Recovering Time-Varying Network Structures from Panel Data using Break Detection

Author:
Alexander R. J. Gruisen (416038)

Supervisor:
dr. Yutao Sun

Second assessor:
dr. Andreas Pick

April 29, 2020

Master Thesis: programme Econometrics

Econometrics and Management Science

Erasmus School of Economics

Erasmus University Rotterdam

Abstract

I extend existing spatial panel data models to incorporate time variation in unknown network structures by means of structural breaks. For optimization I propose a two-step Adaptive Lasso algorithm to sequentially detect multiple unknown breaks in the spatial dependence and fully recover the underlying network. The use of MLE plus regularization enables to estimate the high-dimensional nonlinear model. Via an extensive simulation study I showcase the accuracy and robustness of the algorithm. Moreover, the accuracy maintains when either break detection or network recovery fails. Finally, I employ the algorithm to the regional housing prices in The Netherlands. The results illustrate the time-varying inter-connectivity, for which the possibility of a contiguity-based underlying network is rejected.

Keywords: Spatial weighting matrix, structural breaks, time-varying spatial dependence, maximum likelihood estimation, housing market.

Contents

1 Introduction

Spatial panel data models use a spatial weighting matrix to represent a network structure. These structures consist of connections between individuals or locations. Networks help us understand how data is organized, individuals are related to each other and outcomes are produced. Allowing for time variation can provide insight on how the network changes over time, as individuals and especially their interactions commonly do so.

The potential benefits of time-varying networks urges recent literature to exploit the correct approaches to model such networks. Some papers make assumptions about the network structure, but this is not always feasible in practice. The problem of unknown network structure demands for techniques to recover the network from the data itself. Here we should ask ourselves whether networks change, and if so, when the change happens. However existing literature fails to jointly address and tackle the two problems of 1) unknown network structure in data and 2) absence of time variation in the spatial dependence. To try to fill this void, my research will propose an adequate approach to recover unknown time-varying networks from spatial panel data.

Literature produces many different approaches on how to model the weighting matrix. Conventional specifications are based on contiguity criteria (e.g. k-nearest neighbors (Blume et al., 2011) or geographic distances (Blume et al., 2015)). However such specifications are both time-invariant and postulated. Qu & Lee (2015) and de Paula et al. (2018) stress the presence of a severe bias in the estimation results when we choose a postulated specification when in fact the true structure is unknown. Either way, both postulated and real-life networks are often sparse (Chandrasekhar, 2016). Gratefully, de Paula et al. (2018) succeed in estimating the full weighting matrix while exploiting sparsity in the model using regularization. Their model incorporates an asymmetric network structure, specifically useful in settings where we do not consider a topological contiguity. Despite showcasing significant time differences in the network structure, de Paula et al. (2018) fail to incorporate time variation in the spatial dependence. Omission of the time-varying component causes estimation biases and inconsistencies in the spatial dependence. Therefore a model specification is needed that also incorporates the time-varying component of the network.

Interestingly, Billé et al. (2019) extend previous spatial network literature of Lee & Yu (2012) by relaxing the spatial coefficient over time, making it suitable for macro-panels. The spatial dependence can fluctuate per time instance. Their work is novel, but the pitfalls are evident: they make prior assumptions concerning the network structure. Moreover, Billé et al. (2019)'s simulation study demonstrates vast periods of constant spatial dependence which were not known *a priori*. These drawbacks give rise to first check if and when networks change (and thus also when networks stay constant), whereafter the time-varying network should be estimated.

Checking the time variation can be formulated as a structural break detection in the parameters. Currently only Otto & Steinert (2018) seem to explore the use of break detection in the model while recovering the unknown network. They adapted the break detection method of Chan et al. (2014) to a spatial setting, which extends conventional recursive break detection methods (Bai & Perron, 2003; Baltagi et al., 2016). Essentially, they bring the time variation of Billé et al. (2019) together with unknown network structure recovery of de Paula et al. (2018). Though seemingly promising, Otto & Steinert (2018) fail to recognize several issues. That is, they impose the breaks in the fixed-effect parameters. Consequently, time variation in the spatial dependence does not originate from changes in the network structure. Moreover, they oversimplify break detection by imposing large break values. These pitfalls urge to adapt the break detection method of Otto & Steinert (2018) to a method that checks for breaks explicitly formulated in the spatial dependence.

As a consequence of incorporating breaks *and* recovering the unknown network, the spatial dependence becomes nonlinear. The nonlinearity evokes complications with identifying the parameters if one is to use the estimation methods of de Paula et al. (2018) and Otto & Steinert (2018). Rather, Maximum Likelihood Estimation (MLE) is more suitable. By making an assumption on the weighting matrix and on the error term MLE is capable of overcoming the mentioned identification problems without needing a completely identifiable reduced form, shown by Lee (2004) for spatial models. Thus, MLE with penalty terms for network recovery with preliminary break detection overcomes all encountered pitfalls and brings the solution to the two aforementioned problems.

I extend existing spatial network literature by proposing a two-step Adaptive Lasso algorithm. I use the static nonlinear spatial panel data model specification of Billé et al. (2019) adjusted to incorporate breaks and an unknown network structure. In the first step I propose a modified break detection method of Otto & Steinert (2018) to locate multiple unknown breaks at unknown points in time. In the second step, MLE recovers the unknown time-varying network and estimates the full model, using a Lasso regularization similar to de Paula et al. (2018).

A Monte Carlo simulation study investigates the effectiveness of the two-step

Adaptive Lasso algorithm. In particular, I construct three real-world network structures which vary in pattern, number of links and link value. Furthermore, I incorporate different numbers of breaks in the spatial coefficient. Subsequently, the data is put in the two-step Adaptive Lasso algorithm. For appropriate comparison I consider all combinations of (hyper)parameter initializations. In the results I display figures of the weighting matrices and network topologies, and calculate summary, element-level and descriptive statistics. The simulation results indicate adequate performance across settings. Especially network recovery is accurate. Robustness checks demonstrate in which settings the algorithm falls short regarding break detection. However more importantly, the results suggest the wide and satisfactorily applicability of the algorithm. This applies to the settings of larger networks and fewer observations, inclusion and exclusion of respectively fixed-effect parameters and covariates, and noisy and contaminated data. I provide an empirical study on the regional housing prices in The Netherlands to analyze the inter-connectivity between housing corporation regions. The algorithm recovers a sparse asymmetric network structure. Multiple results reject the possibility of a contiguity-based underlying network. On regional level, urban regions located in the Randstad are less connected in the network, in contrast to the more rural regions. Furthermore, the results suggest four distinct periods in time wherein the spatial dependence alternates between high levels.

The remainder of my paper is structured as follows. Section 2 describes the model specification, assumptions and likelihood derivation. Furthermore it proposes the two-step Adaptive Lasso algorithm, including its pseudo-code. Section 3 outlines the simulation study and robustness checks, and discusses their results. Section 4 applies the algorithm to the study of regional housing prices in The Netherlands. Section 5 concludes my paper. The Appendix provides the derivations of the algorithm and tables with additional results of the simulation study.

2 Methodology

2.1 Model Specification

Panel data entails finite consecutive observations $t = 1, \dots, T$ for a set of individuals $i = 1, \dots, N$. The structural nonlinear panel data model is represented as

$$y_{it} = \alpha_i + \lambda \sum_{j=1, j\neq i}^{N} w_{ij} y_{jt} + \boldsymbol{\beta}' \mathbf{x}_{it} + \varepsilon_{it}, \qquad i = 1, \ldots, N, \quad t = 1, \ldots, T, \quad (1)$$

where y_{it} is an outcome and ε_{it} is an unobserved and normally distributed error term. I incorporate k covariates by allowing the covariate parameter $\boldsymbol{\beta}$ be a $(k \times 1)$ vector. Scalar α_i is a fixed-effect parameter representing the unobserved heterogeneity across individuals. Linkage of individual i to individual j is captured by the pair-specific weighted parameter w_{ij}, which is the (i, j)th element of the spatial weighting matrix $\mathbf{W}$. Scalar λ is a constant spatial autocorrelation coefficient that represents the general magnitude of the linkage. The term $\lambda \sum_{j=1, j\neq i}^{N} w_{ij}$ denotes the spatial dependence for individual i.

I create time variation in the linkage by including an unknown number of common breaks m at unknown break points $\{\mathcal{T}_1, \ldots, \mathcal{T}_m\}$ in the data. The breaks are assumed to only appear in the spatial coefficients. In model (1) I modify $\lambda = \lambda_{s_t}$ for $t = \mathcal{T}_{s_t - 1}, \ldots, \mathcal{T}_{s_t} - 1$ and $s_t = 1, \ldots, m + 1$. As a consequence, the structural model is formulated as

$$y_{it} = \alpha_i + \lambda_{s_t} \sum_{j=1, j\neq i}^{N} w_{ij} y_{jt} + \boldsymbol{\beta}' \mathbf{x}_{it} + \varepsilon_{it}, \qquad i = 1, \ldots, N, \quad t = 1, \ldots, T. \quad (2)$$

I assume $N, T \to \infty$ with condition $N^2/T \to 0$. Furthermore, m is considered fixed and much smaller than the time dimension ($m << T$) [1]. I set $\mathcal{T}_0 = 1$ and $\mathcal{T}_{m+1} = T + 1$, assume $\mathcal{T}_1 \geq 2$ and allow $\mathcal{T}_m = T$. I restrict break intervals to a minimum of five observations (see step-wise procedure 11 in Section 2.3.2).

The structural model (2) is written in matrix form as

$$\mathbf{y}_t = \boldsymbol{\alpha} + \lambda_{s_t} \mathbf{W} \mathbf{y}_t + \mathbf{X}_t \boldsymbol{\beta} + \boldsymbol{\varepsilon}_t, \qquad t = 1, \ldots, T, \quad (3)$$

where vector $\mathbf{y}_t = (y_{1t}, \ldots, y_{Nt})'$ collects the individual outcomes at t, structured similarly for $\boldsymbol{\varepsilon}_t$. The matrix of covariates $\mathbf{X}_t = (\mathbf{x}_{1t}, \ldots, \mathbf{x}_{Nt})'$ is a $(N \times k)$ variable. The weighting matrix $\mathbf{W}$ is of dimension $(N \times N)$, vector of fixed-effect parameters $\boldsymbol{\alpha}$ of $(N \times 1)$, $\boldsymbol{\beta}$ of $(k \times 1)$ and λ_{s_t} a scalar. The term $\lambda_{s_t}\mathbf{W}$ indicates the overall spatial dependence.

Several assumptions are needed to make model (3) suitable for estimation. I

[1] If $m \to T$, the spatial dependence is assumed to change at (nearly) every time instance. This is undesired and hence unconsidered, given the increasing optimization burden and evidence of periods of constant spatial dependence (de Paula et al., 2018; Billé et al., 2019)

specify the following assumptions:

Assumptions. **i)** $w_{ii} = 0$ for $i = 1, \dots, N$; **ii)** $0 \leq w_{ij} \leq 1$ for $i, j = 1, \dots, N$ and $i \neq j$; **iii)** $\sum_{j=1, j\neq i}^{N} w_{ij} \leq 1$ for $i = 1, \dots, N$; **iv)** $0 < \lambda_{s_t} < 1$, for $t = \mathcal{T}_{s_t - 1}, \dots, \mathcal{T}_{s_t} - 1$ and $s_t = 1, \dots, m+1$; **v)** $\exists\, i$ such that $\sum_{j=1, j\neq i}^{N} w_{ij} = 1$; **vi)** $\varepsilon_{it} \sim i.i.d.N(0, \sigma^2)$ for $i = 1, \dots. N$ and $t = 1, \dots, T$.

Assumption i) prohibits an individual to be interacted with itself, standard in spatial network theory. Assumptions ii) and iii) keeps the matrix elements stable in a small interval. Assumption iv) prevents the network structure from disappearing ($\lambda_{s_t} = 0$) and the series from becoming non-stationary[2] ($\lambda_{s_t} \geq 1$). Let $\mathbf{I}_N$ denote the $(N \times N)$ identity matrix. Together Assumptions ii) - iv) imply a non-negative network structure and ensure that $(\mathbf{I}_N - \lambda_{s_t}\mathbf{W})$ is a non-singular matrix and thus invertible, needed for a feasible reduced form. Moreover, the overall spatial dependence $\lambda_{s_t}\mathbf{W}$ is always less or equal to λ_{s_t}, i.e. $||\lambda_{s_t}\mathbf{W}||_\infty \leq \lambda_{s_t} < 1$ with $||\cdot||_\infty$ the infinity norm of a matrix. Assumption v) ensures that at least one row of $\mathbf{W}$ is equal to one, a necessary condition for parameter identification. However implementing Assumption v) in the optimization algorithm is complicated while keeping the gradients of the objective function differentiable. Plus, successfully implementing the assumption renders the algorithm slow. Therefore I post-implement the assumption as specified in Section 2.3.1. As a result λ_{s_t} and $\mathbf{W}$ are not theoretically identified and their estimates should be interpreted with caution. However the break pattern in λ_{s_t} hence break detection is not distorted. Lastly, Assumption vi) provide conditions on the error terms to obtain the log-likelihood of model (3).

2.2 Likelihood Derivation

The log-likelihood of model (3) can be written as

$$\begin{aligned} \ell(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = &- \frac{T}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{t=1}^{T} \boldsymbol{\varepsilon}_t \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t + \sum_{t=1}^{T} \log |\mathbf{I}_N - \lambda_{s_t}\mathbf{W}|, \\ \boldsymbol{\varepsilon}_t = &\mathbf{y}_t - \boldsymbol{\alpha} - \lambda_{s_t}\mathbf{W}\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta}, \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{m+1})$ and parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N+k+1} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2\}$, with a total of $q = N(N-1) + (m+1) + N + k + 1$ distinct parameters. Here $|\cdot|$ denotes the determinant of a matrix. The term $\sum_{t=1}^{T} \log |\mathbf{I}_N - \lambda_{s_t}\mathbf{W}|$ in the

[2]Terminologically speaking, model (2) can be seen as a piece-wise stationary model. The break-separated groups are stationary, but the concatenated model can but does not have to be stationary in presence of breaks.

log-likelihood expresses the Jacobian term of the transformation of $\boldsymbol{\varepsilon}_t$ to $\mathbf{y}_t$, due to the endogeneity of $\lambda_{s_t}\mathbf{W}\mathbf{y}_t$ (Elhorst, 2014). Because of Assumption vi), the variance-covariance matrix of the error term $\boldsymbol{\Sigma}$ equals $\sigma^2\mathbf{I}_N$. As a consequence term $-\frac{T}{2}\log|\boldsymbol{\Sigma}|$ in the first line of Equation (4) simplifies to $-\frac{NT}{2}\log\sigma^2$. Including $\boldsymbol{\alpha}$ can cause asymptotic MLE problems due to the incidental parameter problem (IPP). The intuition is that MLE maximizes the log-likelihood over $\boldsymbol{\alpha}$ along with all other parameters $(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta})$. When N becomes large compared to T, the randomness of the nonlinear model is not averaged out through $\boldsymbol{\alpha}$. My simulation study will focus on settings where N is substantially smaller than T and I apply $N^2/T \to 0$, hence no further actions concerning IPP will be taken.

2.3 Two-Step Adaptive Lasso Algorithm

Sections 2.3.1 - 2.3.3 describe the two-step Adaptive Lasso algorithm. I use a preliminary optimization of model (3) to obtain initial estimates $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$ with constant spatial dependence $\hat{\boldsymbol{\lambda}}^{(0)} = \hat{\lambda}^{(0)}$. In the first step I locate the optimal number of breaks $\hat{m}^{(2)}$ and their break points $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(2)}}$. In the second step, I use a regularized MLE to recover the time-varying network structure and to estimate all parameters $(\hat{\mathbf{W}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}})$. Below I describe the pseudo-code of the algorithm. Details of parameters, variables and models as well as motivation are found in the next sections.

Algorithm 1: Two-Step Adaptive Lasso Algorithm

Preliminary Step

Input : N/A

Output: Preliminary parameter estimates $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$.

