GLM(L1) and KS(L1) perform best if the sample size is very small ($n < 46$). A possible explanation for this difference is that the bias toward zero of the regularization is less disadvantageous for the linear increasing/decreasing functions, because its parameter values are on average only half as large as for the constant nonzero function. Within time-varying functions, the KS method performs better than the KS(L1) methods, which makes sense because the true parameter function is nonzero. For the same reason, the GLM method outperforms the GAM(st) method. The KS methods perform better than the GAM methods for sample sizes up to n=530. The reason is that the estimates of the GAM methods have a larger sampling variance (see Figure 11 in Appendix A). The errors in estimating the sigmoid function are very similar to the linear increasing/decreasing functions, since their functional forms are very similar.

3.1.4.3. Step function. The errors in estimating the step function are again similar to the linear and the sigmoid case, except for two differences: first, the time-varying methods become better than the stationary methods already between $n = 46$ and $n = 69$. And second, the regularized KS(L1) performs better than KS, and the thresholded GAM(st) method performs better than the GAM method. The reason is that in half of the time series the parameter value is zero, which can be recovered exactly with the KS(L1) and the GAM(st) methods. This advantage seems to outweigh the bias these methods have in the other half of the time series in which the parameter function is nonzero.

3.1.4.4. Constant zero function. In the case of the constant zero function the errors are roughly stacked up the reverse order as in the constant nonzero function. The regularized GLM(L1) and KS(L1) do best, followed by the thresholded GAM(st) method. Among the unregularized methods the GLM and KS methods perform quite similarly, with the GLM method being slightly better, because the true parameter function is constant. Interestingly, the GAM method performs far worse, which is again due to its high variance (see Figure 11 in Appendix A).

3.1.4.5. Summary. We saw that stationary methods outperform time-varying methods when the true parameter function is a constant, and time-varying methods out-perform stationary methods if the true parameter function is time-varying, and if the sample size is large enough. The sample size at which the time-varying methods become better depends on how time-varying the true parameter is: the more time-varying it is, the smaller the sample size n at which time-varying methods become better than stationary ones. Within time-varying methods, the KS methods outperformed the GAM methods for smaller sample sizes, while the GAM based methods became better with very large sample sizes ($n > 530$).

Finally, we saw that regularized methods perform better if the true parameter function is zero, while unregularized methods perform better if the true parameter function is nonzero, as expected. In order to choose between regularized and unregularized methods, one therefore needs to judge how many of the parameters in the true time-varying VAR model

$$
\begin{array}{c} \\ X_{1,t} \\ X_{2,t} \\ X_{3,t} \\ X_{4,t} \\ X_{5,t} \\ X_{6,t} \end{array}
\begin{array}{c}
\begin{array}{cccccc} X_{1,t-1} & X_{2,t-1} & X_{3,t-1} & X_{4,t-1} & X_{5,t-1} & X_{6,t-1} \end{array} \\
\left(\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array} \right)
\end{array}
$$

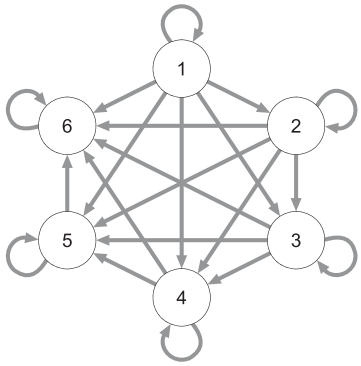

Figure 8. Left: the upper-diagonal pattern of nonzero parameters used in the time-varying VAR model in the second simulation, here shown for six variables. The row sums are equal to the indegree of the respective nodes, which results in a frequency of one for each indegree value. Right: visualization of the upper-diagonal pattern as a directed graph. The graph used in the simulation has the same structure but is comprised of 20 nodes.

are nonzero. Given the expected sparsity of the true VAR model, one could compute a weighted average of the errors shown in this section in order to determine which method has the lowest overall error. However, to evaluate the performance of the different methods for different levels of sparsity more directly, we performed a second simulation study in which we vary the sparsity of the VAR model.

3.2. Simulation B: varying sparsity

In this simulation we evaluate the absolute estimation error of all methods for the different parameter functions and for the combined time-varying VAR model, as a function of sparsity. Specifically, we evaluate the estimation error of recovering the time-varying predictors of a given variable in the VAR model, depending on how many predictors are nonzero. From a network perspective the number of predictors on a given node is equal to its indegree. We will vary the indegree from 1 to 20. The average indegree in Simulation A was $1 + 9 \times P(\text{edge}) = 2.61$.

3.2.1. Data generation

We vary sparsity by specifying the structure of the initial VAR matrix to be upper-triangular. We show the structure of such a matrix, and the corresponding directed network in Figure 8.

In such a model, the first variable has one predictor (itself at $t-1$), the second variables has two predictors (itself and variable 1 at $t-1$), the third variable has three predictors, etc. and the last variable has p predictors. As defined in Section 2, the number of nonzero predictor variables (or the indegree from a network perspective) is a local (i.e. for some variable X) measure of sparsity. In the simulation we use the same initial VAR matrix, except that we use a VAR model with $p = 20$ variables. All additional steps of the data generation (Section 3.1.1), and the specification of the estimation methods (Section 3.1.2) are the same as in Simulation A.

3.2.2. Results

Figure 9 displays the mean absolute error separately for the five different time-varying parameter functions and for indegrees 1, 10, 20. Similarly to Simulation A, we collapsed symmetric increasing and decreasing functions into single categories and report their average performance. The first row of Figure 9 shows the performance averaged over time points and types of time-varying parameters for indegree 1, 10 and 20. The most obvious result is that all methods become worse when increasing the indegree. This is what one would expect since more parameters are nonzero and more predictors are correlated. In addition, there are several interactions between indegree and estimation methods. First, the regularized methods perform best when indegree is low, and worst when indegree is high. This makes sense: the bias toward zero of the regularization is more beneficial if almost all parameter functions are zero. However, if most parameter functions are nonzero, a bias toward zero leads to high estimation error. Second, we see that the drop in performance is lower for the GAM based methods compared to the KS based methods. The combined results in the first row are the weighted average of the remaining rows. The estimation errors for the time-varying functions show a similar pattern as in Figure 5 of Simulation A, except that the GAM methods perform better for indegree values 10 and 20.

3.2.3. Discussion

The results of Simulation B depicted the relative performance of all methods as a function of sparsity, which

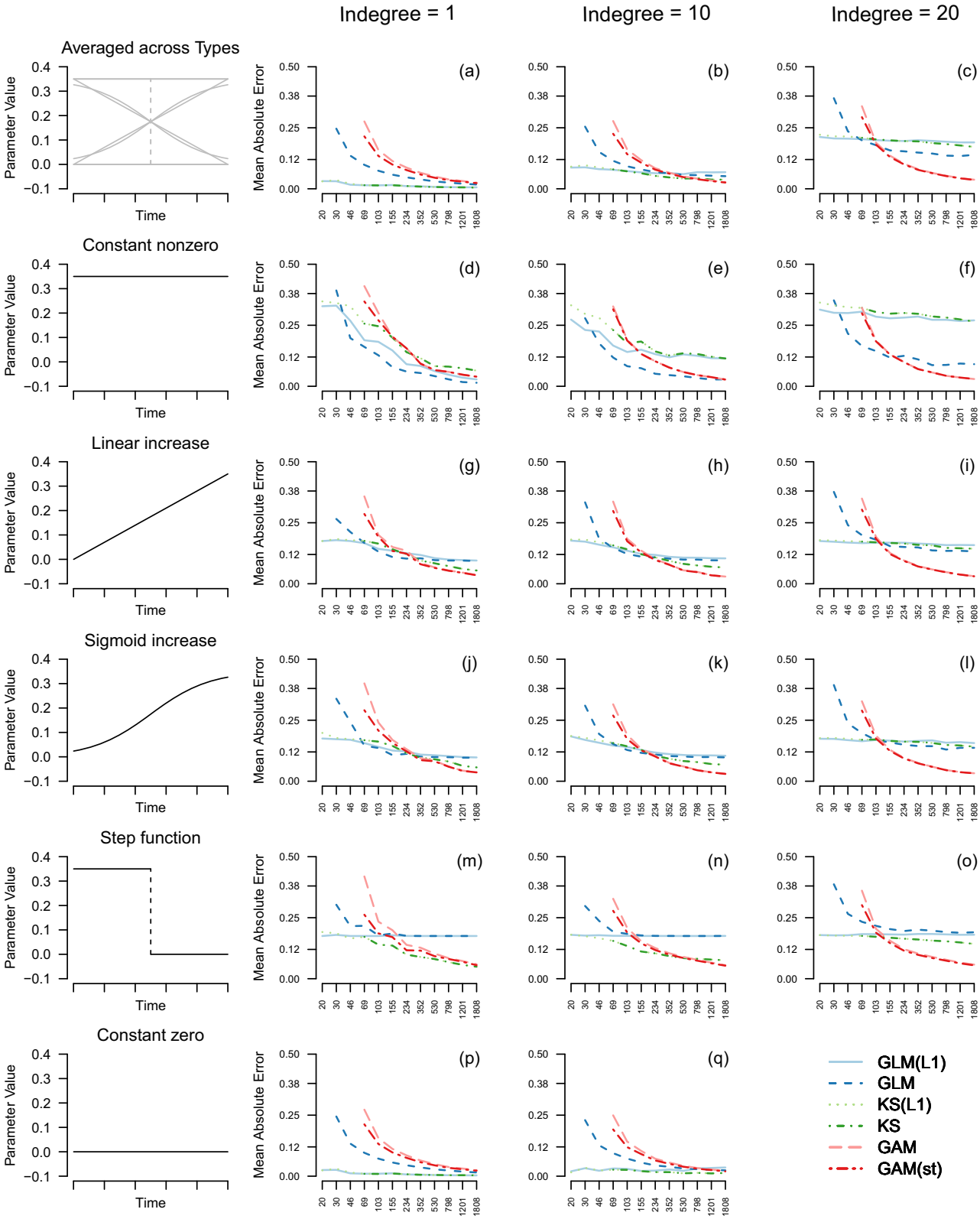

Figure 9. The mean average error for estimates of the upper-triangular model for all five estimation methods for the same sequence of numbers of time points *n* as in the first simulation. The results are conditioned on three different indegrees (1, 10, 20) and shown averaged across (a–c) and separately for the time-varying parameter types (d–q).

we analyzed locally as indegree. As expected, regularized methods perform better when indegree is low and worse if indegree is high. Interestingly, among the time-varying methods, the GAM based methods perform better than the KS based methods when indegree is high.

