Penalisation Methods in Fitting HighDimensional Cointegrated Vector Autoregressive Models
A Review
Levakova, Marie; Ditlevsen, Susanne

Published in:
International Statistical Review

DOI:
10.1111/insr.12553

Publication date:
2023

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
Penalisation Methods in Fitting High-Dimensional Cointegrated Vector Autoregressive Models: A Review

Marie Levakova and Susanne Ditlevsen

Department of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, Copenhagen Ø, Denmark
Email: male@math.ku.dk and susanne@math.ku.dk

Summary

Cointegration has shown useful for modeling non-stationary data with long-run equilibrium relationships among variables, with applications in many fields such as econometrics, climate research and biology. However, the analyses of vector autoregressive models are becoming more difficult as data sets of higher dimensions are becoming available, in particular because the number of parameters is quadratic in the number of variables. This leads to lack of statistical robustness, and regularisation methods are paramount for obtaining valid estimates. In the last decade, many papers have appeared suggesting different penalisation approaches to the inference problem. Here, we make a comprehensive review of different penalisation methods adapted to the specific structure of vector cointegrated models suggested in the literature, with relevant references to software packages. The methods are evaluated and compared according to a range of error measures in a simulation study, considering combinations of low and high dimension of the system and small and large sample sizes.

Key words: cointegration; decomposition of cointegration matrix; lasso; rank selection; regularisation; ridge regression; sparsity; vector autoregressive models.

1 Introduction

High-dimensional data sets containing large collections of interrelated time series are becoming more and more common as measurement and recording techniques and ways of data collection advance. A popular statistical model for multivariate time series is the vector autoregressive (VAR) process. Statistical inference of stationary VAR processes has traditionally been based on standard linear regression models (Lütkepohl, 2005). Many applications have employed nonstationary cointegrated VAR models, which means that some of the time series move together in the long run although they have stochastic trends. Then the standard linear regression models may lead to spurious regression (Granger & Newbold, 1974) and cointegration analysis...
is the right method of statistical inference. Therefore, cointegration analysis is a popular method in econometrics and is also used in other fields, such as in biology (Dahlhaus et al., 2018; Østergaard et al., 2017), agriculture (Zapata & Gil, 1999) or climatology (Schmith et al., 2012; Friedrich et al., 2023). Recently, attempts to fit cointegrated VAR models in high-dimensional settings started appearing, such as for big data in macroeconomics and finance (Wilms & Croux, 2016; Smeekes & Wijler, 2020; Chen & Schienle, 2022), EEG data (Levakova et al., 2022), phase synchronisation (Østergaard et al., 2022) and in structural health monitoring problems (Cross et al., 2011; Mousavi & Gandomi, 2021).

While upscaling a linear regression model for a high-dimensional stationary VAR process has already its challenges, high dimensions of cointegrated VAR processes induce specific issues for model fitting. The most challenging aspects are the determination of the cointegration rank and robust estimation of the matrix of coefficients with the specified rank. The asymptotic distributions of cointegration tests when the dimension and the sample size grow proportionally has been studied recently in (Onatski & Wang, 2018, 2019). If the rank is misspecified, the variance is inflated for overestimated ranks, whereas a bias is introduced for underestimated rank, but often with a lower variance (Holberg & Ditlevsen, 2022). Even for settings with a fixed number of dimensions mildly exceeding two, existing techniques for fitting cointegrated models often fail to provide accurate, testable and computationally tractable estimates (Ho & Sorensen, 1996). A modification of the popular Johansen procedure has recently been proposed in Bykhovskaya & Gorin (2022), and alternative approaches such as factor models have been used (Smeekes & Wijler, 2020).

Regularisation methods in estimation are a popular way to deal with high-dimensional problems, where the number of estimated parameters is close to or even higher than the number of observations and standard estimators suffer from extreme variance or are not defined. Regularisation methods typically place a penalty on some of the estimated parameters, which are estimated with a tolerable bias and lower variance. The appropriate use of penalisation methods when some of the variables are nonstationary has been attracting increasing attention, especially in a single-equation framework (Koo et al., 2020; Lee et al., 2022; Masini & Medeiros, 2021; Smeekes & Wijler, 2021).