Initialize $(\mathbf{W}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)})$. Specify penalty parameter p_0.

Estimate model (3) with $\lambda_{s_t} = \lambda$ for $t = \mathcal{T}_{s_t-1}, \ldots, \mathcal{T}_{s_t} - 1$ and $s_t = 1, \ldots, m+1$, by minimizing problem (5) using MLE.

First Step

Input : Preliminary parameter estimates $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$.

Output: Optimal number of breaks $\hat{m}_2$ and break points $\{\mathcal{T}_s\}_{s=1}^{\hat{m}_2}$.

Mean shift $\mathbf{y}_t \to \mathbf{y}_t^*$ and use $\boldsymbol{v}_i^*$. Calculate $\mathbf{K}_i^*$ and mean center columns of $\mathbf{K}_i^*$. Specify penalty parameter p_1.

Estimate model (9) by minimizing problem (10) using the LARS algorithm.

Restrict $\hat{\boldsymbol{\gamma}}_1$ to only allow for one break per interval of five observations, using rolling-window concatenation.

Normalize $\hat{\boldsymbol{\gamma}}_1$ and truncate values below threshold 0.05.

Scale $\boldsymbol{v}_i^* \to \tilde{\boldsymbol{v}}_i^*$. Calculate $\tilde{\mathbf{D}}_i^*$ and mean center columns of $\tilde{\mathbf{D}}_i^*$. Specify penalty parameter η.

Estimate model (12) by minimizing problem (13) using the LARS algorithm.

Normalize $\hat{\boldsymbol{\gamma}}_2$.

Determine the optimal subset of $\hat{m}^{(2)}$ by solving problem (14).

Second Step

Input : Preliminary parameter estimates $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$, optimal number of breaks $\hat{m}_2$ and break points $\{\mathcal{T}_s\}_{s=1}^{\hat{m}_2}$.

Output: Optimal parameter estimates $(\hat{\mathbf{W}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}})$.

Initialize $(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ using $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$. Set $\hat{\mathbf{W}}^{(0)}$ as adaptive weights. Use $\{\mathcal{T}_s\}_{s=1}^{\hat{m}_2}$ to set up λ_{s_t}. Specify penalty parameter p_2.

Estimate model (3) by minimizing problem (5) using MLE.

2.3.1 Preliminary step: Initial Parameter Estimates

As this step only functions as a prior guess and break detection is yet to be done, I fix $\lambda_{s_t} = \lambda$ for $s_t = 1, \ldots, m+1$. The same reasoning substantiates the use of a Lasso regularization instead of an adaptive version. The preliminary optimization problem is denoted as

$$(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})(p_0) = \underset{(\mathbf{W}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)})}{\text{argmin}} \quad \ell(\mathbf{W}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)}) + p_0 \sum_{i,j=1, i\neq j}^{N} |w_{ij}^{(0)}|, \tag{5}$$

where $\ell(\mathbf{W}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)})$ as in Equation (4) with $\boldsymbol{\lambda}^{(0)} = \lambda^{(0)}$ and $\boldsymbol{\theta}^{(0)} = \{\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}, \sigma^{2(0)}\}$. Scalar p_0 is the penalty term tuning parameter and $|\cdot|$ the absolute value term. The Lasso penalty term regularizes the elements of $\mathbf{W}$ by shrinking ineffective elements towards zero. Inevitably, the parameter estimates, especially $\hat{\lambda}^{(0)}$, are biased. However $\hat{\mathbf{W}}^{(0)}$ still approximates the network structure. Preliminary simulations demonstrate that this approximation is sufficient for accurate break detection in the next step. I deploy the Interior-Point algorithm[3] to optimize problem (5) using MLE with Lasso regularization. This algorithm tackles high-dimensional and sparse constrained nonlinear convex problems, which conform model (5). The algorithm is built in MATLAB function *fmincon*. In the *options* attribute of *fmincon* I specify the gradient of the objective function for faster computation. The absolute value bracket in $|w_{ij}^{(0)}|$ has no effect due to Assumption ii), and thus all first-order derivatives exist. Appendix A.1 provides the first-order of optimization problem (5) with respect to the parameters in $(\mathbf{W}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)})$. I post-implement row-normalization of Assumption v) as follows. Suppose row $\hat{\imath}$ of $\hat{\mathbf{W}}^{(0)}$ has maximum row-summation of $c = \sum_{j=1,\hat{\imath}\neq j}^{N} \hat{w}_{\hat{\imath}j}^{(0)}$. I divide all matrix elements in $\hat{\mathbf{W}}^{(0)}$ by c and I multiply $\hat{\lambda}^{(0)}$ with c. The post-normalization is not optimal, but $\hat{\mathbf{W}}^{(0)}$ is time-invariant and stays so, whereas $\hat{\lambda}^{(0)}$ is identified up to scale c.

2.3.2 First Step: Break Detection

It is essential to not underestimate m. If done, certain λ_t parameters are incorrectly grouped together in λ_{s_t} rendering inconsistency among parameters. To find m and $\{\mathcal{T}_s\}_{s=1}^{m}$ in λ_{s_t} I propose the use of univariate time series model

$$\boldsymbol{\upsilon}_i = \mathbf{K}_i \boldsymbol{\gamma}^{(1)} + \boldsymbol{\varepsilon}_i, \qquad i = 1, \ldots, N, \tag{6}$$

where $\boldsymbol{\upsilon}_i = (y_{i1}, \ldots, y_{iT})'$ entails all outcomes for individual i and where $\boldsymbol{\varepsilon}_i$ is a $(T\times 1)$ vector of residuals. $\mathbf{K}_i$ a $(T\times T)$ lower-triangular matrix with on each row t the network values $\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_t$, where the $(1\times N)$ row vector $\mathbf{w}_{i:}$ refers to row i of $\mathbf{W}$. The columns of $\mathbf{K}_i$ are mean centered by subtracting $\hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_t$

[3] A thorough description of the algorithm is outlined in Byrd et al. (1999).

from each column element. An example of $\mathbf{K}_i$ is depicted in Appendix A.2. Consider $\boldsymbol{\gamma}^{(1)}$ as a $(T \times 1)$ proxy vector of the spatial coefficient relaxed over time (λ_t). I set $\gamma_1^{(1)} = \lambda_1$ and for $t = 2, \dots, T$ it holds

$$\gamma_t^{(1)} = \begin{cases} \lambda_{t+1} - \lambda_t, & \text{when } t = \mathcal{T}_s \text{ for some } s \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Relation (7) indicates that each non-zero parameter in $\hat{\boldsymbol{\gamma}}^{(1)}$ denotes a possible change in coefficient values. Estimated zeros are considered time-invariant observations. Main model (3) only allows time variation in λ_{s_t}, such that the time variation found in model (6) solely matches potential breaks in the spatial dependence.

As the spatial dependence is stable and piece-wise stationary, λ_{s_t} will only have a limited effect on the outcomes. Smaller break differences in the outcomes complicates the detection of breaks. When data is also centered, seen as $\boldsymbol{\alpha} = \mathbf{0}$,[4] the proposed model (6) lacks the capability to correctly detect breaks[5]. Therefore I propose an adjustment to model (6). I uncenter $\mathbf{y}_t$ by mean shifting the outcomes,

$$\mathbf{y}_t^* = \mathbf{y}_t + (\mathbf{I}_N - \hat{\lambda}^{(0)}\hat{\mathbf{W}}^{(0)})^{-1}\boldsymbol{\alpha}, \qquad t = 1, \dots, T, \tag{8}$$

with $\boldsymbol{\alpha} = \mathbf{10}$ for a reasonable mean level shift. This results in an enlargement of the spatial dependence and hence should simplify break detection. The uncentering is done for all t and thus independently of number of breaks and their break points. Consequently I adapt model (6) to

$$\boldsymbol{\upsilon}_i^* = \mathbf{K}_i^* \boldsymbol{\gamma}^{(1)} + \boldsymbol{\varepsilon}_i, \qquad i = 1, \dots, N, \tag{9}$$

where $\boldsymbol{\upsilon}_i^* = (y_{i1}^*, \dots, y_{iT}^*)$ and $\mathbf{K}_i^*$ are constructed like respectively $\boldsymbol{\upsilon}_i$ and $\mathbf{K}_i$ but here with $\mathbf{y}_t^*$. Relation (7) is not pertained as a consequence of the resulting enlargement of the mean level. Now the estimates $\hat{\boldsymbol{\gamma}}^{(1)}$ only capture the pattern in λ_{s_t}. But, as wanted, capturing the pattern is equal to detecting the breaks

[4]Nevertheless, generating data through the reduced form still creates a slight intercept in the outcomes due to the included covariates.

[5]Previous break detection literature that use the same break detection method either uses uncentered data and large break changes (Otto & Steinert, 2018) or linear and less restrictive model specifications (Chan et al., 2014; Safikhani & Shojaie, 2017). Both settings substantially enlarge the impact of the break on the outcomes. Naturally, this simplifies break detection. In my paper I make stability and stationarity assumptions and impose a nonlinear model specification. As a consequence, the breaks in the spatial dependence have much lower impact on the outcomes, which on its part complicates the break detection.

and thus the model is still competent for break detection.

Without regularization, every parameter in $\boldsymbol{\gamma}^{(1)}$ is estimated to non-zero: every point in time is a break ($m + 1 = T$). I use a Lasso penalty term for $\boldsymbol{\gamma}^{(1)}$ to distinguish the constant points in time from the varying ones. Formally, the parameter estimates are found by

$$\hat{\boldsymbol{\gamma}}^{(1)}(p_1) = \underset{\boldsymbol{\gamma}^{(1)}}{\operatorname{argmin}} \quad \sum_{i=1}^{N} ||\boldsymbol{v}_i^* - \mathbf{K}_i^* \boldsymbol{\gamma}^{(1)}||_2^2 + p_1 ||\boldsymbol{\gamma}^{(1)}||_1, \tag{10}$$

where $||\cdot||_1$ and $||\cdot||_2$ respectively the ℓ^1-norm ℓ^2-norm of a vector and scalar p_1 the tuning parameter. In line with the preliminary step but opposite to existing literature (Otto & Steinert, 2018), no adaptive weights are used[6]. The optimization problem (10) is linear and convex. Therefore the problem can be efficiently solved by modifying the Least Angle Regression (LARS) algorithm of Efron et al. (2004) to a Lasso version. The LARS algorithm uses a less greedy and much faster version than conventional model selection routines. This is here particularly useful considering the high-dimensional parameter space. I implement a loop over the number of breaks for generality.

Let $\hat{m}^{(1)}$ be the estimated number of breaks and $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(1)}}$ their estimated break points. Safikhani & Shojaie (2017) demonstrate that $\hat{m}^{(1)}$ is consistently overestimated ($\hat{m}^{(1)} > m$), causing efficiency loss. This is partly due to some $\hat{\boldsymbol{\gamma}}^{(1)}$ values are estimated close to zero yet are non-zero, specifying a false break. Furthermore, the algorithm disperses non-zero estimates $\hat{\boldsymbol{\gamma}}^{(1)}$ across the neighborhood of the true break point instead of only at the true break point itself. Both issues cause inaccuracy and inefficiency. I propose three actions to filter the falsely estimated breaks from the set of detected breaks. Firstly, I correct the scattering of non-zeros in $\hat{\boldsymbol{\gamma}}^{(1)}$. I restrict the potential breaks to only occur once in an interval of five consecutive time periods[7]. For $t = 1, \ldots, T-4$, I use

[6]The extensiveness of an additional optimization and the aim to not underestimate the number of breaks do not urge the need for adaptive weights here. Choosing the correct tuning parameter will generate consistency as well as asymptotic normality in the parameter estimates (Otto & Steinert, 2018), with or without adaptive weights.

[7]Preliminary simulations demonstrate that an interval length of five functions as an upper bound of the scattering without wrongly concatenating $\hat{\boldsymbol{\gamma}}^{(1)}$.

step-wise procedure

1) Indicate highest value in $(\hat{\gamma}_t^{(1)}, `\hat{\gamma}_{t+4}^{(1)})$ as true break $\hat{\mathcal{T}}$.

$$\hat{\mathcal{T}} = \underset{\mathcal{T}}{\text{argmax}}(|\hat{\gamma}_t^{(1)}|, \ldots, |\hat{\gamma}_{t+4}^{(1)}|)$$

2) Set parameter $\hat{\gamma}_{\hat{\mathcal{T}}}^{(1)}$ equal to the sum of the estimates in the interval.

$$\hat{\gamma}_{\hat{\mathcal{T}}}^{(1)} = \sum_{\tau=0}^{4} \hat{\gamma}_{t+\tau}^{(1)} \tag{11}$$

3) Set all other estimates of $\hat{\boldsymbol{\gamma}}^{(1)}$ in the interval to zero.

$$\hat{\gamma}_{\tau}^{(1)} = 0 \text{ for } \tau = t, \ldots, t+4 \text{ and } \tau \neq \hat{\mathcal{T}}$$

If an interval contains only zero estimates, I move to next t. The second step combines the full break impact into one value. Secondly, parameters $\hat{\boldsymbol{\gamma}}^{(1)}$ are sequentially normalized and truncated to zero for values below the threshold of 0.05.[8] Thirdly, the set of remaining non-zero estimates $\hat{\boldsymbol{\gamma}}^{(1)}$ are put in a correction layer of break detection to lastly pinpoint the true breaks. Unlike model (9), I use here a specification that includes all initial parameters to resemble model (3). I use $\hat{\boldsymbol{\alpha}}^{(0)}$ and $\hat{\boldsymbol{\beta}}^{(0)}$ from the preliminary step, which are here kept fixed and thus not estimated. Consider $\hat{m}^{(1)}$ and $\{\hat{\mathcal{T}}_s^{(1)}\}_{s=1}^{\hat{m}^{(1)}}$ as fixed. Let $\boldsymbol{\mathcal{X}}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iT})'$ be the $(T \times k)$ matrix of time series of the covariates for individual i. The new model takes form as

$$\tilde{\boldsymbol{\upsilon}}_i^* = \tilde{\mathbf{D}}_i^* \boldsymbol{\gamma}^{(2)} + \hat{\alpha}_i^{(0)} \boldsymbol{\iota} + \boldsymbol{\mathcal{X}}_i \hat{\boldsymbol{\beta}}^{(0)} + \boldsymbol{\varepsilon}_i, \qquad i = 1, \ldots, N, \tag{12}$$

where $\tilde{\mathbf{D}}_i^*$ a $(T \times (\hat{m}^{(1)}+1))$ block-diagonal matrix with on each column s a block of scaled network values $\frac{1}{\mathcal{T}_s - \mathcal{T}_{s-1}} \hat{\mathbf{w}}_{i:}^{(0)} \mathbf{y}_t^*$ for $s = 1, \ldots, \hat{m}^{(1)}+1$. Proxy vector $\boldsymbol{\gamma}^{(2)}$ is of dimension $((\hat{m}^{(1)}+1) \times 1)$. $\tilde{\boldsymbol{\upsilon}}_i^*$ is the scaled version of $\boldsymbol{\upsilon}_i^*$, and scaled in the same way as $\tilde{\mathbf{D}}_i^*$. The columns of $\tilde{\mathbf{D}}_i^*$ are also mean centered, as with $\mathbf{K}_i^*$. Appendix A.2 shows an example of $\tilde{\mathbf{D}}_i^*$.