3.3. Overall discussion of simulation results

Here we discuss the overall strengths and weaknesses of all considered methods in light of the results of both simulations.

3.3.1. Stationary vs. time-varying methods. We saw that stationary methods outperform time-varying methods if the true parameter function is constant, as one would expect. If the parameter function is time-varying, then the stationary methods are better for very small sample sizes, but for larger sample sizes, the time-varying methods become better. The exact sample size n at which time-varying methods start to perform better depends on how strongly the true parameters vary with time: the stronger the variation, the smaller the n. For the choice of true parameter functions in our simulations, we found that the best time-varying method outperformed the stationary methods at already $n > 46$.

3.3.2. Unregularized vs. regularized methods. The results in both simulations showed that if most true parameter functions are zero (high sparsity), regularized methods and the thresholded GAM(st) method performed better compared to their unregularized/unthresholded counterparts. On the other hand, if most true parameter functions are nonzero (low sparsity), the unregularized/unthresholded functions perform better. In Simulation B we specifically mapped out the performance of methods as a function of sparsity and found that unregularized methods are better at an indegree of 10 or larger.

3.3.3. Kernel-smoothing vs. GAM methods. If sparsity is high, that is, if most parameter functions are zero, the KS based methods outperformed the GAM based methods for most sample size regimes. Only if the sample size is very large the GAM based methods showed a performance that is equal or slightly better than the KS based methods. However, if sparsity is low, the GAM based methods outperformed the KS based methods.

Accordingly, applied researchers should choose the KS based methods when they expect the time-varying VAR model to be relatively sparse and if they only have a moderate sample size ($n < 200$–300). If one expects that only few parameter functions are nonzero, the KS based method should be combined with regularization. If one expects the parameter functions of the time-varying VAR model to be largely nonzero, and if one has a large sample size, the GAM based methods are likely to perform better.

3.3.4. Limitations. Several limitations of the simulation studies require discussion. First, the signal to noise ratio $\mathrm{S/N} = \frac{\theta}{\sigma} = 3.5$ in parameter values could be larger or smaller in a given application and the performance results would accordingly be better or worse. Similarly, the signal to noise ratio would be smaller if we increased the number of variables p, because more parameters have to be estimated. However, note that S/N is also a function of n. Hence if we assume a lower S/N this simply means that we need more observations to obtain the same performance, while all qualitative relationships between time-varying parameters, structure in the VAR model and estimators remain the same.

Second, the time-varying parameters could be more time-varying. For example, we could have chosen functions that increase and decrease multiple times instead of being monotone increasing/decreasing. However, for estimation purposes, the extent to which a function is time-varying is determined by how much it varies over a specified time period *relative* to how many observation are available in the time period. Thus the n-variations can also be seen as a variation of the extent to which parameters are varying over time: From this perspective, the time-varying parameter functions with $n = 20$ are very much varying over time, while the parameter functions with $n = 1808$ are hardly varying over time. Since we chose n-variations stretching from unacceptable performance ($n = 20$) to very high performance ($n = 1808$), we simultaneously varied the extent to which parameters are time-varying.

Third, we only investigated time-varying VAR models with $p = 10$ variables and a single lag. In terms of the performance in estimating (time-varying) VAR parameters, adding more variables or lags boils down to increasing the indegree of a VAR model with a single lag and fixed p. In general, the larger the indegree and the higher the correlations between the predictors, the harder it is to estimate the parameters associated with a variable. Part of the motivation for Simulation B in Section 3.2 was to address this limitation.

Finally, we would like to stress that all statements with respect to sample size refer to the effective sample size available to estimate the VAR model. We mention this because the effective sample size that is used to estimate a VAR model is often considerably lower than the number of measurement points in an ESM study. This is both because of planned (e.g., at the

day/night shift) and unplanned missing values. For example, if an ESM study has five measurements a day with a measurement interval of 3 h and the fourth measurement is missing, then the effective sample size is only three, because only for three time points (2, 3, and 4) a measurement 3 h before is available.

4. Estimating time-varying VAR model on mood time series

In this section we provide a step-by-step tutorial on how to estimate a time-varying VAR model on a mood time series using the KS(L1) method. In addition, we show how to compute time-varying prediction errors for all nodes, how to assess the reliability of all estimates, and how to visualize some aspects of the estimated time-varying VAR model. Finally, we briefly discuss how to select between stationary and time-varying models. All analyses are performed using the R-package *mgm* (version 1.2-8) (Haslbeck & Waldorp, 2018b) and R-version 3.6.0, and the code below can also be found as an R-file on Github: https://github.com/jmbh/tvvar_paper. In Appendix D we show how to fit the same model with the GAM(st) method using the R-package *tvvarGAM*.

4.1. Data

We illustrate how to fit a time-varying VAR model on a symptom time series with 12 variables related to mood measured on 1476 time points during 238 consecutive days from an individual diagnosed with major depression (Wichers et al., 2016). The measurements were taken at 10 pseudo-randomized time intervals with average length of 90 minutes between 07:30 and 22:30. During the measured time period, a double-blind medication dose reduction was carried out, consisting of a baseline period, the dose reduction, and two post assessment periods (See Figure 10, the points on the time line correspond to the two dose reductions). For a detailed description of this data set see Kossakowski et al. (2017).

4.2. Load R-packages and dataset

The above described symptom dataset automatically available when loading the R-package *mgm*. After loading the package, we subset the 12 mood variables contained in this dataset:

```
library(mgm) # Version 1.2-8

mood_data <-as.matrix(symptom_data$data[, 1:12]) # Subset variables
mood_labels <- symptom_data$colnames[1:12] # Subset variable labels
colnames(mood_data) <- mood_labels
time_data <- symptom_data$data_time
```

The object `mood_data` is a 1476×12 matrix with measurements of 12 mood variables:

```
> dim(mood_data)
[1] 1476   12

> head(mood_data[,1:7])
     Relaxed Down Irritated Satisfied Lonely Anxious Enthusiastic
[1,]       5   -1         1         5     -1      -1            4
[2,]       4    0         3         3      0       0            3
[3,]       4    0         2         3      0       0            4
[4,]       4    0         1         4      0       0            4
[5,]       4    0         2         4      0       0            4
[6,]       5    0         1         4      0       0            3
```

The matrix `time_data` contains information about the time stamps of each measurement. This information is needed for the data preprocessing in the next section.

```
> head(time_data)
date dayno beepno beeptime resptime_s resptime_e time_norm
1 13/08/12 226  1 08:58 08:58:56 09:00:15 0.000000000
2 14/08/12 227  5 14:32 14:32:09 14:33:25 0.005164874
3 14/08/12 227  6 16:17 16:17:13 16:23:16 0.005470574
4 14/08/12 227  8 18:04 18:04:10 18:06:29 0.005782097
5 14/08/12 227  9 20:57 20:58:23 21:00:18 0.006285774
6 14/08/12 227 10 21:54 21:54:15 21:56:05 0.006451726
```

For a sizable number of measurement points the individual did not provide a response. The *mgm* package takes care of this automatically, by only using those time points to estimate a VAR(1) model for which a measurement at the previous time point is available.

Some of the variables in this data set are highly skewed, which can lead to unreliable parameter estimates. Here we deal with this issue by computing bootstrapped confidence intervals (KS method) and credible intervals (GAM method), to judge how reliable the estimates are. Since the focus in this tutorial is on estimating time-varying VAR models, we do not investigate the skewness of variables in detail. However, in practice the marginal distributions should always be inspected before fitting a (time-varying) VAR model. A possible remedy for skewed variables is to transform them, typically by taking a root, the log, or transformations such as the nonparanormal transform (Liu et al., 2009). A disadvantage of this approach is that the parameters are more difficult to interpret. For example, if applying the log-transform to X, then the cross-lagged effect $\beta_{X,Y}$ of Y on X is interpreted as "When increasing Y at $t-1$ by 1 unit, the log of X at t increases by $\beta_{X,Y}$, when keeping all other variables at $t-1$ constant".