The goal of this article is to describe statistical inference and penalisation methods for the analysis of cointegrated multidimensional data, and to provide tools for choosing an adequate method for a given problem. The choice of method depends on the purpose of the analysis. If the purpose is to infer an interaction graph, it is important to identify zero components in the cointegration matrix; if the purpose is prediction, minimisation of squared errors might be appropriate; or if correct identification of cointegration relations are of interest, yet other methods might be the relevant ones to implement. Furthermore, some methods might outperform other methods in high-dimensional settings where the number of observations is small compared with number of parameters, even if they are not competitive in low dimensions. In this review, we consider all of the above mentioned objectives. A thorough discussion of how to deal with high-dimensional cointegrated data through factor models with the specific focus on forecasting can be found in Smeekes & Wijler (2020).

We give an overview of possible penalisations used in fitting high-dimensional cointegrated models. For example, we present a direct penalisation of the coefficients describing interactions between univariate processes, ways to regularise cointegration relationships or adjusting mechanisms and how to shrink the cointegration rank. The main focus of the paper is on the implementation of the methods. In the final section, we present a simulation study. We compare and evaluate all methods with a list of different error measures, and for different scenarios of sample size and system dimension, which highlights advantages and disadvantages of each method.
Penalisation in Fitting Cointegrated VAR Models

The paper is structured as follows: First, we provide a brief review of cointegrated vector autoregressive models and their properties in Section 2. In Section 3, we briefly summarise penalisation methods, namely the lasso, ridge regression, elastic net and nuclear norm penalty. In Section 4, we provide a review of the many algorithms with different types of penalisation proposed in the literature to deal with high-dimensional cointegration models, starting with the basis of un-penalised methods, namely ordinary least squares (OLS) and the Johansen procedure. We also include the two main tests, the trace test and the maximum eigenvalue test, to determine the rank of the system, as well as the rank selection criterion and the method of Pedroni et al. (2015), which are more appropriate for large dimensions. In Section 5, we conduct a simulation study, evaluating the methods through different error measures. Finally, in Section 6, we discuss the results and provide some recommendations.

1.1 Notation

A $p$-dimensional column vector of zeros is denoted by $0_p$, a $p$-dimensional column vector of ones is denoted by $1_p$, and $1_p$ is the $p \times p$ unit matrix. The trace of a generic matrix $A$ is $\text{tr}(A)$, the transpose of $A$ is $A'$, the generalised inverse of $A$ is $A^{-}$. If $A \in \mathbb{R}^{p \times r}$ with $r \leq p$ is a full rank matrix, a matrix $A_{\perp} \in \mathbb{R}^{p \times (p-r)}$ denotes the orthogonal complement, that is, $A'A_{\perp} = 0$. The symbol $A_{i,.}$ represents the $i$-th row of the matrix $A$, $A_{.j}$ is the $i$-th column of $A$, $A_{(-i)}$ is the matrix $A$ with $i$-th row omitted and $A_{(-i)}$ is the matrix $A$ with $i$-th column omitted. The $l_2$-norm of a vector $v \in \mathbb{R}^p$ is $\|v\|_2 = \sqrt{\sum_{i=1}^p v_i^2}$, the $l_1$-norm is $\|v\|_1 = \sum_{i=1}^p |v_i|$, and the inner product of two vectors $u, v \in \mathbb{R}^p$ is $\langle u, v \rangle = \sum_{i=1}^p u_i v_i$. The Frobenius inner product of matrices $A, B \in \mathbb{R}^{p \times q}$ is defined as $\langle A, B \rangle_F = \text{tr}(A'B)$ and the Frobenius norm of a matrix $A$ is $\|A\|_F = \sqrt{\text{tr}(A'A)} = \sqrt{\sum_{i,j} A_{ij}^2}$. The $l_1$ norm of a matrix $A$ is $\|A\|_1 = \sum_{i,j} |A_{ij}|$. A diagonal matrix with the elements of a vector $v$ on the diagonal is denoted by $\text{diag}(v)$. Similarly, we sometimes write $\text{diag}(A)$ where the argument $A$ is a matrix, to denote a diagonal matrix with the same elements on the diagonal as in $A$. Finally, a hat marks an estimate, that is, $\hat{\mu}$ is an estimate of the parameter $\mu$.