Due to the construction of $\tilde{\mathbf{D}}_i^*$, model (9) resembles model (3) but with switched dimensions. In fact, given $\hat{m}^{(1)}$ and $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(1)}}$, $\boldsymbol{\gamma}^{(2)}$ is a nearly one-on-one proxy of λ_{s_t}. The slight difference is caused by the mean shift in $\tilde{\boldsymbol{\upsilon}}_i^*$ which affects $\hat{\boldsymbol{\gamma}}^{(2)}$. However this step focuses only on break detection such that potential biases in $\hat{\boldsymbol{\gamma}}_2$ are not of importance here. Unbiased parameter estimates are of interest in the second step. Parameter vector $\boldsymbol{\gamma}^{(2)}$ is estimated optimization

[8] Preliminary simulation results demonstrate that this threshold never filters out true breaks from $\hat{\boldsymbol{\gamma}}^{(1)}$.

problem

$$\hat{\boldsymbol{\gamma}}^{(2)}(\eta) = \underset{\boldsymbol{\gamma}^{(2)}}{\operatorname{argmin}} \quad \sum_{i=1}^{N} ||\tilde{\boldsymbol{v}}_i^* - \tilde{\mathbf{D}}_i^* \boldsymbol{\gamma}^{(2)} - \hat{\alpha}_i^{(0)} \boldsymbol{\iota} - \boldsymbol{\mathcal{X}}_i \hat{\boldsymbol{\beta}}^{(0)}||_2^2 + \eta ||\boldsymbol{\gamma}^{(2)}||_1, \tag{13}$$

where scalar $\eta = \frac{\log N \log T}{T}$ is a fixed tuning parameter. Optimization problem (13) is efficiently optimized using the Lasso modified LARS algorithm. Estimates $\hat{\boldsymbol{\gamma}}^{(2)}$ are only normalized and not truncated, due to the proxy's nature. Then, I implement an iterative algorithm that considers all possible subsets $\{\mathcal{T}_s\}_{s=1}^{m^{(2)}}$ from the candidate breaks $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(1)}}$ to identify the true breaks[9]. The optimal subset is chosen using a Bayesian information Criterion (BIC). Consider $0 \leq m^{(2)} \leq \hat{m}^{(1)}$ and $\{\mathcal{T}_s\}_{s=1}^{m^{(2)}} \in \{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(1)}}$. Mathematically,

$$\begin{aligned}(\hat{m}^{(2)}, \{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(2)}}) = \underset{m^{(2)}, \{\mathcal{T}_s\}_{s=1}^{m^{(2)}}}{\operatorname{argmin}} \log \left(\sum_{i=1}^{N} ||\boldsymbol{v}_i^* - \mathbf{D}_i^* \hat{\boldsymbol{\gamma}}^{(2)} - \hat{\alpha}_i^{(0)} \boldsymbol{\iota} - \boldsymbol{\mathcal{X}}_i \hat{\boldsymbol{\beta}}^{(0)}||_2^2 \right) \\ + (\hat{m}^{(2)} + 1)\omega,\end{aligned} \tag{14}$$

where $\omega = \frac{\log NT}{NT}$ following Safikhani & Shojaie (2017). Here the unscaled $\boldsymbol{v}_i^*$ and $\mathbf{D}_i^*$ are used. It is not guaranteed that the algorithm works in absence of breaks. In that case the algorithm consistently overestimate the number of breaks, resulting in a loss of efficiency. In all other cases the resulting optimal number of breaks $\hat{m}^{(2)}$ and their break points $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(2)}}$ are consistent estimates of the true underlying break formation.

2.3.3 Second step: Network Estimation

With $\hat{m}^{(2)}$ and $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(2)}}$ I advance to estimating model (3). Here I deploy MLE plus Adaptive Lasso regularization. I use an adaptive version to include prior information on the network structure. Fortunately, $\hat{\mathbf{W}}^{(0)}$ is suitable to function as the adaptive weights. The optimal solution for $(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ is found by the convex optimization problem

$$(\hat{\mathbf{W}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}})(p_2) = \underset{(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta})}{\operatorname{argmin}} \quad \ell(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta}) + p_2 \sum_{\substack{\{i,j=1, i \neq j: \\ \hat{w}_{ij}^{(0)} \neq 0\}}}^{N} \frac{|w_{ij}|}{|\hat{w}_{ij}^{(0)}|^{\kappa}}, \tag{15}$$

[9] Safikhani & Shojaie (2017) use a Backward Elimination Algorithm (BEA) to iteratively determine the optimal number of breaks and break points. Despite the reduction of computational time, BEA consistently underestimates the number of breaks plus inaccurately locates them in settings with an high number of breaks. Therefore I use the algorithm described here.

where $|\hat{w}_{ij}^{(0)}|^{-\kappa}$ for $i, j = 1, \ldots, N$ are the adaptive weights corresponding to the full network recovery penalty term and scalar p_2 is the tuning parameter. I set $\kappa = 1$ for direct impact of the weights on the parameter estimates. Parameter vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{\hat{m}^{(2)}+1})$ in the log-likelihood consists of $\hat{m}^{(2)} + 1$ grouped parameters. When p_2 increases, the regularization shrinks more $\mathbf{W}$ elements to zero[10]. Furthermore, the penalty term now has an additional condition in its summation, namely $\{i, j : \hat{w}_{ij}^{(0)} \neq 0\}$. This condition implies that each w_{ij}, if $\hat{w}_{ij}^{(0)} = 0$, does not need to be penalized again and are thus kept at zero. I deploy the same estimation procedure for optimization problem (15) as with problem (5). However in this step λ_{s_t} is fixed to the breaks and adaptive weights are used, such that the gradients are required to be recalculated and implemented. Appendix A.1 provides the first-order derivatives of problem (15) for correct and efficient optimization. I post-implement Assumption v) according to the same strategy outlined in Section 2.3.1. In this step each $\hat{\lambda}_{s_t}$ is equally scaled by c. The detected break points from the first step are still identified and their results are thus not changed.

3 Simulation Study

3.1 Set-Up

I conduct a series of Monte Carlo simulations to assess the performance of the two-step Adaptive Lasso algorithm. In the simulation set-up I consider three different network specifications, each referring to a well-known network topology type in practice. The networks differ in strength and dispersion of the links. For all networks I assume linkage sparsity and that each individual has at least one connection, i.e. only non-isolated nodes. Firstly, I random sample a (20×20) matrix $\mathbf{W}_1$ resembling a modified Erdos-Renyi graph. Individual i is linked to individual j with a probability of 10%, where multiple links per individual are allowed. The linkage is not necessary reciprocal, such that an asymmetric structure can be maintained. This specification can be seen as a partially meshed topology, due to the potential but not complete connectivity of an individual. Next, I simulate $\mathbf{W}_2$ to resemble a (20×20) directed hub and spoke network, where hubs

[10] Caner & Zhang (2014) stress that the use of penalization falsely shrinks small parameter estimates towards zero, when they are in fact non-zero. This false shrinkage leads to overfitting zero elements of the weighting matrix. The literature suggests to use values larger than $\frac{1}{\sqrt{N}}$ to circumvent this issue. However I use the network as specified in Section 3.1 and determine how many elements in the weighting matrix have a value lower than the threshold. Then I calculate the percentage of falsely shrunk weighting matrix elements. Its implications are discussed in the results of the simulation study.

are individuals and spokes are links. Individual 1 has reciprocal linkage with a select number of individuals $j = 2, \dots, N/5$, which on their part have incoming links with a share of the remaining individuals $l = N(j-1)/5+1, \dots, Nj/5$ for each $l > 1$. By plotting the network structure we can recognize the extended star topology. Thirdly, I specify a ring topology in $\mathbf{W}_3$ by imitating a (20×20) Queen's Contiguity matrix. Here linkage is only happening between individuals that are nearby in the weighting matrix. To be precise, each individual i neighbors both his predecessor $i-1$ and successor $i+1$, with individual $i = 0 = N$ and $i = N+1 = 1$. In this way the structure forms a circle of links. By definition, the links are reciprocal such that a symmetric distribution of the links in the weighting matrix is achieved. With an in-degree of two for each individual and hence only $\frac{2}{N-1}\%$ of possible connection uphold, this network is sparse. For each row in each weighting matrix I randomly assign one element to 0.70. If other elements are specified, I assign their values to equal values such that Assumption iii) is obeyed[11]. Despite that $\mathbf{W}_3$ is symmetric in linkage dispersion, it is, just as $\mathbf{W}_1$ and $\mathbf{W}_2$, an asymmetric matrix due to different element values of reciprocal links. The resulted network graphs and their matrix representation are shown in Section 3.2.

The networks are simulated simultaneously with the time-varying spatial coefficient λ_{s_t}, here for convenience described per time instance as λ_t. I specify different numbers of breaks. As time variation is of interest, I do not consider the case of $m = 0$ breaks. I use a moderate spatial dependence of $\lambda_t = \lambda_0 = 0.30$ for $t = 1, \dots, T$ and set the break step $\delta = 0.40$ for moderate change. Firstly, I incorporate $m = 1$ break by adding δ to λ_t at $\mathcal{T}_1 = \left\lfloor \frac{T}{2} \right\rfloor + 1$, where $\lfloor \cdot \rfloor$ denotes the greatest integer less than or equal to the number. Secondly, I include $m = 2$ breaks by respectively adding and subtracting δ from λ_t at $\mathcal{T}_s = \left\lfloor \frac{sT}{3} \right\rfloor + 1$ for $s = 1, 2$, such that the second break returns the coefficient to its former spatial dependence. Lastly the case of increasing share of breaks present in the data is considered. Let $m = \left\lfloor \frac{T}{15} \right\rfloor$ and use $\mathcal{T}_s = \left\lfloor \frac{sT}{m+1} \right\rfloor + 1$ for $s = 1, \dots, m$, where I sequentially alternate between adding and subtracting δ.

The panel data is simulated using its data generating process (DGP)

$$\mathbf{y}_t = (\mathbf{I}_N - \lambda_{s_t}\mathbf{W})^{-1}(\boldsymbol{\alpha} + \mathbf{X}_t\boldsymbol{\beta} + \boldsymbol{\varepsilon}_t), \tag{16}$$

[11] For instance, if the row contains two non-zero elements, then one element equals 0.70 and the other 0.30. In case of three non-zero elements, one random element equals 0.70 and the remaining elements have a value of 0.15. When only one element is specified, possible in both $\mathbf{W}_1$ and $\mathbf{W}_2$, the sum of the row equals that element's value of 0.70. Hence, the sum of the row can but does not have to equal one, differing to the network specification in de Paula et al. (2018). Regarding footnote 10, elements can therefore only be falsely shrunk when a row of W contain three or more elements.

where λ_{s_t} depends on the number of breaks, $\mathbf{W}$ specified as one of the three network specifications, $\boldsymbol{\alpha} = \mathbf{0}$ to consider centered data, $\mathbf{X}_t$ is generated from the multivariate $N(\mathbf{1}, \mathbf{I}_N)$ distribution and $\boldsymbol{\beta} = \mathbf{2}$, and $\boldsymbol{\varepsilon}_t$ from the $N(\mathbf{0}, \sigma^2\mathbf{I}_N)$ distribution with $\sigma^2 = 1$. Furthermore, I consider $N = 20$ individuals, $T = \{100, 200\}$ discrete observations and $k = 2$ covariates. Different (hyper)parameter initializations are discussed in Section 3.3.

I consider all combinations of parameters, each referred to as a setting. In each setting the parameters are kept fixed whereas the variables are resampled over 500 simulation runs. The tuning parameter vector $\boldsymbol{p}^* = (p_1, p_2, p_3)$ is optimized by minimizing a BIC while searching over a predetermined grid[12] in each run. To be precise, I use

$$\text{BIC} = \ell(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta}) + \hat{q}_{>0}\omega,$$

where $\hat{q}_{>0}$ the number of non-zero parameter estimates of $(\hat{\mathbf{W}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}})$ and $\omega = \frac{\log NT}{NT}$ a scale factor. For the linear optimization problems (10) and (13) the $\ell(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ in BIC simplifies to the logarithm of the fitted residual in the L^2-norm, similar to the criterion in problem (14).

Considering four grid values for each tuning parameter results in $3^4 = 81$ possible options of $\boldsymbol{p}^*$ in each run. This becomes computationally infeasible combined with the number of simulation runs. To reduce computation intensity, I optimize each tuning parameter sequentially and separately. This is justified as each tuning parameter connects to an individual step in the two-step algorithm[13]. That is, starting with p_0, the tuning parameter is optimized over a grid containing 4 different values, resulting in $\hat{p}_0$. This is done for p_1 and p_2 as well. Doing this reduces the possible options of $\boldsymbol{p}^*$ to $3 \times 4 = 12$ in each run. Despite the reduction, preliminary simulation results indicate that the major part of the computational burden originates from tuning p_0 and p_2. Moreover, tuning parameter p_1 is consistently chosen at a fixed value, regardless of the setting and number of simulation runs. Therefore, I propose to use an initial period of 50 simulation runs to calibrate the optimal $\hat{\boldsymbol{p}}^*$. Subsequently, I fix the tuning parameters in the remaining simulation runs to the median of $\hat{\boldsymbol{p}}^*$ over the calibration runs[14]. This procedure uses only one combination per run to

[12] I use grids $p_0 = [0.25, 0.50, 0.75, 1.00]$, $p_1 = [0, 0.01, 0.10, 0.25]$ and $p_2 = [0.25, 0.50, 0.75, 1.00]$, where the minimum (maximum) grid values are based on the points where lower (higher) tuning parameter values were not chosen during preliminary simulations.

[13] Preliminary simulations demonstrate similar $\hat{\boldsymbol{p}^*}$ when optimizing separately or when considering all possible options of $\boldsymbol{p}^*$. To not generalize this result, I discuss its impact in terms of parameter bias in the result section.

[14] de Paula et al. (2018) demonstrate that this procedure worsens the performance as the

optimize, which roughly decreases the running time by a 10-fold[15].

The parameters in the first step do not need to be initialized due to the used regression method. Contrarily, The use of MLE in the preliminary and second step requires initial parameter values. In the preliminary step I initialize $\mathbf{W}^{(0)}$ elements at $1/N$ while conforming Assumptions i) - iii) and v), $\lambda^{(0)}$ equals the moderate value 0.50 and the remaining parameters $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$, $\boldsymbol{\beta}^{(0)} = \mathbf{1}$ and $\sigma^{2(0)} = 1$. In the second step I use $(\hat{\mathbf{W}}^{(0)}, \hat{\boldsymbol{\lambda}}^{(0)}, \hat{\boldsymbol{\theta}}^{(0)})$ obtained in the preliminary step as initialization, where λ_{s_t} is randomly assigned values from the interval $(\hat{\lambda}^{(0)} \pm 0.30)$ while subjected to Assumption iv).

To measure the performance of the two-step Adaptive Lasso algorithm I separately assess the network recovery and break detection, followed by an overall model evaluation. Regarding network recovery, a prior visual inspection is performed on the estimated network graphs and their matrix representations for the setting with $N = 20$, $T = 100$ and $m = 1$. Then, I compare the estimated matrices to the true ones, both found in Section 3.2. These matrices map the dispersion and values of the true and estimated elements. To substantiate visual evaluation I propose summary and element-level statistics such as number of strong and weak elements, number of reciprocated elements, in- and out-degree distributions and individuals with highest degree, following network literature (de Paula et al., 2018). Furthermore, I calculate descriptive statistics to make comparisons across settings. In light of sparsity, I evaluate the number of zero elements in $\hat{\mathbf{W}}$ and compare it with those in $\mathbf{W}$. Let $\mathcal{Z}(\hat{\mathbf{W}})$ and $\mathcal{Z}(\mathbf{W})$ be the set of zero elements of $\hat{\mathbf{W}}$ and $\mathbf{W}$ respectively, $|\mathbf{m}|$ the cardinality of a vector $\mathbf{m}$ and $\mathbf{A} \cap \mathbf{B}$ the intersection matrix between two matrices $\mathbf{A}$ and $\mathbf{B}$. Then the ratio of the cardinality of correctly estimated zeroes and the cardinality of true zeroes is specified as

$$\Psi_0 = \frac{|\mathcal{Z}(\hat{\mathbf{W}}) \cap \mathcal{Z}(\mathbf{W})|}{|\mathcal{Z}(\mathbf{W})|}.$$

The closer the ratio gets to one, the more zero weighting matrix elements are recovered correctly. However, oversparsifying $\hat{\mathbf{W}}$ to a zero matrix can lead to $\Psi_0 = 1$, though being meaningless and incorrect. Therefore the counterpart of

train-test sample ratio is here in disadvantage to the majority of the simulation runs. Though this can be generalized to other methods, it is not guaranteed for two-step Adaptive Lasso algorithm. Therefore I discuss the potential implications in terms of parameter bias in the result section.