4.3. Estimating time-varying VAR model

Here we describe how to use the function `tvmvar()` of the *mgm* package to estimate a time-varying VAR model. A more detailed description of this function can be found in the help file `?tvmvar`. After providing the `data` via the data argument, we specify the type and levels of each variable. The latter is necessary because *mgm* allows one to estimate models including different types of variables. In the present case we only have continuous variables modeled as conditional Gaussian distributions, and we therefore specify `type = rep("g", 12)`. By convention the number of levels for continuous variables is specified as one `level = rep(1, 12)`.

Via the argument `estpoints` we specify that we would like to have 20 estimation points that are equally spaced across the time series (for details see `?tvmvar`). The number of estimation points can be chosen arbitrarily large, however at some point adding more estimation points becomes useless because adjacent estimation points become indistinguishable. Via the argument `timepoints` we provide a vector containing the time point of each measurement. The time points are used to distribute the estimation points correctly on the time interval. If no `time-points` argument is provided, the function assumes that all measurement points are equidistant. See Section 2.5 in Haslbeck and Waldorp (2018b) for a more detailed explanation how the time points are used in *mgm* and an illustration of the problems following from incorrectly assuming equidistant measurement points.

Next, we specify the bandwidth parameter b, which determines how many observations close to an estimation point are used to estimate the model at that point. Here we select $b = 0.34$, which we obtained by searching a candidate sequence of bandwidth parameters, and selected the value that minimized the out-of-bag cross-validation error. The latter is implemented in the function `bwSelect()` (for details on this time-stratified cross-validation scheme see Section 3.1.2). Since `bwSelect()` repeatedly fits time-varying VAR models with different bandwidth parameters, the specification of `bwSelect()` and the estimation function `tvmvar` are very similar. We therefore refer the reader for the code to specify `bwSelect()` to Appendix C.

After that, we provide the number of the notification on a given day and the number of the day itself via the arguments `beepvar` and `dayvar`, respectively. This information is used to exclude cases from the analysis which do not have sufficient previous measurements to fit the specified VAR model. This can be both due to randomly missing data, or because of missingness by design. In the present dataset we have both: within a given day the individual did not always answer at all 10 times. And by design, there is a break between day and night. When not considering the correct successiveness, the estimated parameters do not only reflect effects from t_{t-1} on t but also effects over (possibly) many other time-lags (for instance 10 h over night instead of the intended 1h30).

Via the argument `lags = 1` we specify to fit a first order VAR model and specify with the argument `lambdaSel = "CV"` to select the penalty parameters λ with cross-validation. Finally, with the argument `scale = TRUE` we specify that all variables should be scaled to mean zero and standard deviation 1 before the model is fit. This is recommended when using ℓ_1-regularization, because otherwise the strength of the penalization of a parameter depends on the variance of the predictor variable (see Hastie et al., 2015, p.9). Since the cross-validation scheme uses random draws to define the folds, we set a seed to ensure reproducibility.

```
set.seed(1)
tvvar_obj <- tvmvar(data = mood_data,
                    type = rep("g", 12),
                    level = rep(1, 12),
                    lambdaSel = "CV",
                    timepoints = time_
                    data$time_norm,
                    estpoints =
                    seq(0, 1, length = 20),
                    bandwidth = 0.34,
                    lags = 1,
                    beepvar = time_data$beepno,
                    dayvar = time_data$dayno,
                    scale = TRUE)
```

Before looking at the results we check how many of the 1476 time points were used for estimation, which is shown in the summary that is printed when calling the output object in the console:

```
> tvvar_obj
mgm fit-object

Model class: Time-varying mixed Vector Autore-
gressive (tv-mVAR) model
Lags: 1
Rows included in VAR design matrix: 876/1475
(59.39%)
Nodes: 12
Estimation points: 20
```

This means that the VAR design matrix that is used for estimation has 876 rows. One of the removed time points is the first time point, since it does not have a previous time point. Other time points were excluded because of (a) missing measurements during the day or (b) the day-night break. As an example, from the six rows of the time stamps shown above,

we could use three time points, since a measurement at the previous time point is available.

The absolute values of the estimated VAR coefficients are stored in the object `tvvar_obj$wadj`, which is an array of dimensions $p \times p \times \text{lags} \times$ estpoints, lags is the number of lags, and estpoints is the number of estimation points. For example, the array entry `tvvar_obj$wadj[1, 3, 1, 9]` returns the cross-lagged effect of variable 3 on variable 1 with the first specified lag size (here 1) at estimation point 9. Due to the large number of estimated parameters, we do not show this object here but instead visualize some aspect of it in Figure 10. The signs of all parameters are stored separately in `tvvar_obj$signs`, because signs may not be defined in the presence of categorical variables (which is not the case here). The intercepts are stored in `tvvar_obj$intercepts`.

4.4. Assessing reliability of parameter estimates

To judge the reliability of parameter estimates, we approximate the sampling distribution of all parameters using the nonparametric block bootstrap. The function `resample()` implements this bootstrap scheme and returns the sampling distribution and a selection of its quantiles of each parameter. First we provide the model object `object = tvvar_obj` and the data `data = mood_data`. `resample()` then fits the model specified as in `tvvar_obj` on 50 (nB = 50) different block bootstrap samples, where we specify the number of blocks via blocks. The argument `seeds` provides a random seed for each bootstrap sample and `quantiles` specifies the quantiles shown in the output.

```
res_obj <- resample(object = tvvar_obj,
           data = mood_data,
           nB = 50,
           blocks = 10,
           seeds = 1:50,
           quantiles = c(.05, .95))
```

The $p \times p \times \text{lags} \times \text{estpoints} \times nB$ array `res_obj$bootParameters` contains the sampling distribution of each parameter. For instance, the array entry `res_obj$bootParameters[1, 3, 1, 9,]` contains the sampling distribution of the cross-lagged effect of variable 3 on variable 1 with the first specified lag size (here 1) at time point 9. Due to its size, we do not show this object here but show the 5% and 95% quantiles of the empirical sampling distribution of three time-varying parameters in Figure 10. Also note that the resampling procedure is computationally expensive. With 50 bootstrap samples as specified above, the `resample()` runs approximately 10 minutes.

It is important to keep in mind that the quantiles of these bootstrapped sampling distributions are not confidence intervals around the true parameter. The reason is that the ℓ_1-penalty biases all estimates and hence the whole sampling distribution toward zero which implies that the latter is not centered on the true parameter value.

4.5. Computing time-varying prediction error

Here we show how to compute time-varying nodewise prediction errors. Nodewise prediction errors indicate how well the model fits the data on an absolute scale and is therefore useful to judge the practical relevance of (parts of) a VAR model. See Haslbeck and Waldorp (2018a) for a detailed description of nodewise prediction error (or predictability) in the context of network models and Haslbeck and Fried (2017) for an analysis of predictability in 18 datasets in the field of psychopathology.

The function `predict()` computes predictions and prediction errors from a given *mgm* model object. We first provide the model object `object = tvvar_obj` and the data `data = mood_data`. We then specify the desired types of predictions, here `R2` for the proportion of explained variance and `RMSE` for the Root Mean Squared Error. `tvMethod = "weighted"` specifies how to combine all time-varying models to arrive at a single prediction for each variable across the whole time series (for details see `?predict`). Finally, we provide `consec = time_data$beepno` for the same reasons as above.

```
pred_obj <- predict(object = tvvar_obj,
            data = mood_data,
            errorCon = c("R2", "RMSE"),
            tvMethod = "weighted",
            consec = time_data$beepno)
```

The predictions are stored in `pred_obj$predicted` and the error of the predictions of all time-varying models combined are in `pred_obj$errors`:

```
> pred_obj$errors
      Variable Error.RMSE Error.R2
1      Relaxed       0.939    0.155
2      Down          0.825    0.297
3      Irritated     0.942    0.119
4      Satisfied     0.879    0.201
5      Lonely        0.921    0.182
6      Anxious       0.950    0.086
7      Enthusiastic  0.922    0.169
8      Suspicious    0.818    0.247
9      Cheerful      0.889    0.200
10     Guilty        0.928    0.175
11     Doubt         0.871    0.268
12     Strong        0.896    0.195
```