2 Cointegration

Assume a sample of a $p$-dimensional stochastic process $\{x_n\}_{n=0}^N$ observed at equidistant time points $t_0, \ldots, t_N$. An observation made at time $t_n$ is the vector $x_n = (x_{1n}, \ldots, x_{pn})' \in \mathbb{R}^p$. The series $\{x_n\}_{n=0}^N$ is said to be a realisation of a vector autoregressive (VAR) process of order $k$, if

$$x_n = \mu + A_1 x_{n-1} + \ldots + A_k x_{n-k} + \varepsilon_n, \quad n = k + 1, \ldots, N,$$

(1)

where $A_1, \ldots, A_k \in \mathbb{R}^{p \times p}$, $\mu \in \mathbb{R}^p$ and $\{\varepsilon_n\}_{n=0}^N$ are $p$-dimensional i.i.d. errors with zero mean and constant variance $\text{Var}(\varepsilon_n) = \Sigma$, where $\Sigma \in \mathbb{R}^{p \times p}$ is a positive semidefinite matrix. For the sake of the methods relying on the likelihood approach, we further assume $\varepsilon_n \sim N_p(0_p, \Sigma)$.

However, some of the estimation methods presented in this review do not require the normality assumption. For convenience, we use the vector error-correction (VEC) form of the model,

$$\Delta x_n = \mu + \Pi x_{n-1} + \sum_{i=1}^{k-1} \Gamma_i \Delta x_{n-i} + \varepsilon_n,$$

(2)
where $\Delta x_n = x_n - x_{n-1}$, $\Pi = -(I_p - A_1 - \ldots - A_k)$ and $\Gamma_i = -(A_{i+1} + \ldots + A_k)$, for $i = 1, \ldots, k - 1$.

Cointegration (Engle & Granger, 1987; Johansen, 1995) can only arise when $\{x_n\}^N_{n=0}$ is non-stationary with a stochastic trend, that is, a realisation of an integrated process. A process, whose first differences $\Delta x_n$ are asymptotically stationary (i.e. the first and the second moment tend to a constant), is termed an integrated process of order 1 or an $I(1)$ process. Similarly, a process, whose first differences $\Delta x_n$ are not asymptotically stationary, but second differences $\Delta^2 x_n = \Delta x_n - \Delta x_{n-1}$ are, is an $I(2)$ process, and so on. The notion of an integrated process can also be established through the characteristic polynomial $|(1 - z)(I_p - \Gamma_1 z - \ldots - \Gamma_k z^{k-1}) - \Pi z| = 0$, where $z \in \mathbb{C}$. The process $\{x_n\}^N_{n=0}$ is integrated if the characteristic polynomial has at least one root $z = 1$, while all the other roots lie outside of the unit circle. A process can be cointegrated if it is integrated of order 1 or more, but we will focus only on $I(1)$ cointegrated processes in this review, because it is the most common scenario in applications.

The rank of $\Pi$ is termed cointegration rank, $r = \text{rank}(\Pi)$, because it has fundamental implications for the properties of the VAR process. We can distinguish 3 qualitatively different forms of the process $\{x_n\}^N_{n=0}$, depending on the value of the cointegration rank:

$r = 0$: $\{x_n\}^N_{n=0}$ is driven by $p$ independent stochastic trends.

$r = p$: $\{x_n\}^N_{n=0}$ is asymptotically stationary and contains neither a stochastic trend, nor a linear trend (Lütkepohl, 2005).

$0 < r < p$: $\{x_n\}^N_{n=0}$ is driven by exactly $p - r$ independent stochastic trends. Moreover, there exist $r$ linear combinations of the vector process $\{x_n\}^N_{n=0}$, termed cointegration relationships, that yield an asymptotically stationary one-dimensional process. Linear trends due to the drift $\mu$ are possible too.

The third case is investigated in cointegration analysis and this is also the case we will consider from now on. Because the matrix $\Pi$ describes the long-term behavior of the process and represents the main challenge in fitting the model because of the restriction on its rank, while the matrices $\Gamma_1, \ldots, \Gamma_{k-1}$ describing the short-term effects are of arbitrary rank, we assume for simplicity only VAR processes of order 1, that is, $\Delta x_n = \mu + \Pi x_{n-1} + \varepsilon_n$, $n = 2, \ldots, N$.

If $0 < r < p$, we can find matrices $\alpha, \beta \in \mathbb{R}^p \times r$ of full rank $r$ such that $\Pi = a\beta'$. The matrices $\alpha$ and $\beta$ are not unique, because we can take an arbitrary invertible matrix $Q \in \mathbb{R}^r \times r$ and find another decomposition $\Pi = (aQ)(Q^{-1}\beta') = a^*\beta'\varepsilon'$. However, the subspaces spanned by the columns of $\alpha$ and $\beta$, $\text{sp}(\alpha)$ and $\text{sp}(\beta)$, are unique (Kessler & Rahbek, 2004). The matrices $\alpha$ and $\beta$ play a special role in the interpretation of the VEC model.