[15]The reduction is not equal to the decrease in combinations per iteration, as optimizing p_1 is not linearly time-consuming whereas p_0 and p_2 are.

correctly specified non-zero elements of $\hat{\mathbf{W}}$ is also needed to give purpose to both statistics. Define $\mathcal{Z}^c(\hat{\mathbf{W}})$ and $\mathcal{Z}^c(\mathbf{W})$ as the set of non-zero elements of $\hat{\mathbf{W}}$ and $\mathbf{W}$ respectively, such that

$$\Psi_{\mathbf{W}} = \frac{|\mathcal{Z}^c(\hat{\mathbf{W}}) \cap \mathcal{Z}^c(\mathbf{W})|}{|\mathcal{Z}^c(\mathbf{W})|}.$$

A high $\Psi_{\mathbf{W}}$ value indicates a high share of correct non-zero weighting matrix elements recovered. Weighting matrix elements falsely shrunk to zero affect Ψ_0 and $\Psi_{\mathbf{W}}$. To see whether this is present, I calculate the falsely shrunk estimates as a ratio of the total number of network elements below threshold $\frac{1}{\sqrt{N}}$, i.e.

$$\text{FSE}_{\mathbf{W}} = \frac{\sum_{i,j=1,i\neq j;0<w_{ij}<\frac{1}{\sqrt{N}}}^{N} \mathbb{1}(\hat{w}_{ij} = 0)}{\sum_{i,j=1,i\neq j}^{N} \mathbb{1}(0 < w_{ij} < \frac{1}{\sqrt{N}})},$$

where $\mathbb{1}(\cdot)$ is the indicator function. To assess the bias of the estimated elements, I propose the use of the Mean Absolute Deviation (MAD), defined as

$$\text{MAD}_{\mathbf{W}} = \frac{1}{N(N-1)} \sum_{i,j=1,i\neq j}^{N} |\hat{w}_{ij} - w_{ij}|.$$

Regarding break detection, I calculate the correctly detected $m = \{1, 2, \lfloor T/15 \rfloor\}$ breaks as a ratio of the simulation runs. Another statistic is the Hausdorff Distance (HD), which indicates how far a missing or extra break is located from the closest true break point (Huttenlocher et al., 1993; Qian & Su, 2016). Therefore HD functions as a general measure for the accuracy of the estimated breaks. The HD is calculated between $\{\mathcal{T}_s\}_{s=1}^{\hat{m}^{(2)}}$ and $\{\mathcal{T}_s\}_{s=1}^{m}$. and expressed as a percentage of the time dimension, i.e. $100 \times \text{HD}(\hat{\mathcal{T}}, \mathcal{T})/T$. Supplementing existing theory, I make a distinction between HD conditional (HD_{con}) and unconditional (HD_{unc}) on the correct number of breaks. This informs us whether miss-specifying the number of breaks has consequences on locating the breaks. Furthermore, I calculate the average bias of the estimated spatial coefficients as

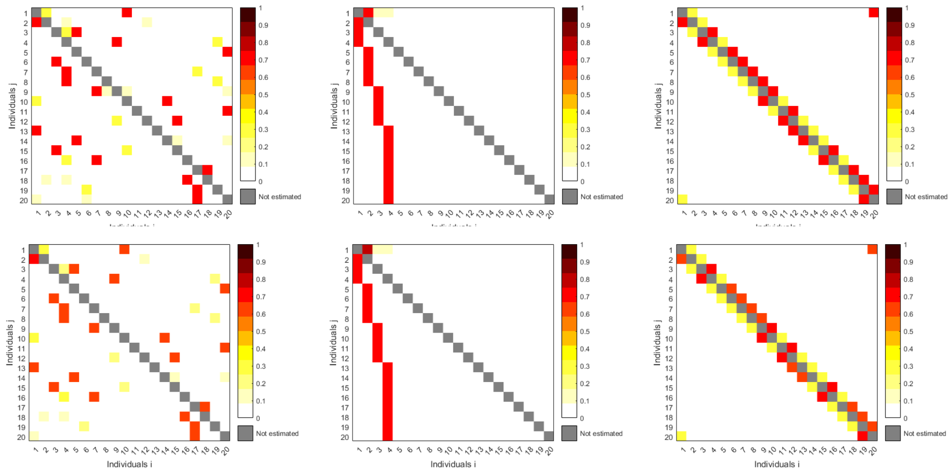
$$\text{MAD}_{\boldsymbol{\lambda}} = \frac{1}{T} \sum_{s_t=1}^{\hat{m}^{(2)}+1} \sum_{t=\mathcal{T}_{s_t-1}}^{\mathcal{T}_{s_t}-1} |\hat{\lambda}_{s_t} - \lambda_{s_t}|.$$

Lastly I evaluate the performance of the remaining parameters by their MAD. I do not use the mean estimated parameters as both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are respectively of dimension $N > 1$ and $k > 1$. The overall model is evaluated by means of the average Root Mean Squared Error (RMSE) of the estimated outcomes, that is

$$\text{RMSE}_y = \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(\hat{y}_{it} - y_{it})^2}.$$

All summary statistics and their standard deviations are reported as an average over the simulation runs.

3.2 Results



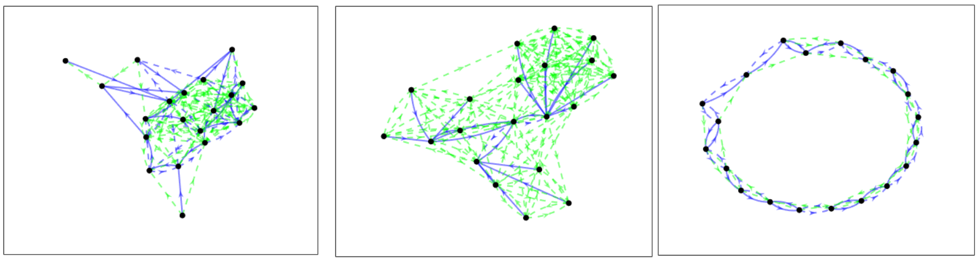
(a) Modified Erdos-Renyi **(b)** Hubs and Spokes **(c)** Queen's Contiguity

Figure 1: True spatial weighting matrices $\mathbf{W}$ (first row) and their estimated counterparts (second row)

Notes: First row displays the true spatial weighting matrices $\mathbf{W}$: (a) the modified Erdos-Renyi graph $\mathbf{W}_1$, (b) the Hubs and Spokes network $\mathbf{W}_2$ and (c) the Queen's Contiguity matrix $\mathbf{W}_3$. The second row displays the spatial weighting matrices estimated using the two-step Adaptive Lasso algorithm. The estimated matrices are rescaled to conform Assumption v). I use $N = 20$, $T = 100$, $m = 1$ and set parameters at $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\lambda_0 = 0.30$ and $\sigma^2 = 1$. The colors in the color bar indicate the value of the weighting matrix elements. Due to Assumption i) the diagonal elements are not estimated and colored grey.

Figure 1 shows the true and estimated matrices for the different network specifications. Overall, the pattern in true matrices looks identical to that of the estimated counterparts. The color bar indicates that high values are estimated high and low values are kept low by the algorithm. Only a few elements are slightly underestimated. This is promising, as it reveals the first evidence of the algorithm's ability to retrieve the network structure in a nonlinear model under presence of breaks. However there are more non-zero elements present than can be taken from Figure 1. Their values are very close to zero (majority equals

0.010 or less), such that the matrix elements are colored white. If an element is estimated at non-zero in only a few runs, it still has a substantive effect on the average over the runs. Put in perspective, the values are that low in magnitude that the outcomes are barely affected by this specific set of non-zero estimates. Figure 2 showcases the presence of all elements.



(a) Modified Erdos-Renyi **(b)** Hubs and Spokes **(c)** Queen's Contiguity

Figure 2: Network topologies of the estimated spatial weighting matrices

Notes: Directed graphs of the network topologies for the spatial weighting matrices $\mathbf{W}$: (a) the modified Erdos-Renyi graph $\mathbf{W}_1$, (b) the Hubs and Spokes network $\mathbf{W}_2$ and (c) the Queen's Contiguity matrix $\mathbf{W}_3$. All networks are estimated using the two-step Adaptive Lasso algorithm. The weighting matrices of the networks are rescaled to conform Assumption v). I use $N = 20$, $T = 100$, $m = 1$ and parameters $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\lambda_0 = 0.30$ and $\sigma^2 = 1$. Blue lines indicate kept elements: links that are estimated non-zero in at least 5% of the simulation runs and are also non-zero in the true network. Green lines indicate new elements: links that are estimated non-zero in at least 5% of the simulation runs but are zero in the true network. Red lines indicate removed elements: links that are estimated zero in at least 5% of the simulation runs but are non-zero in the true network. Furthermore, weak links are displayed with dashed lines and strong links are displayed with solid lines.

Each subfigure in Figure 2 displays the network topology of the corresponding specification. In all three topologies there are no red lines, indicating no removed edges in the estimated network and hence a perfect recovery of all non-zero weighting matrix elements. As a consequence, the blue lines indicate the true network formation. The estimated network is formed by both blue and green lines. These green lines are present in all formations, showcasing the fact of more estimated elements than seen in Figure 1. However, all green lines are dashed and close to zero. Regarding patterns, the Queen's Contiguity topology showcases the most clear resemblance to its true ring topology. The modified Erdos-Renyi and Hubs and Spokes topologies respectively mirror the partly meshed topology and extended star topology. Figure 2 is substantiated by Appendix Table B1. The estimated networks have a larger share of links, which are all estimated weak in linkage. This additional share also increases the in- and out-degree distributions of the estimated networks. However, it does not affect the number of reciprocal edges, which are equal for the true and corresponding estimated networks. With the low magnitude of the new elements I can conclude that network recovery is accurate for the investigated settings of $N = 20$, $T = 100$ and $m = 1$, with the Queen's Contiguity network as foremost.

The first part of Table 1 provides the descriptive statistics of the estimated networks in the various settings. The shares of correctly estimated zeros and non-zeros elements are high and consistent over time and number of breaks, outperforming Otto & Steinert (2018) and matching de Paula et al. (2018). It is worth noticing the average precision in recovering the correct zeros (98.0%) and non-zeros (98.6%) of the Queen's Contiguity network. This means that in total only 6 links are wrongly connected or omitted. The average bias in the network elements is low and shows a minimal pattern: the bias slightly increases with the number of breaks, with minimal differences across networks. Again, the algorithm seems to most accurately estimate the Queen's Contiguity network (estimated elements across settings on average 0.006 off). Possibly the symmetric structure simplifies network recovery. These conclusions are in line with the results from Figure 1 and 2. However in contrast to those figures, Table 1 indicates the presence of falsely shrunk estimates in the modified Erdos-Renyi and Hubs and Spokes networks. The explanation is that the descriptive statistics are calculated in each run, such that edges can be removed in some simulation runs but on average are present. The average values for the modified Erdos-Renyi (0.767) and Hubs and Spokes networks (0.304) partly explain the algorithm's deficiency to obtain the full 100% recovered non-zeros elements. The ratio diminishes when the time dimension is increased, confirming the presumption that more data aids network recovery. The Queen's Contiguity network has a ratio of zero as the matrix does not have elements with a value below threshold $\frac{1}{\sqrt{N}}$. Combined the descriptive statistics indicate that the algorithm is able to recover the network structure and accurately estimate the network elements.

The second part of Table 1 outlines the descriptive statistics of break detection and spatial coefficient estimation. The share of correctly estimated breaks indicate moderate to high values which differ across networks and time. The highest accuracy is when there are two breaks incorporated in the data. Furthermore, increasing the time dimension does not seem to improve the break detection. The Hubs and Spokes network has the most distinct pattern with relatively the lowest precision, followed by the modified Erdos-Renyi network. When the Queen's Contiguity network is used, the algorithm consistently detects any specified number of breaks and maintains an average of 92%. A closer

Table 1: Simulation results of descriptive statistics of the estimated networks and break detection using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi						$\mathbf{W}_2$: Hubs and Spokes						$\mathbf{W}_3$: Queen's Contiguity					
	$T = 100$			$T = 200$			$T = 100$			$T = 200$			$T = 100$			$T = 200$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
Network statistics																		
Ψ_0	0.948	0.930	0.919	0.952	0.952	0.861	0.935	0.916	0.881	0.943	0.937	0.791	0.979	0.971	0.974	0.991	0.984	0.985
	(0.014)	(0.018)	(0.045)	(0.018)	(0.014)	(0.062)	(0.017)	(0.023)	(0.059)	(0.017)	(0.019)	(0.048)	(0.009)	(0.009)	(0.010)	(0.008)	(0.013)	(0.025)
$\Psi_{\mathbf{W}}$	0.865	0.871	0.914	0.844	0.800	0.792	0.959	0.963	0.958	0.982	0.978	0.990	0.986	0.977	0.982	0.995	0.985	0.991
	(0.048)	(0.050)	(0.038)	(0.048)	(0.052)	(0.050)	(0.029)	(0.030)	(0.031)	(0.024)	(0.024)	(0.019)	(0.020)	(0.025)	(0.022)	(0.014)	(0.022)	(0.020)
$\text{FSE}_{\mathbf{W}}$	0.762	0.810	0.653	0.792	0.826	0.761	0.435	0.394	0.451	0.197	0.242	0.102	0.000	0.000	0.000	0.000	0.000	0.000
	(0.052)	(0.054)	(0.125)	(0.087)	(0.045)	(0.055)	(0.309)	(0.319)	(0.328)	(0.258)	(0.269)	(0.213)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
$\text{MAD}_{\mathbf{W}}$	0.012	0.013	0.014	0.010	0.010	0.023	0.009	0.012	0.017	0.007	0.008	0.028	0.007	0.008	0.007	0.004	0.005	0.005
	(0.002)	(0.003)	(0.007)	(0.003)	(0.002)	(0.009)	(0.002)	(0.003)	(0.009)	(0.002)	(0.002)	(0.007)	(0.002)	(0.002)	(0.002)	(0.001)	(0.002)	(0.004)
Break statistics																		
Correct Breaks Ratio	0.749	0.887	0.727	0.702	0.938	0.171	0.435	0.764	0.584	0.433	0.731	0.004	0.827	0.995	0.909	0.896	0.996	0.896
	(0.434)	(0.317)	(0.446)	(0.458)	(0.242)	(0.377)	(0.496)	(0.424)	(0.493)	(0.496)	(0.444)	(0.067)	(0.378)	(0.066)	(0.288)	(0.306)	(0.067)	(0.306)
HD_{unc}	4.700	1.600	6.400	4.500	0.600	13.000	14.000	3.200	9.000	10.700	2.300	18.400	2.700	0.100	0.100	0.500	0.000	1.400
	(10.300)	(4.000)	(0.100)	(10.400)	(2.800)	(11.600)	(15.700)	(5.000)	(10.300)	(14.200)	(4.900)	(11.100)	(7.900)	(0.900)	(0.300)	(2.200)	(0.200)	(3.500)
HD_{con}	0.100	0.500	1.000	0.100	0.200	0.700	0.600	1.100	1.800	0.400	0.600	2.000	0.010	0.000	0.100	0.010	0.000	0.300
	(0.300)	(0.800)	(1.000)	(0.300)	(0.300)	(0.400)	(0.500)	(1.300)	(1.300)	(0.600)	(0.700)	(1.400)	(0.200)	(0.0100)	(0.300)	(0.100)	(0.010)	(0.300)
$\text{MAD}_{\boldsymbol{\lambda}}$	0.039	0.048	0.096	0.029	0.025	0.207	0.034	0.042	0.103	0.034	0.028	0.189	0.030	0.031	0.032	0.025	0.025	0.043
	(0.032)	(0.024)	(0.088)	(0.027)	(0.014)	(0.114)	(0.015)	(0.022)	(0.074)	(0.015)	(0.013)	(0.036)	(0.016)	(0.013)	(0.015)	(0.019)	(0.014)	(0.055)

Notes: The unconditional and conditional HD statistics are in percentages of time dimension T. 500 simulation runs were performed. Standard errors are in parentheses.

investigation in each run shows that when $m = 1$ and $m = 2$ the algorithm occasionally overestimates the correct number of breaks. Though not desired, overestimation only causes inefficiency in the parameter estimation. That is, the bias in the spatial coefficients should not be affected too much. The MAD of the spatial coefficient confirms this assumption. The highest bias equals 0.048, whereas the lowest bias is only 0.025.