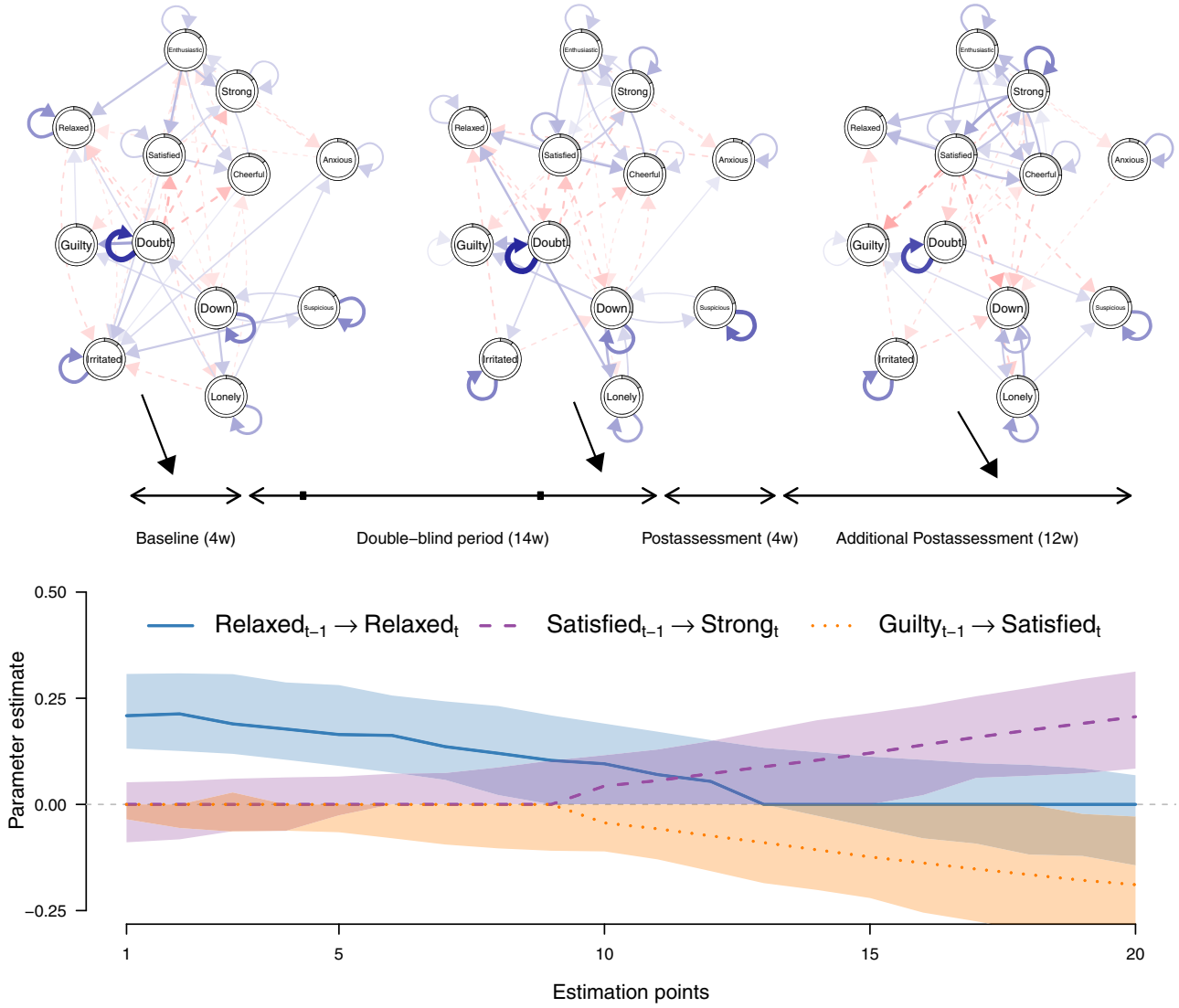

Figure 10. Top row: visualization of VAR(1) models at estimation points 2, 10 and 18. Blue solid arrows indicate positive relationships, red dashed arrows indicate negative relationships, and the width of the arrows is proportional to the absolute value of the corresponding parameter. The self-loops indicate autocorrelations. The colored parts of the ring around each node represents the respective within sample proportion of explained variance (R^2). Bottom row: three parameters plotted as a function of time; the points are the point estimate obtained from the full dataset, the shaded areas indicate the 5% and 95% quantiles of the bootstrapped sampling distribution at each estimation point.

The prediction errors of each time-varying model separately are stored in `pred_obj$tverrors`. Note that here we weight the errors using the same weight vector as used for estimation (see Section 2.3). For details see `?predict.mgm`. In the following section we visualize the time-varying nodewise estimation error for a subset of estimation points.

4.6. Visualizing time-varying VAR model

Figure 10 visualizes a part of the time-varying VAR parameters estimated above. The top row shows visualizations of the VAR parameters for the estimation points 2, 10 and 18. Blue solid arrows indicate positive relationships, red dashed arrows indicate negative relationships. The width of the arrows is proportional to the absolute value of the corresponding parameter. The gray part of the ring around each node indicates the proportion of explained variance of each variables by all other variables in the model. Comparing the VAR estimates across the three shown estimation points reveals that some parameters are strongly time-varying. For example, there is an autocorrelation effect of Relaxed at estimation point 2, which becomes smaller at estimation point 10 and vanishes at estimation point 18. On the other hand, the cross lagged effects $\text{Satisfied}_{t-1} \rightarrow \text{Strong}_t$ and $\text{Guilty}_{t-1} \rightarrow \text{Satisfied}_t$ are equal to zero at estimation point 2 and become larger in estimation point 10 and

18. To better evaluate the time-varying nature of those three parameters we plot them as a line graph in the lower panel of Figure 10. Relating time-varying parameter functions with additional information available about an individual may allow one to explain the changes in parameters. For example, we see that the three time-varying parameters in the lower panel show their largest change after the second reduction of the antidepressant medication. This suggests that the medication reduction could be part of the explanation for this change in parameters. Next to individual interaction parameters, possible analyses can also focus on the changes in intercepts or aggregates of several parameters. For example, one could investigate how the density of the entire or parts of the VAR model changes across time. The code to fully reproduce Figure 10 is not shown here due to its length, but can be obtained from Github (https://github.com/jmbh/tvvar_paper).

4.7. Selecting between stationary and time-varying models

While model selection between stationary and time-varying models is not the topic of this paper and requires a separate treatment to be addressed adequately, we briefly comment on this issue in relation to the methods presented here. One possible way to select between a stationary and a time-varying (VAR) model is to divide the time series into a training and test set. Then one can fit each model on the training set and evaluate on the test set which model has the lower prediction error. In fact, this is the procedure that is implemented in the function `bwSelect()` which we used in Appendix C to select an appropriate bandwidth parameter, and which we described in detail in Section 3.1.2. Thus, if one includes large bandwidths ($b > 1$) that are essentially leading to the same estimates as a stationary model, this bandwidth selection procedure includes a model selection procedure between stationary and time-varying models. However, selecting a (roughly) stationary model with this procedure does not necessarily imply that the data generating process is stationary. The reason is that the procedure strikes a balance between stability of estimates and sensitivity to estimate time-varying parameters. If the sample size is low, the procedure will therefore select a stationary model even if the data generating process is time-varying.

Another possibility is to rely on information criteria such as the AIC (see e.g., Bringmann et al., 2018). Finally, one could construct a hypothesis test with the null hypothesis that the data generating process is stationary VAR model. This could be done by estimating a stationary VAR model on the data set at hand, and then generating B time series of the same length as the original time series from this model. Then one fits a time-varying VAR model to each of those data sets and records a mean (over variables) prediction error. This way we obtain the sampling distribution of the prediction error under the null hypothesis, and we can perform a hypothesis test using the prediction error of the time-varying VAR model on the actual data as the test-statistic. We could for instance set $\alpha = 0.05$, which would mean that we would accept the time-varying model if its error is smaller than the 5% quantile of the sampling distribution. For the data in this tutorial this leads to the rejection of the null-hypothesis, which means that the data generating mechanism is not a stationary VAR model and it is therefore more appropriate to fit a time-varying VAR model. We provide the code to reproduce this test on in the supplementary materials and Github https://github.com/jmbh/tvvar_paper.

5. Discussion

We compared the performance of GAM and kernel-smoothing (KS) based methods in combination with and without regularization in estimating time-varying VAR models in situations that are typical for psychological applications. Our simulation results allow researchers to select the best method amongst the ones we considered here based on sample size and their assumptions about the sparsity of the true VAR model. In addition, we provided step-by-step tutorials for the KS based method using the R-package *mgm* (Section 4) and for the GAM based method using the R-package *tvvarGAM* (Appendix D).

Next to assessing the relative performance of different methods, our paper also provides the first overview of how many observations are roughly necessary to estimate time-varying VAR models. For the time-varying functions studied in our paper, already for $n > 46$ the best time-varying method outperformed stationary methods, suggesting that time-varying methods can be applied to typical ESM data. However, it is important to keep in mind that if the sample size is low, the time-varying methods return very similar estimates as their stationary counterparts. Thus, if the true parameter function is heavily depending on time, and the sample size is small, time-varying methods will not be able to recover most of this dependency on time.

There are several interesting avenues for future research on time-varying VAR models. First, in the present paper we focused on frequentist methods.

However, time-varying VAR models can also be estimated in a Bayesian framework (Krueger, 2015). It would be interesting to compare the performance of these methods to the methods presented in this paper. Second, the methods presented here could be extended to beyond the standard VAR models. Examples are mixed VAR models, which allow to jointly model variables defined on different domains (Haslbeck & Waldorp, 2018b), unified Structural Equation Models (SEM) that allow an extension of SEM models to different domains (Kim et al., 2007), or the graphical VAR model (Abegaz & Wit, 2013), which estimates both the VAR parameters and the residual structure Σ (see Section 2.1). In this model, identifying time-varying parameters is especially important, because spurious relations in the residual structure can be induced by time-varying parameters. Third, all methods discussed in this paper are based on the assumption that the true parameters are smooth functions of time. However, in some situations it might be more appropriate to assume different kinds of local stationarity, for example piece-wise constant functions (e.g., Bringmann & Albers, 2019; Gibberd & Nelson, 2017). It would be useful to make those alternative estimation methods available to applied researchers, and possibly combine them with the methods presented here. Fourth, the Gaussian kernel in the KS method could be replaced by kernels with finite domains such as the box car function, in order to improve the computational efficiency of the algorithm. Finally, in this paper we focused on the population performance of the two presented methods in a variety of settings. However, we did not discuss in detail how to select between models (for example stationary vs. time-varying) in a practical application. We believe that a conclusive discussion of different model selection strategies in a variety of realistic situations would be an important avenue for future work.