The matrix $\beta$ is called the cointegration matrix, because the columns of $\beta$ contain cointegration vectors. A cointegration vector $v \in \mathbb{R}^p$ is a vector such that $v'x_n$ is an asymptotically stationary univariate process. Cointegration vectors form a linear subspace, which means that any linear combination of 2 distinct cointegration vectors is also a cointegration vector. The columns of $\beta$ represent one possible basis of the cointegration subspace.

The matrix $\alpha$ is called the loading matrix. It describes the correcting mechanism which ensures that $\beta'x_n$ is always pushed to the long-term mean $\mathbb{E}(\beta'x) = \lim_{n \to \infty} \mathbb{E}(\beta'x_n)$. This can be seen easily by re-writing the VEC model as...


\[
\Delta x_n = \mu + a \mathbb{E}(\beta' x_{n-1}) + \alpha \left[ \beta' x_{n-1} - \mathbb{E}(\beta' x_{n-1}) \right] + \epsilon_n. \tag{3}
\]

The matrix $\alpha$ thus reveals the rate at which the system reacts to the deviations of the cointegration relationships $\beta' x_{n-1}$ from the asymptotic mean $\mathbb{E}(\beta' x_{n-1})$. The natural interpretation of $\Pi$ is that the elements $\Pi_{ij}$ describe interactions between the univariate processes $\{x_{in}\}_{n=0}^N$ and $\{x_{jn}\}_{n=0}^N$, $i, j \in \{1, \ldots, p\}$. In a more concise matrix form, we have

\[
\Delta X = 1_N \mu' + \Pi \Pi' + \epsilon, \tag{4}
\]

where $\Delta X = (\Delta x_1, \ldots, \Delta x_N)'$, $X = (x_0, \ldots, x_{N-1})'$ and $\epsilon = (\epsilon_1, \ldots, \epsilon_N)'$.

To rule out the possibility of an $I(1)$ cointegrated process of order $k \geq 2$, an additional condition $|\alpha'_i \beta_j| \neq 0$ must hold according to the Granger representation theorem (Engle & Granger, 1987). The conditions to secure that $\{x_n\}_{n=0}^N$ is $I(1)$ and cointegrated can be formulated using the Jordan normal form $\Pi = PJP^{-1}$, where $J$ is the associated block-diagonal Jordan matrix $J = \text{diag}(J_{p_1}(\lambda_1), \ldots, J_{p_m}(\lambda_m))$. The Jordan blocks $J_{p_i}(\lambda_i)$ are of dimension $p_i$, such that $\sum_{i=1}^m p_i = p$, with $\lambda_i$ on the diagonal. The $J_{p_i}(0)$ blocks give rise to stochastic trends. The matrix $P$ is an orthogonal $p \times p$ matrix. The condition $|\alpha'_i \beta_j| \neq 0$ holds if and only if there is no $J_{p_i}(0)$ block with $p_i \geq 2$ (Archontakis, 1999). Note that $J$ is not diagonal and $\Pi$ is not diagonalisable in general, if blocks $J_{p_i}(\lambda_i)$ with $p_i \geq 2$ have $|\lambda_i + 1| < 1$, which then have unit elements above the diagonal.

For estimation purposes, other factorisations than $\Pi = \alpha' \beta$ and $\Pi = PJP^{-1}$ are sometimes useful. One of them is the singular value decomposition (SVD), $\Pi = UDV'$, where $U = (U_1, \ldots, U_r)$, $V = (V_1, \ldots, V_r) \in \mathbb{R}^{p \times r}$ are orthonormal matrices of left and right singular vectors, respectively, and $D = \text{diag}(d_1, \ldots, d_r)$ is a diagonal matrix of singular values. One of the uses of the SVD is to find a possible decomposition of $\Pi$ into $\alpha$ and $\beta$. We can either choose $\alpha = U$, $\beta = VD$, or $\alpha = U \beta = V$. Another useful factorisation is the QR decomposition $\Pi' = QR$, where $Q \in \mathbb{R}^{p \times p}$ is an orthogonal matrix and $R \in \mathbb{R}^{p \times p}$ is upper triangular.