Regarding break points, the unconditional HD shows low percentages across settings. Even in simulation runs with incorrect estimated number of breaks the estimated break points are still close to the true points. For the Queen's Contiguity network the discrepancy is only 0.8% in terms of the time dimension. The unconditional HD values converge to the conditional HD counterparts when the percentage of correctly detected number of breaks increases. Moreover, the conditional HD statistics have near zero percentages with a maximum value of 2%. These results come close to Qian & Su (2016) which use a more simplified and static linear model. So, apart from two settings, the results indicates adequate performance of the algorithm to detect breaks and their break points.

Appendix Table B2 shows the descriptive statistics of the remaining parameters and the overall model performance. For all parameters it holds that the average bias increases along with the number of breaks. The Queen's Contiguity network performance here seems to be invariant across settings. I observe low but slightly higher biases in the fixed-effects parameters. Regardless, the biases in $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are low (unbiased to the first decimal) and showing consistency across settings. The estimated outcomes are on average 1 off, majorly due to the inaccurate fixed-effect parameters.

Taken together, the results suggest that the algorithm accurately recovers the underlying network structure as well as their link values. The same conclusion applies to break detection when one or two breaks are present. An increasing share of breaks is only accurately detected when the underlying network structure is symmetric. These findings only applies to one environment of parameter initializations. Next section shows that even higher break detection and network recovery accuracy can be obtained, possibly independently of each other, in various other simulation set-ups.

3.3 Robustness

As my work is the first in literature to propose the two-step Adaptive Lasso algorithm, it is desired to do robustness checks to guarantee the algorithm's wide applicability. I mainly focus on the network recovery and break detection, though making a comment about other parameters if worth the notice. I also make comparisons with the main results in Table 1. To keep it to a manageable degree, I deploy the different robust settings only to $\mathbf{W}_1$. This specification has a random asymmetric structure and thus is inherently not contiguity-based. In particular, I investigate when condition $N^2/T \to 0$ is less applicable which could affect parameter estimation. To do this I consider a higher-dimensional case by setting $N = 30$, which increases the number of parameters by 500. On the other hand, the case of limited time series data with $T = 50$.

Table 2: Simulation results of all descriptive statistics of settings with $\mathbf{W}_1$ and robustness checks $N = 30$ and $T = 50$ using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi								
	$N = 30$						$N = 20$		
	$T = 100$			$T = 200$			$T = 50$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
Network statistics									
Ψ_0	0.963	0.950	0.950	0.967	0.962	0.966	0.927	0.919	0.925
	(0.009)	(0.009)	(0.010)	(0.010)	(0.014)	(0.017)	(0.016)	(0.019)	(0.018)
$\Psi_{\mathbf{W}}$	0.633	0.570	0.582	0.598	0.656	0.670	0.736	0.740	0.734
	(0.042)	(0.042)	(0.044)	(0.044)	(0.044)	(0.049)	(0.046)	(0.049)	(0.043)
$\text{FSE}_{\mathbf{W}}$	0.897	0.869	0.867	0.893	0.889	0.890	0.784	0.792	0.786
	(0.020)	(0.017)	(0.018)	(0.016)	(0.019)	(0.022)	(0.057)	(0.061)	(0.053)
$\text{MAD}_{\mathbf{W}}$	0.011	0.013	0.012	0.010	0.010	0.009	0.016	0.020	0.017
	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.003)	(0.003)	(0.004)	(0.003)
Break statistics									
Correct Breaks Ratio	0.782	0.989	1	0.872	0.987	0.894	0.616	0.931	0.949
	(0.413)	(0.105)	(0.000)	(0.334)	(0.115)	(0.309)	(0.487)	(0.254)	(0.220)
HD_{unc}	3.900	0.100	0.100	1.600	0.100	1.300	10.900	1.500	1.800
	(9.100)	(0.800)	(0.400)	(6.500)	(1.500)	(2.900)	(15.200)	(4.000)	(3.500)
HD_{con}	0.000	0.100	0.100	0.100	0.000	0.400	0.400	0.500	1.100
	(0.300)	(0.300)	(0.400)	(0.200)	(0.100)	(0.400)	(1.200)	(1.200)	(1.700)
$\text{MAD}_{\boldsymbol{\lambda}}$	0.057	0.027	0.043	0.127	0.028	0.091	0.059	0.076	0.062
	(0.068)	(0.023)	(0.046)	(0.089)	(0.037)	(0.069)	(0.031)	(0.038)	(0.036)
Remaining statistics									
$\text{MAD}_{\boldsymbol{\alpha}}$	0.599	0.308	0.464	1.117	0.379	0.797	0.250	0.239	0.262
	(0.521)	(0.161)	(0.347)	(0.636)	(0.243)	(0.537)	(0.085)	(0.105)	(0.087)
$\text{MAD}_{\boldsymbol{\beta}}$	0.079	0.093	0.090	0.082	0.094	0.083	0.066	0.066	0.065
	(0.034)	(0.033)	(0.034)	(0.032)	(0.038)	(0.036)	(0.033)	(0.034)	(0.031)
MAD_{σ^2}	0.093	0.038	0.067	0.250	0.066	0.267	0.083	0.086	0.072
	(0.113)	(0.047)	(0.076)	(0.250)	(0.073)	(0.396)	(0.047)	(0.047)	(0.044)
RMSE_y	1.039	1.008	1.026	1.110	1.031	1.107	0.961	0.960	0.974
	(0.055)	(0.027)	(0.039)	(0.066)	(0.034)	(0.063)	(0.030)	(0.031)	(0.036)

Notes: Simulation study results for robustness checks. The unconditional and conditional HD statistics are in percentages of time dimension T. 500 simulation runs were performed. I use $\mathbf{W}_1$ and set $N = 20$, $T = \{100, 200\}$, $k = 2$, $m = \{1, 2, \lfloor \frac{T}{15} \rfloor\}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\sigma^2 = 1$, $\lambda_0 = 0.30$, $\delta = 0.40$ and $\rho = 0$. In the first six columns I imposed the change $N = 30$. In the last three columns I imposed the change $T = 50$. Standard errors are in parentheses.

Table 2 results contradict the stated assumption regarding $N^2/T \to 0$. In spite of a slight decrease in the share of recovered non-zero parameters the statistics are similar and thus promising. Actually, break detection is even better than in Table 1. Furthermore, the results demonstrate accurate network recovery and break detection when $T = 50$. Surprisingly, there is less bias in the fixed-effect parameters despite the reduce in distinct observations. So, the results showcase the algorithm's robustness, both when the size of the network increases or when there are less observations available.

Next I investigate what happens when the differences in spatial coefficients at the break points are small. Imposing $\delta = 0.20$ translates to a smaller break impact and possible tougher break detection. Moreover, differences around low (high) initial levels of spatial dependence are considered, that is $\lambda_0 = 0.10$ (0.90)[16]. Appendix Table B3 displays the results for settings with $\delta = 0.20$ and $\lambda_0 = \{0.10, 0.90\}$. The lower share of correct breaks and higher HD statistics agree with the assumption of tougher break detection. Still for $m = 2$ the break detection is fairly accurate. When the spatial dependence is low the algorithm is also less able to detect the breaks. Contrarily, high spatial dependence associates with easier break detection. But the table shows a more remarkable result. For all the settings in Appendix Table B3, the network recovery descriptive statistics are similar and some even outperform those in Table 1. From this can be concluded that full network recovery can still be accurate when breaks detection partly fails. Thus the two steps of the algorithm can work separately from and independently of each other, where failure of one step does not inherently lead to failure of the other.

As mentioned in Section 2.3.2, uncentered data simplifies the break detection hence more accurately estimate the parameters. I investigate the assumption that uncentered data simplifies break detection by setting $\boldsymbol{\alpha} = \mathbf{10}$. The case of absence of covariates is implemented by setting $\boldsymbol{\beta} = \mathbf{0}$. In these settings the outcomes are only generated by neighbouring outcomes, rendering even more centered data. Appendix Table B4 confirms the stated assumption. Actually, the share of correct breaks are close to 1 and the HD statistics just slightly off 0 indicating a near perfect break detection throughout settings with $\boldsymbol{\alpha} = \mathbf{10}$. Network recovery is slightly more imprecise, whereas network estimation improved. Break detection diminishes when $\boldsymbol{\beta} = \mathbf{0}$, in line with the assumption. However network recovery stays robust to these settings. Overall, the results in Appendix Table B4 mark the relation between accurate break detection and the degree of centered data.

To provide insight on how noisy data is handled by the algorithm, I consider the case of more volatile errors by setting $\sigma^2 = 2$. Furthermore, despite that MLE depends on homoscedastic errors due to Assumption vi), it is possible to have cross-sectional contamination in practice. So, I impose a correlation coefficient ρ between the error terms by setting the off-diagonals of $\boldsymbol{\Sigma}$ equal to $\rho = \{0.5, 1\}$.

[16] When $\lambda_0 = 0.90$ the spatial coefficient is generated by sequentially alternating between subtracting and adding δ from λ_0 at the given break points, reversely to the procedure for other values of λ_0 described in Section 3.1.

As seen in Appendix Table B5, network recovery and break detection is robust to settings with higher volatility in the errors. Furthermore, imposing either of the common shocks does not affect the summary statistics. Together the results illustrate that the algorithm's network recovery and break detection is highly robust in presence of noisy and contaminated data.

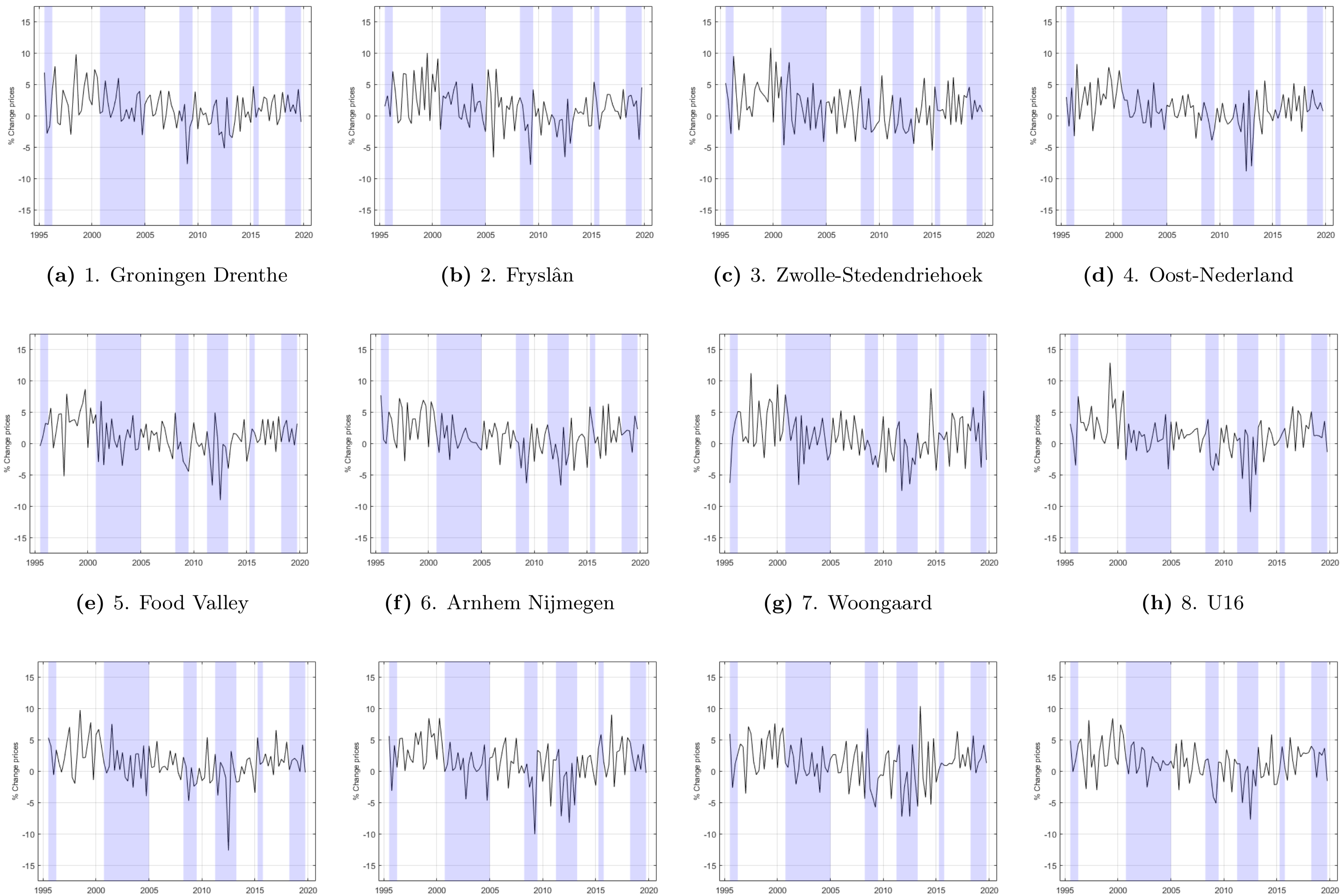
All robust settings considered, the results suggest that the two-step Adaptive Lasso algorithm is robust and satisfactorily recovers the network and detects the breaks. When there is nearly no spatial dependence, either with small break value difference or with low spatial coefficient value, the break detection worsens. Nevertheless the precision in network recovery is maintained throughout all settings. This applies to settings with larger networks and fewer observations, inclusion and exclusion of respectively fixed-effect parameters and covariates, and noisy and contaminated data.

4 Empirical Application

Housing prices are often described by spatial characteristics, from income level and unemployment rate to cultural activities and infrastructure level. If the characteristics are favorable, the housing prices tend to increase compared to regions where circumstances are less attractive. However the spatial characteristics are not always measurable and available. Another parameter is the price of houses in proximity. High surrounding housing prices generate more interest in the region which pushes up the demand hence prices of houses in proximity. Regions far away are less likely to affect the other regions pricing policy. However it is not guaranteed that these distance-based assumptions hold between all regions. Moreover, despite that the location of regions are time-invariant, the pricing policies between regions can become more or less inter-connected. The changes can happen as a consequence of nation-wide events, but also less evidently due to a regularization in the housing market. It is therefore useful to map the relations between regions without presuming a distance-based network *and* by allowing periods of change at unknown points in time.

4.1 Data

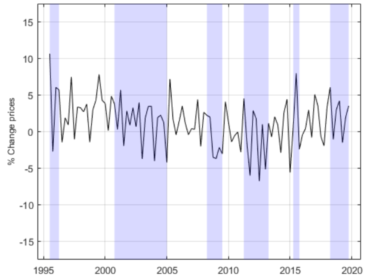
I focus on the regional housing prices in The Netherlands. In particular, I use the regions of all nineteen housing corporations present. Each corporation is restricted to and only engaged in its own region. The housing prices data is from



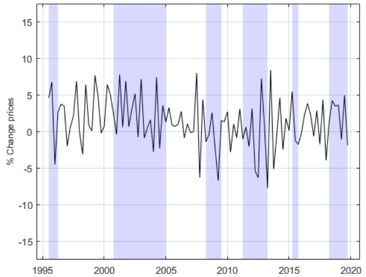
(a) 1. Groningen Drenthe
(b) 2. Fryslân
(c) 3. Zwolle-Stedendriehoek
(d) 4. Oost-Nederland
(e) 5. Food Valley
(f) 6. Arnhem Nijmegen
(g) 7. Woongaard
(h) 8. U16
(i) 9. Noord-Holland Noord
(j) 10. Metropolitan Amsterdam
(k) 11. Holland Rijnland
(l) 12. Haaglanden Rotterdam

Figure 3: Quarterly percentage change of housing prices in housing corporation regions 1-12 in The Netherlands

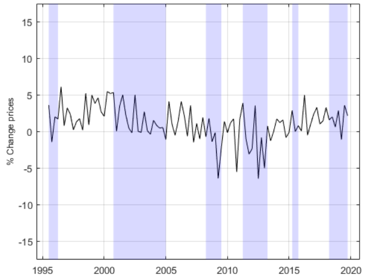
Notes: The data spans the period from 1995Q3 to 2019Q4 ($T = 98$). Blue shades bars are periods of recession in The Netherlands according to OECD based Recession Indicators for Netherlands from the Period following the Peak through the Trough (NDLREC).