Article Information

Conflict of interest disclosures: Each author signed a form for disclosure of potential conflicts of interest. No authors reported any financial or other conflicts of interest in relation to the work described.

Ethical principles: The authors affirm having followed professional ethical guidelines in preparing this work. These guidelines include obtaining informed consent from human participants, maintaining ethical treatment and respect for the rights of human or animal participants, and ensuring the privacy of participants and their data, such as ensuring that individual participants cannot be identified in reported results or from publicly available original or archival data.

Funding: This work was supported by the European Research Council Consolidator Grant no. 647209 and NWO Veni Grant no. 451-17-017.

Role of the funders/sponsors: None of the funders or sponsors of this research had any role in the design and conduct of the study; collection, management, analysis, and interpretation of data; preparation, review, or approval of the manuscript; or decision to submit the manuscript for publication.

References

Aan Het Rot, M., Hogenelst, K., & Schoevers, R. A. (2012). Mood disorders in everyday life: A systematic review of experience sampling and ecological momentary assessment studies. *Clinical Psychology Review*, *32*(6), 510–523. doi:10.1016/j.cpr.2012.05.007

Abegaz, F., & Wit, E. (2013). Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics*, *14*(3), 586–599. doi:10.1093/biostatistics/kxt005

Andersen, R. (2009). Nonparametric methods for modeling nonlinearity in regression analysis. *Annual Review of Sociology*, *35*(1), 67–85. doi:10.1146/annurev.soc.34.040507.134631

Bak, M., Drukker, M., Hasmi, L., Os, J. & van, (2016). An n= 1 clinical network analysis of symptoms and treatment in psychosis. *PloS One*, *11*(9), e0162811. doi:10.1371/journal.pone.0162811

Belsley, D. A., & Kuti, E. (1973). Time-varying parameter structures: An overview. In *Annals of economic and social measurement* (vol. 2, no. 4, pp. 375–379). NBER.

Borsboom, D., & Cramer, A. O. (2013). Network analysis: An integrative approach to the structure of psychopathology. *Annual Review of Clinical Psychology*, *9*(1), 91–121. doi:10.1146/annurev-clinpsy-050212-185608

Bringmann, L. F., & Albers, C. J. (2019). Inspecting gradual and abrupt changes in emotion dynamics with the time-varying change point autoregressive model.

Bringmann, L. F., Ferrer, E., Hamaker, E. L., Borsboom, D., & Tuerlinckx, F. (2018). Modeling nonstationary emotion dynamics in dyads using a time-varying vector-autoregressive model. *Multivariate Behavioral Research*, *53*(3), 293–314. doi:10.1080/00273171.2018.1439722

Bringmann, L. F., Hamaker, E. L., Vigo, D. E., Aubert, A., Borsboom, D., & Tuerlinckx, F. (2017). Changing

dynamics: Time-varying autoregressive models using generalized additive modeling. *Psychological Methods*, *22*(3), 409–425. doi:10.1037/met0000085

Bringmann, L. F., Haslbeck, J. M. B. & Tendeiro J. N. (2020). tvvarGAM. https://github.com/LauraBringmann/tvvarGAM. GitHub.

Bringmann, L. F., Vissers, N., Wichers, M., Geschwind, N., Kuppens, P., Peeters, F., Borsboom, D., & Tuerlinckx, F. (2013). A network approach to psychopathology: New insights into clinical longitudinal data. *PloS One*, *8*(4), e60188. doi:10.1371/journal.pone.0060188

Casas, I., Fernandez-Casal, R. (2018). tvreg: Time-varying coefficients linear regression for single and multiple equations [Computer software manual]. (R package version 0.3.0). https://CRAN.R-project.org/package=tvReg

Dakos, V., Lahti, L. (2013). R early warning signals toolbox. *The R Project for Statistical Computing*. http://cran.r-project.org/web/packages/earlywarnings/index.html

Epskamp, S., Waldorp, L. J., Mõttus, R., & Borsboom, D. (2018). The Gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, *53*(4), 453–480. doi:10.1080/00273171.2018.1454823

Fabio Di Narzo, A., Aznarte, J. L., Stigler, M. (2009). tsDyn: Time series analysis based on dynamical systems theory [Computer software manual]. (R package version 0.7). https://cran.r-project.org/package=tsDyn/vignettes/tsDyn.pdf

Fan, J., & Gijbels, I. (1996). Applications of local polynomial modelling. In *Local polynomial modelling and its applications* (pp. 159–216). Springer.

Fisher, A. J., Medaglia, J. D., & Jeronimus, B. F. (2018). Lack of group-to-individual generalizability is a threat to human subjects research. *Proceedings of the National Academy of Sciences*, *114*(27), E6106–E6115. doi:10.1073/pnas.1711978115

Fisher, A. J., Reeves, J. W., Lawyer, G., Medaglia, J. D., & Rubel, J. A. (2017). Exploring the idiographic dynamics of mood and anxiety via network analysis. *Journal of Abnormal Psychology*, *126*(8), 1044–1056. doi:10.1037/abn0000311

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1. doi:10.18637/jss.v033.i01

Gibberd, A. J., & Nelson, J. D. (2017). Regularized estimation of piecewise constant gaussian graphical models: The group-fused graphical lasso. *Journal of Computational and Graphical Statistics (Just-Accepted)*, *26*(3), 623–634. doi:10.1080/10618600.2017.1302340

Gibbert, A. (2017). *GraphTime.* https://github.com/GlooperLabs/GraphTime. GitHub.

Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, *21*(2), 215–223. doi:10.1080/00401706.1979.10489751

Groen, R. N., Snippe, E., Bringmann, L. F., Simons, C. J., Hartmann, J. A., Bos, E. H., & Wichers, M. (2019). Capturing the risk of persisting depressive symptoms: A dynamic network investigation of patients' daily symptom experiences. *Psychiatry Research*, *271*, 640–648. doi:10.1016/j.psychres.2018.12.054

Hamaker, E. L., Grasman, R. P., & Kamphuis, J. H. (2010). Regime-switching models to study psychological process. In P. C. M. Molenaar & K. M. Newell (Eds.), *Individual pathways of change: Statistical models for analyzing learning and development* (p. 155–168). American Psychological Association.

Hamilton, J. D. (1994). *Time series analysis* (Vol. 2). Princeton university press Princeton.

Hartmann, J. A., Wichers, M., Menne-Lothmann, C., Kramer, I., Viechtbauer, W., Peeters, F., Schruers, K. R. J., van Bemmel, A. L., Myin-Germeys, I., Delespaul, P., van Os, J., & Simons, C. J. P. (2015). Experience sampling-based personalized feedback and positive affect: A randomized controlled trial in depressed patients. *PLoS One*, *10*(6), e0128095. doi:10.1371/journal.pone.0128095

Haslbeck, J. M. B., & Fried, E. I. (2017). How predictable are symptoms in psychopathological networks? A reanalysis of 18 published datasets. *Psychological Medicine*, *47*(16), 1–10. doi:10.1017/S0033291717001258

Haslbeck, J. M. B., & Waldorp, L. J. (2018a). How well do network models predict observations? on the importance of predictability in network models. *Behavior Research Methods*, *50*(2), 853–861. doi:10.3758/s13428-017-0910-x

Haslbeck, J. M. B., & Waldorp, L. J. (2018b). MGM: Structure estimation for time-varying mixed graphical models in high-dimensional data. *arXiv preprint arXiv: 1510.06871*.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity*. CRC press.

Holmes, E., Ward, E., Wills, K. (2013). Marss: Multivariate autoregressive state-space modeling [Computer software manual]. (R package version 3.9). http://cran.r-project.org/web/packages/MARSS/

Holmes, E. E., Ward, E. J., & Wills, K. (2012). Marss: Multivariate autoregressive state-space models for analyzing time-series data. *The R Journal*, *4*(1), 11. doi:10.32614/RJ-2012-002

Keele, L. J. (2008). *Semiparametric regression for the social sciences*. John Wiley & Sons.

Kim, J., Zhu, W., Chang, L., Bentler, P. M., & Ernst, T. (2007). Unified structural equation modeling approach for the analysis of multisubject, multivariate functional MRI data. *Human Brain Mapping*, *28*(2), 85–93. doi:10.1002/hbm.20259

Kossakowski, J., Groot, P., Haslbeck, J. M. B., Borsboom, D., & Wichers, M. (2017). Data from 'critical slowing down as a personalized early warning signal for depression. *Journal of Open Psychology Data*, *5*(1), 1. doi:10.5334/jopd.29

Kramer, I., Simons, C. J. P., Hartmann, J. A., Menne-Lothmann, C., Viechtbauer, W., Peeters, F., Schruers, K., van Bemmel, A. L., Myin-Germeys, I., Delespaul, P., van Os, J., & Wichers, M. (2014). A therapeutic application of the experience sampling method in the treatment of depression: A randomized controlled trial. *World Psychiatry*, *13*(1), 68–77. doi:10.1002/wps.20090

Kroeze, R., Van Veen, D., Servaas, M. N., Bastiaansen, J. A., O., Voshaar, R., Borsboom, D., & Riese, H. (2016). Personalized feedback on symptom dynamics of psychopathology: A proof-of-principle study. *Journal for Person-Oriented Research*, *3*(1), 1–10. doi:10.17505/jpor.2017.01

Krueger, F. (2015). bvarsv: Bayesian analysis of a vector autoregressive model with stochastic volatility and time-

varying parameters [Computer software manual]. (R package version 1.1). https://CRAN.R-project.org/package=bvarsv

Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, *10*(Oct), 2295–2328.

Molenaar, P. C. (2004). A manifesto on psychology as idiographic science: Bringing the person back into scientific psychology, this time forever. *Measurement: Interdisciplinary Research & Perspective*, *2*(4), 201–218. doi:10.1207/s15366359mea0204_1

Monti, R. (2014). *pysingle*. https://github.com/piomonti/pySINGLE. GitHub.

Monti, R. P., Hellyer, P., Sharp, D., Leech, R., Anagnostopoulos, C., & Montana, G. (2014). Estimating time-varying brain connectivity networks from functional MRI time series. *NeuroImage*, *103*, 427–443. http://www.sciencedirect.com/science/article/pii/S1053811914006168 doi:10.1016/j.neuroimage.2014.07.033

Olthof, M., Hasselman, F., Strunk, G., Rooij, M., van, Aas, B., Helmich, M. A., … Lichtwarck-Aschoff, A. (2019). Critical fluctuations as an early-warning signal for sudden gains and losses in patients receiving psychotherapy for mood disorders. *Clinical Psychological Science*, *8*(1), 25–35. doi:10.1177/2167702619865969

Ou, L., Hunter, M. D., Chow, S.-M. (2019). dynr: Dynamic modeling in r [Computer software manual]. (R package version 0.1.14-9). https://CRAN.R-project.org/package=dynr

Pe, M. L., Kircanski, K., Thompson, R. J., Bringmann, L. F., Tuerlinckx, F., Mestdagh, M., Mata, J., Jaeggi, S. M., Buschkuehl, M., Jonides, J., Kuppens, P., & Gotlib, I. H. (2015). Emotion-network density in major depressive disorder. *Clinical Psychological Science*, *3*(2), 292–300. doi:10.1177/2167702614540645

Robinaugh, D. J., Hoekstra, R. H. A., & Borsboom, D. (2019). The network approach to psychopathology: A review of the literature 2008-2018.

Robinson, P. M. (1989). Nonparametric estimation of time-varying parameters. In *Statistical analysis and forecasting of economic structural change* (pp. 253–264). Springer.

Scheffer, M., Bascompte, J., Brock, W. A., Brovkin, V., Carpenter, S. R., Dakos, V., Held, H., van Nes, E. H., Rietkerk, M., & Sugihara, G. (2009). Early-warning signals for critical transitions. *Nature*, *461*(7260), 53–59. doi:10.1038/nature08227

Schmittmann, V. D., Cramer, A. O., Waldorp, L. J., Epskamp, S., Kievit, R. A., & Borsboom, D. (2013). Deconstructing the construct: A network perspective on psychological phenomena. *New Ideas in Psychology*, *31*(1), 43–53. doi:10.1016/j.newideapsych.2011.02.007

Shadish, W. R., Zuur, A. F., & Sullivan, K. J. (2014). Using generalized additive (mixed) models to analyze single case designs. *Journal of School Psychology*, *52*(2), 149–178. doi:10.1016/j.jsp.2013.11.004

Snippe, E., Viechtbauer, W., Geschwind, N., Klippel, A., De Jonge, P., & Wichers, M. (2017). The impact of treatments for depression on the dynamic network structure of mental states: Two randomized controlled trials. *Scientific Reports*, *7*(1), 46523. doi:10.1038/srep46523

Tarvainen, M. P., Hiltunen, J. K., Ranta-Aho, P. O., & Karjalainen, P. A. (2004). Estimation of nonstationary EEG with Kalman smoother approach: an application to event-related synchronization (ERS). *IEEE Transactions on Biomedical Engineering*, *51*(3), 516–524. doi:10.1109/TBME.2003.821029

Tong, H., & Lim, K. (1980). Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society: Series B (Methodological))*, *42*(3), 245–268. doi:10.1142/9789812836281_0002

van Borkulo, C., Boschloo, L., Borsboom, D., Penninx, B. W., Waldorp, L. J., & Schoevers, R. A. (2015). Association of symptom network structure with the course of depression. *JAMA Psychiatry*, *72*(12), 1219–1226. doi:10.1001/jamapsychiatry.2015.2079

van de Leemput, I. A., Wichers, M., Cramer, A. O. J., Borsboom, D., Tuerlinckx, F., Kuppens, P., van Nes, E. H., Viechtbauer, W., Giltay, E. J., Aggen, S. H., Derom, C., Jacobs, N., Kendler, K. S., van der Maas, H. L. J., Neale, M. C., Peeters, F., Thiery, E., Zachar, P., & Scheffer, M. (2014). Critical slowing down as early warning for the onset and termination of depression. *Proceedings of the National Academy of Sciences*, *111*(1), 87–92. doi:10.1073/pnas.1312114110

van der Krieke, L., Blaauw, F. J., Emerencia, A. C., Schenk, H. M., Slaets, J. P., Bos, E. H., … Jeronimus, B. F. (2017). Temporal dynamics of health and well-being: A crowdsourcing approach to momentary assessments and automated generation of personalized feedback. *Psychosomatic Medicine*, *79*(2), 213–223. doi:10.1097/PSY.0000000000000378

Wichers, M., Groot, P. C., Psychosystems, E., & Group, E. (2016). Critical slowing down as a personalized early warning signal for depression. *Psychotherapy and Psychosomatics*, *85*(2), 114–116. doi:10.1159/000441458

Wigman, J. T. W., van Os, J., Borsboom, D., Wardenaar, K. J., Epskamp, S., Klippel, A., Viechtbauer, W., Myin-Germeys, I., & Wichers, M. (2015). Exploring the underlying structure of mental disorders: Cross-diagnostic differences and similarities from a network perspective using both a top-down and a bottom-up approach. *Psychological Medicine*, *45*(11), 2375–2387. doi:10.1017/S0033291715000331

Wood, S. N. (2006). *Generalized additive models: An introduction with R*. Chapman and Hall/CRC.

Wood, S. N., & Augustin, N. H. (2002). Gams with integrated model selection using penalized regression splines and applications to environmental modelling. *Ecological Modelling*, *157*(2-3), 157–177. doi:10.1016/S0304-3800(02)00193-X

Appendix

A. Sampling variation around aggregated absolute errors

In Figure 5 we reported the mean absolute error, averaged over time points and iterations. These population level mean errors indicate which method has the lowest *expected* error in a given scenario. However, it is also interesting to evaluate how large the population sampling variance is around the mean errors. We therefore display a version of Figure 5 that includes the 25% and 75% quantiles of the population sampling distribution:

How can we interpret these quantiles? Let's take the performance of GAM and KS for $n = 103$ in panel (b) as an