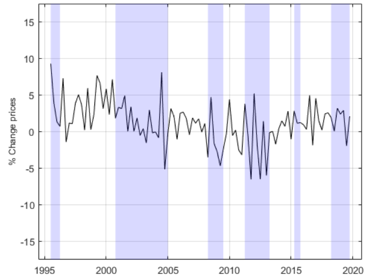
(a) 13. Drechtsteden Hoekse Waard



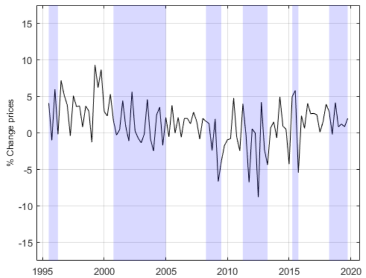
(b) 14. Zeeland



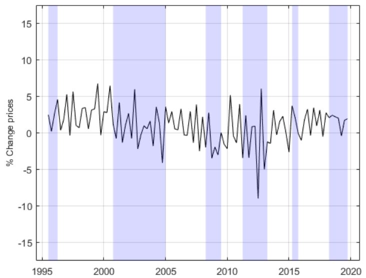
(c) 15. West-Brabant



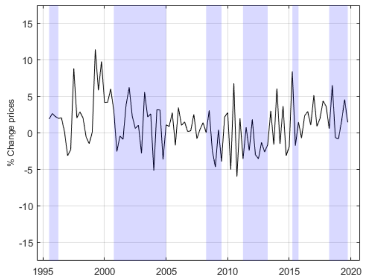
(d) 16. Noordoost Brabant



(e) 17. Metropolitan Eindhoven



(f) 18. Limburg



(g) 19. Amersfoort Zeewolde

Figure 4: Quarterly percentage change of housing prices in housing corporation regions 13-19 in The Netherlands

Notes: The data spans the period from 1995Q3 to 2019Q4 ($T = 98$). Blue shades bars are periods of recession in The Netherlands according to OECD based Recession Indicators for Netherlands from the Period following the Peak through the Trough (NDLREC).

the Central Bureau of Statistics (CBS) website[17], which tracks the prices of all real estate activities. The data spans the period 1995Q1 to 2019Q4 ($T = 100$). I convert the data to the quarterly percentage changes in housing prices to consider stationary data. By doing this the first observation is lost. Initially there are 40 sub-regions based on the COROP-division[18]. These regions are manually grouped together to resemble the nineteen regions of the housing corporations. Figure 3 and Figure 4 plot the series of the percentages changes in housing prices for each region. Low population density regions (e.g. Zwolle-Stedendriehoek, Drechtsteden Hoekse Waard) are less volatile compared to high density regions which combine the Randstad (Metropolitan Amsterdam, U16, Haaglanden Rotterdam). Furthermore, the lowest spikes are found in 2011 during the aftermath of the 'Great Recession' which started in 2008Q2. In Metropolitan Amsterdam the housing prices already dropped during the Great Recession. After 2013, housing prices increased consistently in all regions. Appendix Table B6 displays the summary statistics for each region. For all regions the mean percentage change is roughly 1. Moreover, Both the kurtosis (> 2.767) and skewness ($> |0.055|$) of all regions are larger those of the Normal distribution (both 0), indicating that each regional housing prices data has fat tails and is asymmetric shaped. However I maintain the assumption of normally distributed errors. The robustness checks demonstrate that the outliers should not affect break detection and network recovery.

Characteristics are not publicly available for these regions and the specified time span. Only the population level is known for each region, available on the CBS website[19]. The monthly data is converted to quarterly percentage growth by using the first differences of the logarithmic population levels. Furthermore, I also use the lagged percentage change in housing prices as a covariate. Due to this inclusion, one more observations is lost. So, the empirical study has a total of $N = 19$ individuals (regions), $T = 98$ observations (quarters) and $k = 2$ covariates (characteristics), similar to the main simulation set-ups. I use the proposed two-step Adaptive Lasso algorithm to estimate the model. In the results I solely discuss the break detection in *and* network recovery of the housing corporation market.

[17] https://www.cbs.nl/nl-nl/maatwerk/2020/14/prijsindex-bestaande-koopwoningen-naar-corop-gebied.

[18] The COROP-division are based on the regions with an unique regional core function (nodal principle in network theory), with commuter flows as the dominant factor.

[19] https://www.opendata.cbs.nl/statline/#/CBS/nl/dataset/70072ned/table?ts=1586249387453; https://www.opendata.cbs.nl/statline/#/CBS/nl/dataset/37230ned/table?ts=1586249411659

4.2 Results

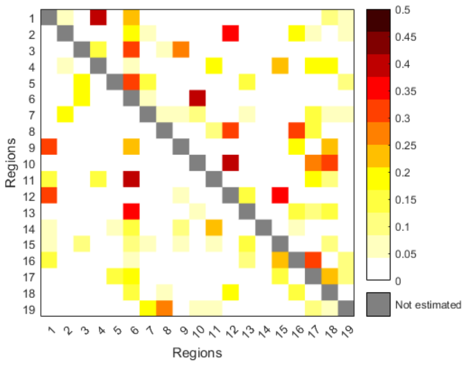
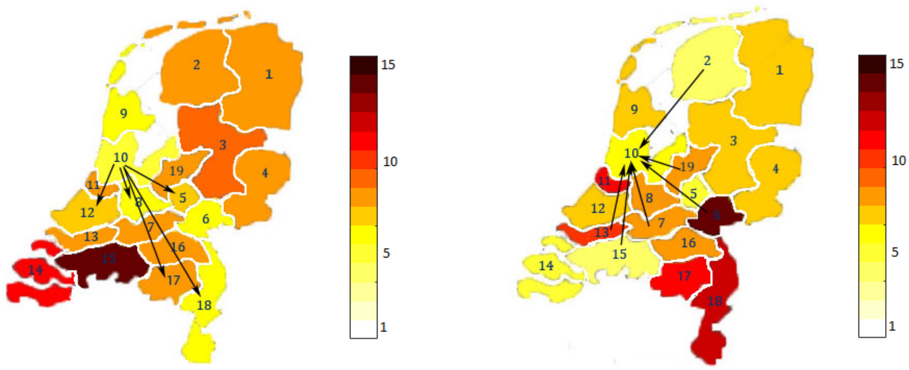


Figure 5: THE ESTIMATED SPATIAL WEIGHTING MATRIX

Notes: The empirical application considers $N = 19$, $T = 98$ and $k = 2$. The colors in the color bar indicate the value of the weighting matrix elements. Due to Assumption i) the diagonal elements are not estimated and colored grey.

Figure 5 shows the estimated weighting matrix for the housing market network. Notice that the color bar upper limit is set at 0.50 for clearer view of the estimated elements. The weighting matrix elements are estimated from low to moderate values. The algorithm recovers a sparse weighting matrix with 149 edges from the potential 342 edges. Thus not all housing market regions are connected with each other. Moreover, the weighting matrix does not show a diagonal pattern in a way similar to the Queen's Contiguity matrix (as in Figure 1), indicating that not all nearby regions have an estimated link between them. This result suggest that the underlying network is not contiguity-based as such specifications fix links between (nearby) regions. The only 44 reciprocal edges indicate an asymmetric network structure, which is on its part more proof that a symmetric contiguity-based network specification is not a correct fit in this application. On regional level, the out- and in-degree networks indicate the level of dependency between regions. Therefore I graph the number of outgoing and incoming links in the estimated network, shown in Figure 6.

Overall, the regions outside the Randstad are relatively more inter-connected in the housing market. To be specific, the regions Arnhem Nijmegen (6), Noordoost Brabant (16) and Limburg (18) have a low out-degree but high in-degree, meaning that the regions are highly influenced by other regions but on their part not affecting other regions. The vice versa applies to the more rural regions Zeeland (14) and West-Brabant (15). On the other hand, the main urban regions in the Randstad (U16 (8), Metropolitan Amsterdam (10) and Haaglanden



(a) Outgoing links

(b) Incoming links

Figure 6: Number of out- and in-degrees per region

Notes: The colors in the color bar indicate the number of the out- and in-degree. The arrows indicate in 6a the outgoing links and in 6b the incoming links of the Metropolitan Amsterdam region (10). The regions are based on the housing market corporations and numbered according to their region rank, defined in Appendix Table B6.

Rotterdam (12)) have both low out- and in-degrees. These regions are high in population and exhibit high housing prices. In these regions inner-region market fluctuations are expected to account for the housing price changes, rather than dependencies in the housing corporation market. The exemplary outgoing links from Metropolitan Amsterdam in Figure 6a are mainly to other densely populated regions such as U16, Haaglanden Rotterdam and Metropolitan Eindhoven (17). These regions are not all spatial neighbors of Metropolitan Amsterdam. Only one out of the five neighboring regions are influenced by the Metropolitan Amsterdam region, indicating the inappropriateness of imposing a spatial contiguity-based network. On the contrary, Figure 6b shows that Metropolitan Amsterdam is, with low link value, dependent on smaller regions, arguably from where citizens tend to move to Randstad and specifically Metropolitan Amsterdam.

Table 3: break-separated periods and their estimated spatial coefficient for the regional housing prices data in The Netherlands

	1	2	3	4
Periods	1995Q3-2002Q4	2003Q1-2007Q4	2008Q1-2015Q2	2015Q3-2019Q4
$\hat{\lambda}_{s_t}$	0.979	0.756	0.932	0.784

Notes: Breaks are detected at 2003Q1, 2008Q1 and 2015Q3.

Table 3 shows the periods divided by the three detected break points along with the estimated spatial coefficients. The detected breaks separate four alternating periods in time of high and moderately high spatial dependence. The spatial dependence stays fairly intact over time, though the differing spatial coefficient values still favor the inclusion of time variation. This is substantiated by the result that high spatial dependence ensures more accurate break detection, as shown in Section 3.3. Interestingly, the number of breaks is much lower than the number of starting and/or ending dates of indicated recessions by the OECD (shown in Figure 3 and Figure 4). Moreover, the break points cohere to some but not all recessions. In particular, the highly inter-connected first period runs until half of the first recession, from where the spatial dependence between regions diminishes. From 2008Q1, one quarter before the start of the Great Recession, the inter-connections on the housing markets strengthen again. However, no breaks are found in the first years after the Great Recession. This absence indicates that the aftermath recession running from 2011Q1 to 2013Q2, wherein many regions exhibit drops in the housing prices, does not readjust the intensity of the housing market. From 2015Q3 until 2019Q4 the spatial dependence between the regions again weakens yet stays on a high level.

Taken the break detection and network recovery together, the housing market exhibits a moderate to high spatial dependence between the different housing corporation regions. On regional-level there are disparities, particularly in the inter-connectivity of regions in the Randstad and others. Furthermore, enough evidence indicates that not spatial contiguity but an asymmetric dispersion is the base of the underlying network. Over time the algorithm locates four distinct periods, which take place independently of the indicated recessions.

5 Conclusion

Existing spatial network literature fails to jointly tackle the two problems of unknown network structure *and* absence of time variation in the spatial dependence. Therefore I extended previous model specifications to incorporate time variation in unknown network structures by means of structural breaks. For optimization I proposed a two-step Adaptive Lasso algorithm to sequentially detect multiple unknown breaks and fully estimate the underlying network. The use of MLE plus regularization enables to estimate the high-dimensional nonlinear model. An extensive simulation study considered different scenarios of sparse networks, number of breaks, spatial dependence magnitudes and combinations of (hyper)parameters. The results showcased the high accuracy in break detection and especially in network recovery. Furthermore, robustness checks indicated where the algorithm falls short. But, more importantly, it demonstrated the wide applicability of the algorithm. In particular, uncentered data and high spatial dependence are indicators of accurate break detection. Moreover, the algorithm's accuracy is still maintained when either break detection or network recovery fails. Finally, I applied the algorithm to the regional housing prices in The Netherlands. The results illustrated the time-varying inter-connectivity between housing corporation regions. Enough evidence rejects the possibility of a contiguity-based underlying network.

The inherent stationarity and stability assumptions of the spatial dependence render inaccurate break detection when break change is small or the spatial dependence is low. This pitfall with break detection amplifies when data is centered. In all cases the outcomes are only slightly affected by the breaks, which raises questions about the true relevance of break detection in such cases. Nevertheless, I proposed one pre-detection and three post-detection adjustments that partly circumvent the issues. It would be of interest to further develop the correct adjustments to obtain more accurate break detection in those specific settings.

Given that the algorithm accurately detects common breaks in the spatial dependence, we can also investigate instances with individual-specific breaks. Actually, Otto & Steinert (2018) already implemented this idea in a linear spatial model. Their results showcased that temporal change can affect only a subset of individuals or locations. I already successfully modified their theory to the nonlinear setting. It should therefore be possible to further adjust my algorithm to allow for individual-specific breaks in the spatial dependence and to accurately detect them.

References

Bai, J., & Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of applied econometrics*, *18*(1), 1–22.

Baltagi, B. H., Feng, Q., & Kao, C. (2016). Estimation of heterogeneous panels with structural breaks. *Journal of Econometrics*, *191*(1), 176–195.

Billé, A. G., Blasques, F., & Catania, L. (2019). Dynamic spatial autoregressive models with time-varying spatial weighting matrices. *Available at SSRN 3241470*.

Blume, L. E., Brock, W. A., Durlauf, S. N., & Ioannides, Y. M. (2011). Identification of social interactions. In *Handbook of social economics* (Vol. 1, pp. 853–964). Elsevier.

Blume, L. E., Brock, W. A., Durlauf, S. N., & Jayaraman, R. (2015). Linear social interactions models. *Journal of Political Economy*, *123*(2), 444–496.

Byrd, R. H., Hribar, M. E., & Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, *9*(4), 877–900.

Caner, M., & Zhang, H. H. (2014). Adaptive elastic net for generalized methods of moments. *Journal of Business & Economic Statistics*, *32*(1), 30–47.

Chan, N. H., Yau, C. Y., & Zhang, R.-M. (2014). Group LASSO for structural break time series. *Journal of the American Statistical Association*, *109*(506), 590–599.

Chandrasekhar, A. (2016). Econometrics of network formation. In *The oxford handbook of the economics of networks* (pp. 303–357). Oxford University Press Oxford.

de Paula, A., Rasul, I., & Souza, P. (2018). Recovering social networks from panel data: identification, simulations and an application. *Available at SSRN 3143442*.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407–499.

Elhorst, J. P. (2014). Spatial panel data models. In *Spatial econometrics* (pp. 37–93). Springer.

Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, *15*(9), 850–863.

Lee, L.-f. (2004). Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models. *Econometrica*, *72*(6), 1899–1925.

Lee, L.-f., & Yu, J. (2012). Qml estimation of spatial dynamic panel data models with time varying spatial weights matrices. *Spatial Economic Analysis*, *7*(1), 31–74.

Otto, P., & Steinert, R. (2018). Estimation of the spatial weighting matrix for spatiotemporal data under the presence of structural breaks. *Unpublished Manuscript. Cornell University, Ithaca (NY).*

Qian, J., & Su, L. (2016). Shrinkage estimation of common breaks in panel data models via adaptive group fused lasso. *Journal of Econometrics*, *191*(1), 86–109.

Qu, X., & Lee, L.-f. (2015). Estimating a spatial autoregressive model with an endogenous spatial weight matrix. *Journal of Econometrics*, *184*(2), 209–232.