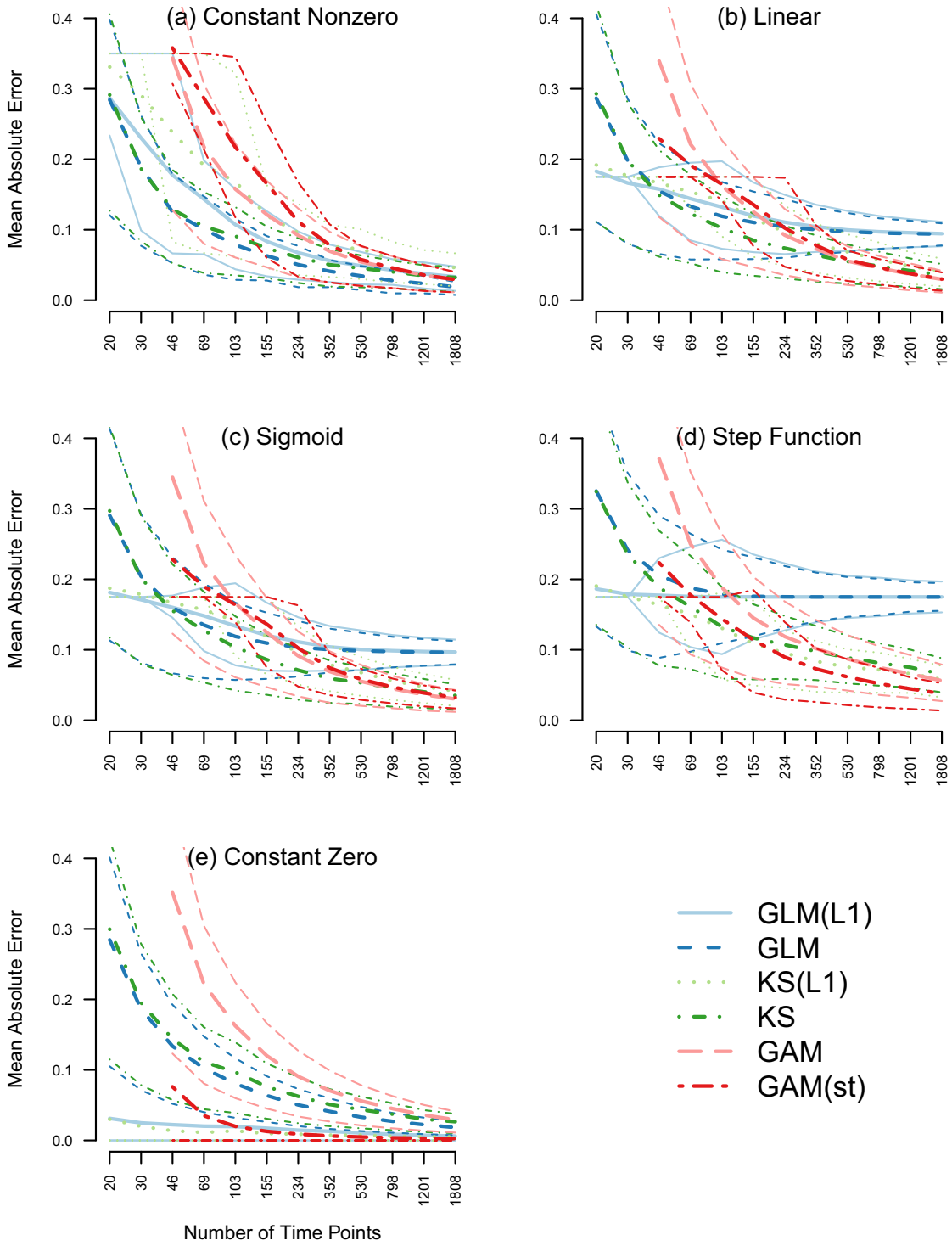

Figure 11. The five panels show the mean absolute estimation error (solid lines) averaged over the same type, time points, and iterations as a function of the number of observations n on a log scale. We report the error of six estimation methods: stationary unregularized regression (blue), stationary ℓ_1-regularized regression (red), time-varying regression via kernel-smoothing (yellow), time-varying ℓ_1-regularized regression via kernel-smoothing (green), time-varying regression via GAM (pink), and time-varying regression via GAM with thresholding at 95% CI (orange). Some data points are missing because the respective models are not identified in that situation (see Section 3.1.2). The dashed lines indicate the 25% and 75% quantiles, averaged over time points.

example. The population mean error is larger for GAM than for KS in this scenario. Note that this difference in mean errors is on the population level and therefore no test is necessary to judge its significance. However, we see that the sampling distributions of the two errors are largely overlapping. This implies that also the difference of the two

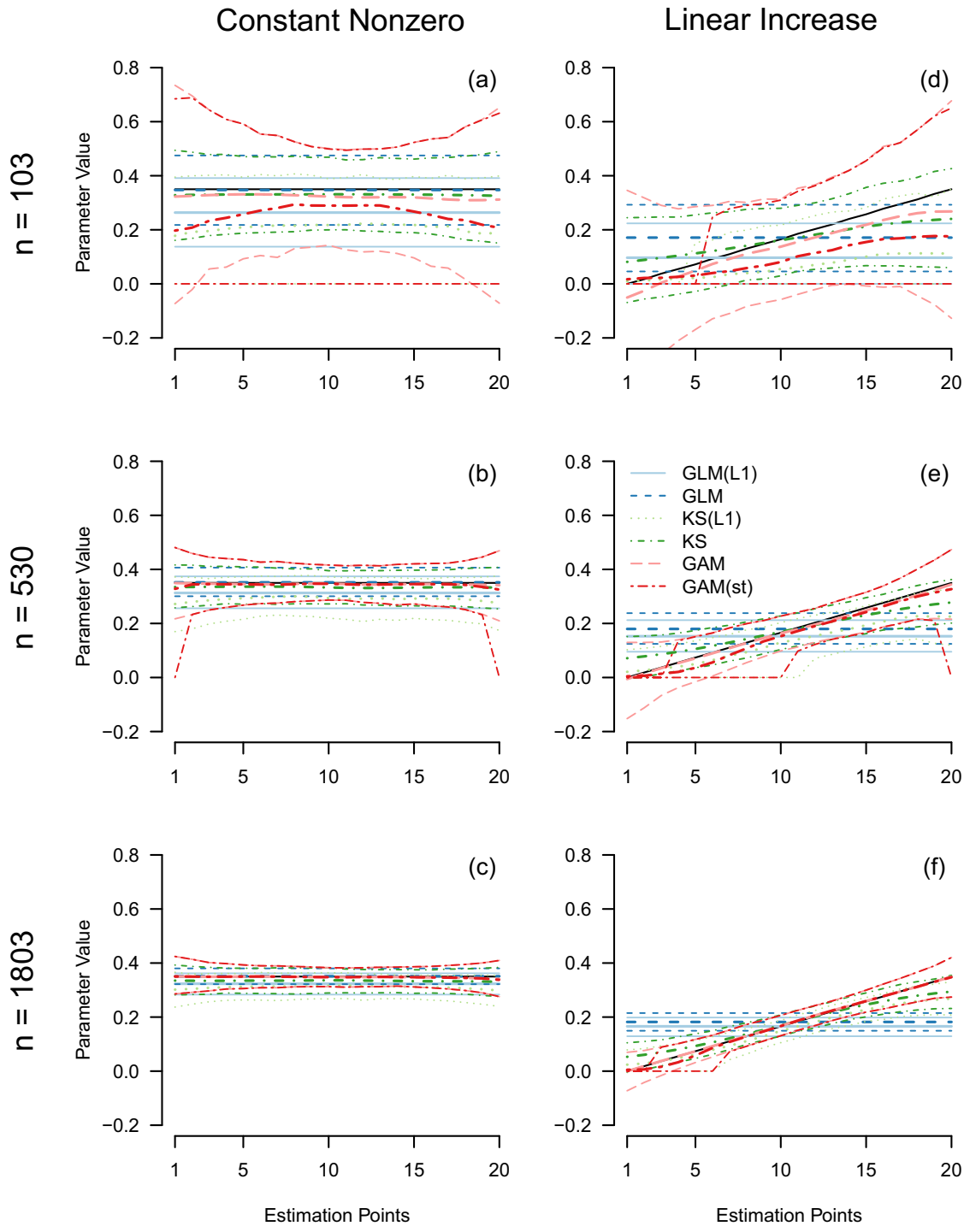

Figure 12. Mean (tick line) and standard deviations (thin line) of estimates for the constant parameter (left column), and the linear increasing parameter (right column), for $n = 103$ (top row), $n = 530$ (second row) and $n = 1803$ (bottom row) averaged over iterations, separately for the five estimation methods: stationary ℓ_1-regularized regression (red), unregularized regression (blue), time-varying ℓ_1-regularized regression via kernel-smoothing (green), time-varying regression via GAM (pink), and time-varying regression via GAM with thresholding at 95% CI (orange).

errors has a large variance, which means that if $n = 103$, it is difficult to predict for a specific sample whether GAM or KS has a larger error.

We see that for unregularized methods the confidence interval is large for small n and becomes smaller when increasing n. For the ℓ_1-regularized methods, the quantiles are first small, then increase, and then decrease again as a function of n. The reason is that for small n, these methods set all most estimates to zero, and therefore the upper and lower quantiles have the same value. An extreme

case is the true zero constant function in Figure 11 panel (e). Here both quantiles are zero for all n, while the mean absolute error is larger than 0 and approaches 0 with increasing n.

B. Sampling variation around absolute errors over time

Figure 12 displays the mean estimates also shown in Figure 6 in Section 3.1.3, but in addition displays the 10% and 90% quantiles of the estimates. The sampling variance is small for $n = 103$, but approaches zero as n becomes large.

C. Code to select appropriate bandwidth in KS(L1) method

The function `bwSelect()` fits time-varying VAR models with different bandwidth parameters to a set of training sets and computes the out-of-sample prediction error in the hold-out sets. We then select the bandwidth that minimizes this prediction error across variables and hold-out sets. For details about how these training/test sets are chosen exactly see `?bwSelect` or Haslbeck and Waldorp (2018b).

Since we fit the time-varying VAR model of our choice repeatedly, we provide all parameters we specified to the estimation function `tvmvar()` as described in Section 4.3. In addition, we specify via `bwFolds`, the number of training set/test set splits, via `bwFoldsize` the size of the test sets, and via `bwSeq` the sequence of candidate bandwidth-values. Here, we chose ten equally spaced values in $[0.01, 1]$.

```
bwSeq <- seq(0.01, 1, length = 10)

set.seed(1)
bw_object <- bwSelect(data=mood_data,
                      type=rep("g", 12),
                      level=rep(1, 12),
                      bwSeq=bwSeq,
                      bwFolds = 1,
                      bwFoldsize = 20,
                      modeltype = "mvar",
                      lags = 1,
                      scale=TRUE,
                      timepoints=time_data$time_norm,
                      beepvar=time_data$beepno,
                      dayvar=time_data$dayno,
                      pbar=TRUE)
bandwidth <- bwSeq[which.min(bw_object$meanError)]

[1] 0.34
```

The output object `bw_object` contains all fitted models and unaggregated prediction errors. We see that the bandwidth 0.34 minimized the average out-of-sample prediction error. The full bandwidth path is shown in Figure 13.

The bandwidth value of 0.01 is clearly too small, indicated by a large prediction error. The error then tends to become smaller as a function of b until its minimum at 0.34 and then increases again. Note that if the smallest/largest considered bandwidth value minimizes the error, another search should be conducted with smaller/larger bandwidth values.