Safikhani, A., & Shojaie, A. (2017). Joint structural break detection and parameter estimation in high-dimensional non-stationary var models. *Unpublished Manuscript. Cornell University, Ithaca (NY).*

A Derivations

A.1 Partial Derivatives

Denote the partial derivatives of the spatial weighting matrix as

$$\frac{\partial \mathbf{W}}{\partial w_{ij}} = \begin{pmatrix} \frac{\partial w_{11}}{\partial w_{ij}} & \cdots & \frac{\partial w_{1N}}{\partial w_{ij}} \\ \vdots & \ddots & \vdots \\ \frac{\partial w_{11}}{\partial w_{N1}} & \cdots & \frac{\partial w_{NN}}{\partial w_{ij}} \end{pmatrix},$$

with $\frac{\partial w_{kl}}{\partial w_{ij}} = 1$ if $k = i$, $l = j$ and zero otherwise.

The first-order derivatives of optimization problem (5) in the preliminary step are

$$\begin{aligned}
\frac{\partial \hat{\boldsymbol{\theta}}(p_0)}{\partial \boldsymbol{\alpha}^{(0)}} &= \sum_{t=1}^{T} \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_0)}{\partial w_{ij}^{(0)}} &= \left(\sum_{t=1}^{T} \left(\lambda^{(0)} \frac{\partial \mathbf{W}^{(0)}}{\partial w_{ij}^{(0)}} \mathbf{y}_t \right) \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t \right) - T\,\mathrm{Tr} \left(\left(\mathbf{I}_N - \lambda^{(0)} \mathbf{W}^{(0)} \right)^{-1} \lambda^{(0)} \frac{\partial \mathbf{W}^{(0)}}{\partial w_{ij}^{(0)}} \right) + p_0, \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_0)}{\partial \lambda^{(0)}} &= \left(\sum_{t=1}^{T} \left(\mathbf{W}^{(0)} \mathbf{y}_t \right) \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t \right) - T\,\mathrm{Tr} \left(\left(\mathbf{I}_N - \lambda^{(0)} \mathbf{W}^{(0)} \right)^{-1} \mathbf{W}^{(0)} \right), \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_0)}{\partial \boldsymbol{\beta}^{(0)}} &= \sum_{t=1}^{T} \mathbf{X}_t \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t, \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_0)}{\partial \sigma^{2(0)}} &= -\frac{T}{2} \left(\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}) - \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\Sigma}^{-1} \boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t \boldsymbol{\Sigma}^{-1} \right)_{11},
\end{aligned}$$

for $i, j = 1, \ldots, N$ and $i \neq j$.

The first-order derivatives of optimization problem (15) in the second step are

$$
\begin{aligned}
\frac{\partial \hat{\boldsymbol{\theta}}(p_2)}{\partial \boldsymbol{\alpha}} &= \sum_{t=1}^{T} \boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_t \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_2)}{\partial w_{ij}} &= \left(\sum_{t=1}^{T} \left(\lambda_{s_t} \frac{\partial \mathbf{W}}{\partial w_{ij}} \mathbf{y}_t \right) \boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_t - \mathrm{Tr}\left(\left(\mathbf{I}_N - \lambda_{s_t}\mathbf{W}\right)^{-1} \lambda_{s_t} \frac{\partial \mathbf{W}}{\partial w_{ij}} \right) \right) + \frac{p_2}{|\hat{w}_{ij}^{(0)}|^{\kappa}}, \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_2)}{\partial \lambda_{s_t}} &= \left(\sum_{t=\mathcal{T}_{s_t-1}}^{\mathcal{T}_{s_t}-1} \left(\mathbf{W}\mathbf{y}_t\right) \boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_t \right) - \mathrm{Tr}\left(\left(\mathbf{I}_N - \lambda_{s_t}\mathbf{W}\right)^{-1} \mathbf{W} \right), \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_2)}{\partial \boldsymbol{\beta}} &= \sum_{t=1}^{T} \mathbf{X}_t \boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_t, \\
\frac{\partial \hat{\boldsymbol{\theta}}(p_2)}{\partial \sigma^2} &= -\frac{T}{2}\left(\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}) - \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_t\boldsymbol{\varepsilon}_t\boldsymbol{\Sigma}^{-1} \right)_{11},
\end{aligned}
$$

for $i, j = 1, \ldots, N$, $i \neq j$, $t = \mathcal{T}_{s_t-1}, \ldots, \mathcal{T}_{s_t} - 1$ and $s_t = 1, \ldots, \hat{m}^{(1)} + 1$.

A.2 Matrix $\mathbf{K}_i$ & $\tilde{\mathbf{D}}_i^*$

Example of $\mathbf{K}_i \in \mathbb{R}^{T \times T}$ with $N = 3$ and $T = 5$.

$$
\mathbf{K}_i = \begin{pmatrix}
\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_1 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} & 0 & \ldots & 0 \\
\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_2 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} & \hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_2 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_5 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} & \hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_5 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}} & \ldots & \hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_5 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}
\end{pmatrix}
$$

Example $\tilde{\mathbf{D}}_i^* \in \mathbb{R}^{T \times (\hat{m}^{(1)}+1)}$ with $N = 3$, $T = 5$, $\hat{m}^{(1)} = 2$ and $\{\mathcal{T}_1, \mathcal{T}_2\} = \{2, 4\}$. Remember that $\mathcal{T}_0 = 1$, $\mathcal{T}_{\hat{m}^{(1)}+1} = T + 1$ and by Assumption i) $\hat{w}_{ii}^{(0)} = 0$ for $i = 1, \ldots, N$.

$$
\tilde{\mathbf{D}}_i^* = \begin{pmatrix}
\frac{1}{\mathcal{T}_1 - 1}\left(\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_1 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}\right) & 0 & 0 \\
0 & \frac{1}{\mathcal{T}_2 - \mathcal{T}_1}\left(\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_2 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}\right) & 0 \\
0 & \frac{1}{\mathcal{T}_2 - \mathcal{T}_1}\left(\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_3 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}\right) & 0 \\
0 & 0 & \frac{1}{T + 1 - \mathcal{T}_2}\left(\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_4 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}\right) \\
0 & 0 & \frac{1}{T + 1 - \mathcal{T}_2}\left(\hat{\mathbf{w}}_{i:}^{(0)}\mathbf{y}_5 - \hat{\mathbf{w}}_{i:}^{(0)}\bar{\mathbf{y}}\right)
\end{pmatrix}
$$

B Tables

Table B1: SIMULATION RESULTS OF SUMMARY AND ELEMENT-LEVEL STATISTICS OF THE TRUE AND ESTIMATED NETWORKS USING THE TWO-STEP ADAPTIVE LASSO ALGORITHM

	$\mathbf{W}_1$: Modified Erdos-Renyi		$\mathbf{W}_2$: Hubs and Spokes		$\mathbf{W}_3$: Queen's Contiguity	
	True	Estimated	True	Estimated	True	Estimated
Summary statistics						
Number elements	41	125	22	126	40	58
Number of strong elements	20	20	20	20	20	20
Number of weak elements	21	105	2	106	20	38
Number of reciprocated elements	0	0	0	0	20	20
element-level statistics						
In-degree distribution	2.05 (1.396)	6.25 (3.129)	1.1 (2.406)	6.3 (2.431)	2 (0)	2.900 (0.539)
Out-degree distribution	2.05 (0.805)	6.25 (3.534)	1.1 (0.436)	6.301 (2.076)	2 (0)	2.900 (0.436)
Individuals with highest out-degree	$\{9, 18, 20\}$	$\{8, 9, 20\}$	$\{1, 18^*, 20^*\}$	$\{2, 18, 20\}$	$\{1^*, 2^*, 3^*\}$	$\{1, 2, 3\}$

Notes: Results are based on the setting with $N = 20$, $T = 100$, $k = 2$, $m = 1$ and parameters $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\lambda_0 = 0.30$ and $\sigma^2 = 1$. I use the average estimated network over 500 simulation runs. * indicates that more individuals have the same out-degree. Standard errors are in parentheses.

Table B2: Simulation results of descriptive statistics of the remaining parameters and overall model performance using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi						$\mathbf{W}_2$: Hubs and Spokes						$\mathbf{W}_3$: Queen's Contiguity					
	$T = 100$			$T = 200$			$T = 100$			$T = 200$			$T = 100$			$T = 200$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
MAD_{α}	0.255	0.168	0.373	0.214	0.147	1.007	0.177	0.197	0.516	0.135	0.125	1.279	0.162	0.145	0.160	0.136	0.122	0.204
	(0.234)	(0.046)	(0.370)	(0.174)	(0.045)	(0.647)	(0.046)	(0.082)	(0.434)	(0.035)	(0.044)	(0.355)	(0.076)	(0.038)	(0.061)	(0.143)	(0.046)	(0.306)
MAD_{β}	0.078	0.071	0.067	0.074	0.067	0.077	0.057	0.061	0.062	0.049	0.051	0.055	0.047	0.050	0.051	0.059	0.063	0.065
	(0.031)	(0.030)	(0.027)	(0.024)	(0.022)	(0.026)	(0.025)	(0.024)	(0.026)	(0.018)	(0.017)	(0.018)	(0.027)	(0.027)	(0.028)	(0.031)	(0.028)	(0.031)
MAD_{σ^2}	0.039	0.039	0.123	0.041	0.027	0.427	0.040	0.045	0.148	0.024	0.028	0.459	0.038	0.035	0.039	0.028	0.025	0.098
	(0.049)	(0.028)	(0.132)	(0.050)	(0.023)	(0.198)	(0.029)	(0.034)	(0.120)	(0.020)	(0.027)	(0.072)	(0.034)	(0.024)	(0.035)	(0.055)	(0.024)	(0.159)
RMSE_y	1.000	0.992	1.044	1.016	1.008	1.190	0.989	0.992	1.062	1.002	1.004	1.206	0.992	0.990	0.996	1.000	1.000	1.038
	(0.030)	(0.023)	(0.071)	(0.025)	(0.015)	(0.084)	(0.022)	(0.027)	(0.063)	(0.015)	(0.019)	(0.030)	(0.022)	(0.018)	(0.025)	(0.027)	(0.016)	(0.074)

Notes: 500 simulation runs were performed. Standard errors are in parentheses.

Table B3: Simulation results of all descriptive statistics of settings with $\mathbf{W}_1$ and robustness checks $\delta = 0.20$ and $\lambda_0 = \{0.10, 0.90\}$ using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi																	
	$\delta = 0.20$						$\lambda_0 = 0.10$						$\lambda_0 = 0.90$					
	$T = 100$			$T = 200$			$T = 100$			$T = 200$			$T = 100$			$T = 200$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
Network statistics																		
Ψ_0	0.905	0.885	0.848	0.953	0.929	0.864	0.921	0.905	0.843	0.942	0.931	0.745	0.938	0.949	0.951	0.954	0.964	0.919
	(0.019)	(0.024)	(0.029)	(0.015)	(0.020)	(0.030)	(0.016)	(0.018)	(0.055)	(0.019)	(0.016)	(0.038)	(0.016)	(0.017)	(0.021)	(0.013)	(0.013)	(0.026)
$\Psi_{\mathbf{W}}$	0.787	0.813	0.848	0.902	0.793	0.878	0.801	0.736	0.836	0.812	0.825	0.855	0.716	0.708	0.781	0.721	0.772	0.673
	(0.050)	(0.047)	(0.054)	(0.038)	(0.051)	(0.046)	(0.048)	(0.052)	(0.050)	(0.043)	(0.050)	(0.041)	(0.055)	(0.039)	(0.049)	(0.051)	(0.046)	(0.088)
$\text{FSE}_{\mathbf{W}}$	0.762	0.741	0.682	0.672	0.780	0.680	0.753	0.812	0.672	0.750	0.735	0.601	0.806	0.805	0.773	0.807	0.788	0.849
	(0.041)	(0.040)	(0.071)	(0.059)	(0.035)	(0.048)	(0.053)	(0.048)	(0.127)	(0.060)	(0.045)	(0.074)	(0.059)	(0.040)	(0.041)	(0.032)	(0.047)	(0.049)
$\text{MAD}_{\mathbf{W}}$	0.019	0.024	0.029	0.011	0.016	0.023	0.019	0.025	0.032	0.013	0.014	0.042	0.014	0.011	0.012	0.011	0.009	0.024
	(0.003)	(0.004)	(0.004)	(0.002)	(0.003)	(0.003)	(0.003)	(0.003)	(0.009)	(0.003)	(0.002)	(0.005)	(0.004)	(0.003)	(0.003)	(0.003)	(0.002)	(0.008)
Break statistics																		
Correct Breaks Ratio	0.336	0.584	0.022	0.282	0.569	0.000	0.607	0.900	0.244	0.571	0.831	0.000	0.802	0.727	0.920	0.907	0.771	0.893
	(0.473)	(0.493)	(0.148)	(0.451)	(0.496)	(0.000)	(0.489)	(0.300)	(0.430)	(0.495)	(0.375)	(0.000)	(0.399)	(0.446)	(0.272)	(0.291)	(0.421)	(0.309)
HD_{unc}	17.000	8.300	25.800	15.500	4.000	34.200	8.200	1.300	18.000	7.200	1.700	26.600	6.500	4.800	1.900	3.500	3.300	7.600
	(15.800)	(8.000)	(14.800)	(16.400)	(5.100)	(17.100)	(13.100)	(3.700)	(13.800)	(12.700)	(5.000)	(16.000)	(14.000)	(7.900)	(3.500)	(12.000)	(7.500)	(23.300)
HD_{con}	1.200	0.500	0.250	0.900	1.200	N/A	0.300	0.400	1.300	0.200	0.300	N/A	0.100	0.700	1.000	0.000	0.300	0.200
	(1.600)	(6.400)	(1.500)	(1.300)	(1.400)	N/A	(0.700)	(0.700)	(1.000)	(0.400)	(0.500)	N/A	(0.400)	(1.000)	(1.000)	(0.100)	(0.500)	(0.300)
$\text{MAD}_{\boldsymbol{\lambda}}$	0.094	0.126	0.208	0.059	0.066	0.177	0.061	0.061	0.238	0.045	0.040	0.366	0.026	0.049	0.051	0.037	0.041	0.184
	(0.033)	(0.039)	(0.047)	(0.026)	(0.029)	(0.034)	(0.022)	(0.021)	(0.123)	(0.018)	(0.015)	(0.060)	(0.033)	(0.057)	(0.034)	(0.055)	(0.030)	(0.120)
Remaining statistics																		
$\text{MAD}_{\boldsymbol{\alpha}}$	0.253	0.318	0.661	0.156	0.165	0.587	0.165	0.139	0.660	0.195	0.109	1.383	0.452	0.656	0.501	0.602	0.533	2.371
	(0.059)	(0.112)	(0.193)	(0.031)	(0.041)	(0.146)	(0.080)	(0.043)	(0.396)	(0.143)	(0.056)	(0.297)	(0.311)	(0.557)	(0.261)	(0.480)	(0.248)	(1.242)
$\text{MAD}_{\boldsymbol{\beta}}$	0.056	0.053	0.057	0.038	0.041	0.039	0.061	0.068	0.060	0.051	0.044	0.055	0.129	0.127	0.113	0.138	0.129	0.223
	(0.022)	(0.021)	(0.022)	(0.016)	(0.016)	(0.016)	(0.024)	(0.026)	(0.023)	(0.018)	(0.017)	(0.020)	(0.047)	(0.047)	(0.044)	(0.047)	(0.050)	(0.072)
MAD_{σ^2}	0.050	0.048	0.093	0.022	0.023	0.162	0.046	0.044	0.182	0.026	0.024	0.438	0.186	0.276	0.274	0.214	0.197	1.225
	(0.031)	(0.032)	(0.046)	(0.016)	(0.018)	(0.032)	(0.029)	(0.027)	(0.100)	(0.019)	(0.018)	(0.057)	(0.382)	(0.226)	(0.200)	(0.258)	(0.119)	(0.663)
RMSE_y	0.976	0.985	1.044	0.993	0.997	1.077	0.981	0.983	1.080	0.999	0.996	1.198	1.067	1.117	1.119	1.090	1.086	1.455
	(0.018)	(0.025)	(0.023)	(0.011)	(0.014)	(0.014)	(0.020)	(0.019)	(0.056)	(0.016)	(0.014)	(0.024)	(0.054)	(0.093)	(0.083)	(0.074)	(0.048)	(0.216)

Notes: Simulation study results for robustness checks. The unconditional and conditional HD statistics are in percentages of time dimension T. 500 simulation runs were performed. I use $\mathbf{W}_1$ and set $N = 20$, $T = \{100, 200\}$, $k = 2$, $m = \{1, 2, \lfloor \frac{T}{15} \rfloor\}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\sigma^2 = 1$, $\lambda_0 = 0.30$, $\delta = 0.40$ and $\rho = 0$. In the first six columns I imposed the change $\delta = 0.20$. In the last twelve columns I imposed the change $\lambda_0 = \{0.10, 0.90\}$. Standard errors are in parentheses.

Table B4: Simulation results of all descriptive statistics of settings with $\mathbf{W}_1$ and robustness checks $\boldsymbol{\alpha} = \mathbf{10}$ and $\boldsymbol{\beta} = \mathbf{0}$ using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi											
	$\boldsymbol{\alpha} = 10$						$\boldsymbol{\beta} = 0$					
	$T = 100$			$T = 200$			$T = 100$			$T = 200$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
Network statistics												
Ψ_0	0.877	0.866	0.887	0.877	0.848	0.885	0.783	0.773	0.798	0.812	0.815	0.848
	(0.018)	(0.021)	(0.018)	(0.019)	(0.023)	(0.014)	(0.055)	(0.056)	(0.046)	(0.096)	(0.051)	(0.025)
$\Psi_{\mathbf{W}}$	0.375	0.436	0.470	0.376	0.398	0.448	0.587	0.538	0.610	0.675	0.628	0.708
	(0.067)	(0.054)	(0.052)	(0.070)	(0.053)	(0.052)	(0.08)	(0.093)	(0.085)	(0.081)	(0.070)	(0.071)
$\text{FSE}_{\mathbf{W}}$	0.849	0.813	0.869	0.840	0.803	0.880	0.733	0.699	0.687	0.737	0.719	0.704
	(0.069)	(0.040)	(0.035)	(0.066)	(0.038)	(0.029)	(0.091)	(0.088)	(0.083)	(0.106)	(0.083)	(0.067)
$\text{MAD}_{\mathbf{W}}$	0.051	0.049	0.043	0.051	0.054	0.046	0.053	0.057	0.052	0.042	0.047	0.040
	(0.007)	(0.006)	(0.006)	(0.007)	(0.006)	(0.005)	(0.006)	(0.005)	(0.006)	(0.008)	(0.006)	(0.005)
Break statistics												
Correct Breaks Ratio	0.922	1	1	0.940	0.998	1	0.036	0.320	0.000	0.027	0.276	0.000
	(0.268)	(0.000)	(0.000)	(0.238)	(0.047)	(0.000)	(0.185)	(0.467)	(0.000)	(0.161)	(0.447)	(0.000)
HD_{unc}	1.200	0.000	0.000	0.600	0.000	0.000	37.000	17.800	28.800	37.400	17.000	37.300
	(5.400)	(0.000)	(0.000)	(3.400)	(0.100)	(0.000)	(8.000)	(7.300)	(13.500)	(8.300)	(7.200)	(16.800)
HD_{con}	0.000	0.000	0.000	0.000	0.000	0.000	26.400	17.700	N/A	35.100	17.100	N/A
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(13.000)	(8.700)	N/A	(14.200)	(9.100)	N/A
$\text{MAD}_{\boldsymbol{\lambda}}$	0.117	0.096	0.100	0.115	0.097	0.126	0.210	0.275	0.305	0.169	0.220	0.253
	(0.034)	(0.037)	(0.029)	(0.040)	(0.040)	(0.030)	(0.063)	(0.064)	(0.042)	(0.049)	(0.051)	(0.038)
Remaining statistics												
$\text{MAD}_{\boldsymbol{\alpha}}$	2.634	1.829	1.789	2.630	1.848	2.215	0.086	0.086	0.086	0.060	0.060	0.060
	(0.703)	(0.714)	(0.538)	(0.834)	(0.762)	(0.550)	(0.017)	(0.016)	(0.017)	(0.013)	(0.012)	(0.012)
$\text{MAD}_{\boldsymbol{\beta}}$	0.157	0.119	0.122	0.162	0.103	0.137	0.021	0.022	0.021	0.016	0.015	0.015
	(0.072)	(0.057)	(0.046)	(0.061)	(0.047)	(0.048)	(0.013)	(0.015)	(0.013)	(0.012)	(0.011)	(0.010)
MAD_{σ^2}	0.800	0.526	0.627	0.886	0.668	0.886	0.058	0.048	0.042	0.042	0.030	0.022
	(0.176)	(0.117)	(0.120)	(0.165)	(0.134)	(0.148)	(0.034)	(0.030)	(0.029)	(0.038)	(0.020)	(0.020)
RMSE_y	1.333	1.230	1.271	1.366	1.281	1.367	0.975	0.980	0.984	0.986	0.987	0.995
	(0.057)	(0.044)	(0.049)	(0.057)	(0.048)	(0.051)	(0.023)	(0.020)	(0.021)	(0.022)	(0.013)	(0.014)

Notes: Simulation study results for robustness checks. The unconditional and conditional HD statistics are in percentages of time dimension T. 500 simulation runs were performed. I use $\mathbf{W}_1$ and set $N = 20$, $T = \{100, 200\}$, $k = 2$, $m = \{1, 2, \lfloor \frac{T}{15} \rfloor\}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\sigma^2 = 1$, $\lambda_0 = 0.30$, $\delta = 0.40$ and $\rho = 0$. In the first six columns I imposed the change $\boldsymbol{\alpha} = \mathbf{10}$. In the last six columns I imposed the change $\boldsymbol{\beta} = \mathbf{0}$. Standard errors are in parentheses.

Table B5: Simulation results of all descriptive statistics of settings with $\mathbf{W}_1$ and robustness checks $\sigma^2 = 2$ and $\rho = \{0.50, 1\}$ using the two-step Adaptive Lasso algorithm

	$\mathbf{W}_1$: Modified Erdos-Renyi																	
	$\sigma^2 = 2$						$\rho = 0.50$						$\rho = 1$					
	$T = 100$			$T = 200$			$T = 100$			$T = 200$			$T = 100$			$T = 200$		
	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$	$m = 1$	$m = 2$	$m = \lfloor \frac{T}{15} \rfloor$
Network statistics																		
Ψ_0	0.921	0.917	0.911	0.939	0.923	0.848	0.952	0.942	0.948	0.973	0.971	0.941	0.954	0.947	0.951	0.978	0.966	0.941
	(0.015)	(0.016)	(0.028)	(0.020)	(0.018)	(0.051)	(0.014)	(0.021)	(0.019)	(0.013)	(0.011)	(0.029)	(0.012)	(0.013)	(0.014)	(0.009)	(0.010)	(0.019)
$\Psi_{\mathbf{W}}$	0.783	0.740	0.842	0.819	0.768	0.863	0.792	0.840	0.810	0.906	0.808	0.803	0.769	0.780	0.869	0.934	0.769	0.880
	(0.045)	(0.043)	(0.045)	(0.045)	(0.040)	(0.040)	(0.054)	(0.102)	(0.063)	(0.069)	(0.050)	(0.059)	(0.040)	(0.046)	(0.046)	(0.034)	(0.050)	(0.050)
$\text{FSE}_{\mathbf{W}}$	0.765	0.772	0.715	0.740	0.758	0.691	0.772	0.716	0.733	0.800	0.769	0.749	0.786	0.804	0.778	0.655	0.776	0.730
	(0.064)	(0.056)	(0.062)	(0.060)	(0.054)	(0.060)	(0.059)	(0.058)	(0.062)	(0.048)	(0.053)	(0.068)	(0.046)	(0.040)	(0.047)	(0.109)	(0.056)	(0.048)
$\text{MAD}_{\mathbf{W}}$	0.016	0.018	0.017	0.012	0.014	0.024	0.012	0.013	0.011	0.007	0.010	0.014	0.011	0.013	0.012	0.005	0.009	0.015
	(0.002)	(0.003)	(0.004)	(0.003)	(0.002)	(0.007)	(0.003)	(0.006)	(0.004)	(0.003)	(0.002)	(0.005)	(0.002)	(0.002)	(0.002)	(0.001)	(0.002)	(0.003)
Break statistics																		
Correct Breaks Ratio	0.600	0.911	0.887	0.667	0.927	0.096	0.596	0.889	0.864	0.542	0.960	0.187	0.622	0.956	0.976	0.462	0.962	0.019
	(0.490)	(0.285)	(0.317)	(0.472)	(0.261)	(0.294)	(0.491)	(0.315)	(0.343)	(0.499)	(0.196)	(0.390)	(0.485)	(0.206)	(0.155)	(0.499)	(0.191)	(0.135)
HD_{unc}	8.000	1.300	2.900	4.500	0.600	15.800	8.500	1.300	3.200	7.500	0.400	18.600	7.700	0.700	0.900	9.900	0.300	52.300
	(12.300)	(3.700)	(6.300)	(10.100)	(2.200)	(11.200)	(12.900)	(3.100)	(7.000)	(12.800)	(2.200)	(21.100)	(12.300)	(2.800)	(3.500)	(14.500)	(1.700)	(35.100)
HD_{con}	0.200	0.300	0.800	0.100	0.200	0.900	0.400	0.400	0.700	0.200	0.100	0.800	0.300	0.200	0.400	0.300	0.100	0.900
	(0.500)	(0.600)	(0.800)	(0.300)	(0.400)	(1.000)	(0.800)	(0.700)	(0.800)	(0.400)	(0.200)	(0.500)	(0.700)	(0.500)	(0.600)	(0.600)	(0.200)	(0.300)
$\text{MAD}_{\boldsymbol{\lambda}}$	0.059	0.067	0.080	0.039	0.042	0.218	0.024	0.033	0.043	0.020	0.017	0.132	0.026	0.029	0.033	0.019	0.017	0.187
	(0.026)	(0.027)	(0.059)	(0.022)	(0.021)	(0.089)	(0.017)	(0.028)	(0.040)	(0.022)	(0.013)	(0.075)	(0.016)	(0.017)	(0.022)	(0.016)	(0.011)	(0.043)
Remaining statistics																		
$\text{MAD}_{\boldsymbol{\alpha}}$	0.235	0.216	0.312	0.187	0.159	1.094	0.377	0.256	0.269	0.298	0.391	0.403	0.301	0.250	0.352	0.251	0.364	0.333
	(0.053)	(0.055)	(0.255)	(0.077)	(0.049)	(0.522)	(0.187)	(0.285)	(0.151)	(0.170)	(0.135)	(0.285)	(0.138)	(0.196)	(0.129)	(0.186)	(0.134)	(0.114)
$\text{MAD}_{\boldsymbol{\beta}}$	0.079	0.071	0.072	0.072	0.068	0.069	0.100	0.080	0.085	0.076	0.102	0.095	0.076	0.069	0.090	0.076	0.089	0.064
	(0.031)	(0.031)	(0.029)	(0.027)	(0.023)	(0.026)	(0.047)	(0.044)	(0.036)	(0.041)	(0.051)	(0.082)	(0.030)	(0.027)	(0.035)	(0.025)	(0.032)	(0.037)
MAD_{σ^2}	0.093	0.095	0.099	0.045	0.045	0.443	0.532	0.534	0.606	0.538	0.552	1.288	1.006	1.008	1.064	1.033	1.033	2.193
	(0.055)	(0.058)	(0.078)	(0.036)	(0.032)	(0.176)	(0.170)	(0.269)	(0.235)	(0.129)	(0.061)	(0.463)	(0.055)	(0.117)	(0.163)	(0.130)	(0.072)	(0.303)
RMSE_y	1.383	1.382	1.404	1.405	1.405	1.559	1.017	1.025	1.054	1.022	1.027	1.350	1.005	1.013	1.033	1.026	1.019	1.519
	(0.024)	(0.025)	(0.043)	(0.018)	(0.017)	(0.058)	(0.064)	(0.181)	(0.114)	(0.081)	(0.029)	(0.193)	(0.035)	(0.107)	(0.074)	(0.115)	(0.043)	(0.111)

Notes: Simulation study results for robustness checks. The unconditional and conditional HD statistics are in percentages of time dimension T. 500 simulation runs were performed. I use $\mathbf{W}_1$ and set $N = 20$, $T = \{100, 200\}$, $k = 2$, $m = \{1, 2, \lfloor \frac{T}{15} \rfloor\}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\beta} = \mathbf{2}$, $\sigma^2 = 1$, $\lambda_0 = 0.30$, $\delta = 0.40$ and $\rho = 0$. In the first six columns I imposed the change $\sigma^2 = 2$. In the last twelve columns I imposed the change $\rho = \{0.50, 1\}$. Standard errors are in parentheses.

Table B6: Summary statistics of the percentage change of housing prices in the different housing corporation regions in The Netherlands

	Min	1^{st} Qu.	Mean	Median	3^{rd} Qu.	Max	St. Dev.	Ex. Kurt.	Skew.
1. Groningen Drenthe	-7.678	-0.868	1.184	0.857	3.025	9.827	2.935	3.501	0.186
2. Fryslân	-7.797	-0.569	1.355	1.225	3.222	10.029	3.334	3.365	0.053
3. Zwolle-Stedendriehoek	-5.508	-1.345	1.309	1.093	3.439	10.853	3.355	2.818	0.357
4. Oost-Nederland	-8.827	-0.509	1.153	1.034	2.944	8.265	2.895	4.348	-0.340
5. Food Valley	-9.003	-0.627	1.156	0.945	3.384	8.679	3.049	3.661	-0.349
6. Arnhem Nijmegen	-6.652	-0.597	1.255	1.069	3.191	7.692	3.001	2.858	-0.055
7. Woongaard	-7.526	-1.297	1.093	0.830	3.715	11.218	3.667	3.005	0.146
8. U16	-10.914	-0.240	1.325	1.365	3.000	12.897	3.155	5.950	-0.146
9. Noord-Holland Noord	-12.630	-0.443	1.295	0.902	3.007	9.756	3.092	6.343	-0.527
10. Metropolitan Amsterdam	-10.011	-0.057	1.504	1.583	3.809	9.035	3.404	4.185	-0.663
11. Holland Rijnland	-7.220	-0.543	1.241	1.266	3.892	10.398	3.463	2.926	-0.162
12. Haaglanden Rotterdam	-7.695	-0.387	1.364	1.358	3.339	8.434	2.876	3.467	-0.204
13. Drechtsteden Hoekse Waard	-6.763	-1.048	1.305	1.569	3.506	10.652	3.308	2.970	-0.051
14. Zeeland	-7.739	-0.850	1.329	1.032	3.655	8.418	3.575	2.767	-0.136
15. West-Brabant	-6.431	-0.133	1.182	1.230	2.727	6.134	2.463	4.074	-0.596
16. Noordoost Brabant	-6.519	-0.298	1.292	1.211	3.136	9.288	3.062	3.441	-0.059
17. Metropolitan Eindhoven	-8.792	-0.573	1.307	1.389	3.679	9.309	3.164	3.824	-0.386
18. Limburg	-8.982	-0.426	1.056	1.151	2.934	6.748	2.679	4.002	-0.503
19. Amersfoort Zeewolde	-5.978	-1.293	1.224	1.428	3.019	11.442	3.335	3.333	0.368

Notes: $N = 19$ regions and data spans the period from 1995Q3 to 2019Q4 ($T = 98$). 1^{st} Qu. and 3^{rd} Qu. refer to the first and third quartile of the data, respectively. Other abbreviated statistics refer to standard deviation (St. Dev.), excess of kurtosis (Ex. Kurt.) and skewness (Skew.) of the data.