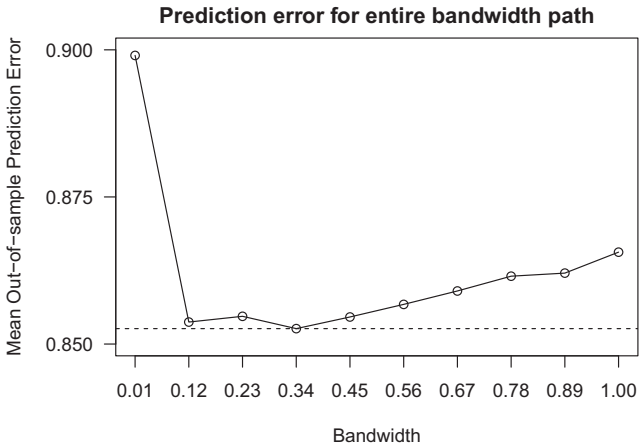

Figure 13. Average out-of-sample prediction error for different bandwidth values obtained from the function. The bandwidth value 0.34 returns the smallest error, indicated by the dashed line.

D. Estimating time-varying VAR model via GAM(st)

Here we show how to estimate a time-varying VAR model via the GAM(st) method. All analyses are performed using the R-package *tvvarGAM* (Bringmann, Haslbeck, & Tendeiro, 2020) and the shown code is fully reproducible, which means that the reader can execute the code while reading. The code below can also be found in an R-file on Github: https://github.com/jmbh/tvvar_paper.

D.1. Load R-packages and dataset

Similar to Section 4.2 we load the dataset from the *mgm* package, and subset the 12 mood related variables. In addition, we load the *tvvarGAM* package (version 0.1.1).

```
library(mgm) # Version 1.2-8

mood_data <- as.matrix(symptom_data$data[, 1:12]) # Subset variables
mood_labels <- symptom_data$colnames[1:12] # Subset variable labels
colnames(mood_data) <- mood_labels
time_data <- symptom_data$data_time

# Install from Github:
library(devtools)
install_github("LauraBringmann/tvvarGAM")
library(tvvarGAM)
```

D.2. Estimating time-varying VAR model

We use the function `tvvarGAM()` to estimate the time-varying VAR model. We provide the data via the data argument and provide an integer vector of length n indicating the successiveness of measurements by specifying the number of the recorded notification and the day

number via the arguments `beepvar` and `dayvar`. The latter is used similarly as in the *mgm* package to compute the VAR design matrix. Via the argument `nb` we specify the number of desired basis functions (see Section 2.2). First, we estimated the model with 10 basis functions. However, because some of the edf of the smooth terms were close to 10, we doubled the number of basis functions (see discussion in Section 2.2).

```
tvvargam_obj <- tvvarGAM(data = mood_data,
                beepvar = time_data$beepno,
                dayvar = time_data$dayno,
                nb = 20,
                scale = TRUE)
```

The output object consists of a list with three entries: `tvvargam_objResultsGAMEstimate` is a $(p+1) \times p \times$ *timepoints* array that contains the parameter estimate at each time point. The first row contains the estimated intercepts. The two other list entries have the same dimensions and contain the 5% and 95% confidence intervals for the estimates in `tvvargam_objResultsGAMEstimate`. Thus, in case of the *tvvarGAM* package no separate resampling scheme is necessary in order to get a measure for the reliability of parameters.

D.3. Visualize time-varying VAR model

Figure 14 visualizes the part of the time-varying VAR like Figure 10 above, however, now with the estimates from the *tvvarGAM* package. Notice that for visualization purposes we used the tresholded version of the time-varying VAR, thus showing only the arrows that are significant (p-value < 0.05).

Similarly to the analysis performed with the KS(L1) method we visualize the VAR parameters at estimation points 2, 10 and 18 (top row Figure 14. We see that less

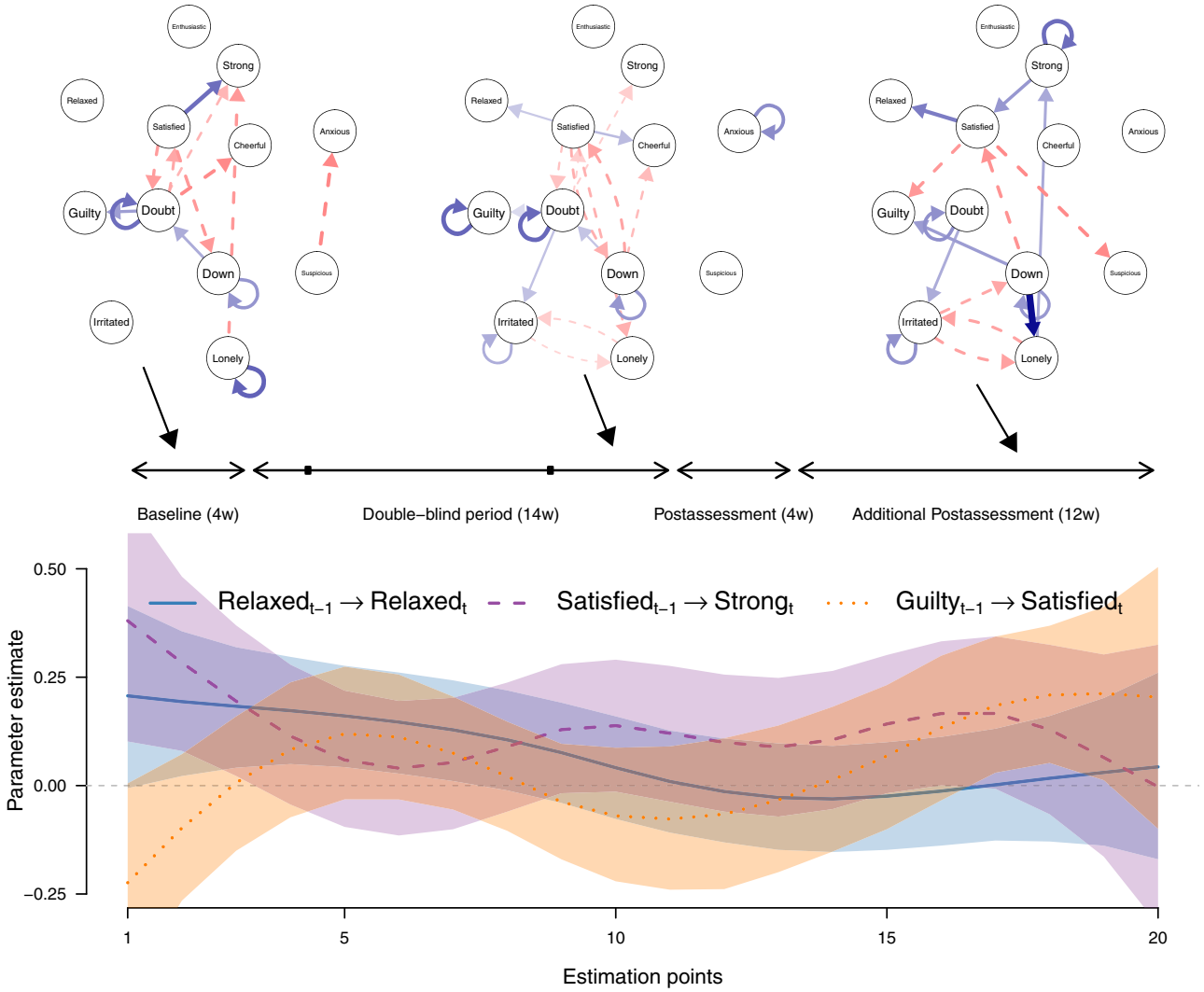

Figure 14. Top row: visualization of thresholded VAR models at estimation points 2, 10 and 18, estimated with the spline-based method. Blue arrows indicate positive relationships, red arrows indicate negative relationships, and the width of the arrows is proportional to the absolute value of the corresponding parameter. The self-loops indicate autocorrelations. Bottom row: three parameters plotted as a function of time; the points are unthresholded point estimates, the shading indicates the 5% and 95% credible intervals at each estimation point.

edges are present than in the results of the KS(L1) method, which indicates that the GAM(ks) method is more conservative. The bottom row of Figure 14 shows a line plot of the same three parameters as in the analysis with the KS(L1) method. We see that the effect of Relaxed on itself tends to decrease over the measured time interval, which is consistent with the results of the KS(L1) method. However, results of the cross-lagged effects of Satisfied on Strong, and of Guilty on Satisfied are only consistent with the results of the KS(1) method in the middle of the time series. The largest difference between the two methods is the increase of the effect of Guilty on Satisfied is noteworthy, while the KS(L1) method estimates a decrease. It seems that the GAM(st) estimates in the second half of the time series are incorrect, because because if one splits the time series in half and estimates two unregularized stationary VAR models, then the effect of Guilty on Satisfied is clearly negative in the second half of the time series. In general, the large changes and the much larger credible intervals at the beginning and the end of the time series indicate that the estimates are very unstable in those regions. This is consistent with the high standard deviation of estimates of the GAM and GAM(st) method shown in Figure 12. The code to fully reproduce Figure 14 is not shown here due to its length, but can be obtained from Github https://github.com/jmbh/tvvar_paper.