3 High Dimension and Penalisation

When estimating parameters of a cointegrated VAR process, a ‘gold standard’ is to apply the Johansen procedure (Johansen, 1995), which is a special case of maximum likelihood estimation for the VEC model (2). However, this estimation method may not be optimal when: (1) the dimension $p$ is high, (2) the number of observations $N$ is low compared with $p$, (3) a substantial portion of elements of $\Pi$ are zeros and we wish to identify them. Then it can be beneficial to impose some restrictions on $\Pi$ to encourage a simpler form.

The VEC model already imposes one restriction on $\Pi$, namely rank($\Pi$) = $r$. In reality, the rank is unknown and must be estimated as well. In the Johansen procedure, this is carried out by sequential likelihood-ratio testing until the cointegration rank is identified. A particular problem with a high dimension $p$ is that the estimation of the rank is technically difficult and tends to suffer from substantial bias (Ostergaard et al., 2022). Estimates of the cointegration relationships are biased for underestimated rank, whereas variance is increased for overestimated rank (Holberg & Ditlevsen, 2022). Therefore, it is worth exploring alternative strategies to identify $r$.

This paper presents a range of estimation approaches that impose some form of restriction on $\Pi$, $r$ or both. Before presenting the methods one by one, we review some basic forms of penalisation in uni- and multivariate regression problems that will appear repeatedly throughout

International Statistical Review (2023)
the paper (see Hastie et al., 2019, for further details). First, consider a univariate regression problem of the form

\[ y = Zb + \varepsilon, \]  

(5)

where \( y \in \mathbb{R}^N \) is the response vector, \( Z \in \mathbb{R}^{N \times q} \) is the design matrix, \( b \in \mathbb{R}^q \) is the vector of unknown parameters and \( \varepsilon \in \mathbb{R}^N \) is a vector of i.i.d. errors.

3.1 Lasso

The lasso estimator of \( b \) is defined as

\[ \hat{b} = \arg \min_{b \in \mathbb{R}^q} \left\{ \frac{1}{2} \| y - Zb \|^2_2 + \lambda \| b \|^1_1 \right\}, \]  

(6)

where \( \lambda \geq 0 \) is a tuning parameter. A well-known property of the lasso estimator is that the \( l_1 \) penalty \( \lambda \| b \|^1_1 \) tends to shrink some of the elements of \( \hat{b} \) to exact zeros and thus performs variable selection. Specifically, \( \hat{b} = 0 \), if \( \lambda \geq \lambda_{\text{max}} = \max_j \langle y, Z \cdot j \rangle \) (Hastie et al., 2019). There exist several algorithms to solve (6) numerically. Probably the most convenient one is the cyclic coordinate descent, which relies on the soft-threshold operator, where \( \lambda \) is a positive constant, defined by

\[ S_\lambda(A) = (\text{sign}(A_{ij})|A_{ij}| - \lambda)_+ / |A_{ij}|, \quad i = 1, \ldots, q, \quad j = 1, \ldots, r. \]  

(7)

The cyclic coordinate descent cycles repeatedly over the predictors \( Z \cdot j \) and updates the respective regression coefficients as

\[ \hat{b}^{(k+1)}_j = \frac{1}{Z \cdot j^T S_\lambda \left( Z' \cdot j \left( y - \sum_{i=1, i \neq j}^{q} Z \cdot i \hat{b}^{(k)}_i \right) \right)} \cdot \frac{y - \sum_{i=1, i \neq j}^{q} Z \cdot i \hat{b}^{(k)}_i}{Z \cdot j^T}. \]  

(8)

The tuning parameter \( \lambda \) can be chosen by estimating \( \hat{b} \) with a range of values of \( \lambda \), calculating some measure of fit for each of them, for example, a crossvalidated error, Akaike information criterion (AIC, Akaike, 1974), Bayes information criterion (BIC, Schwarz, 1978) or Hannan-Quinn criterion (HQ, Hannan & Quinn, 1975), and picking the value of \( \lambda \) providing the best fit.

3.2 Ridge regression

Another popular penalty is the ridge penalty (also termed Tikhonov regularisation). The ridge estimator is defined as

\[ \hat{b} = \arg \min_{b \in \mathbb{R}^q} \left\{ \| y - Zb \|^2_2 + \lambda \| b \|^2_2 \right\}. \]  

(9)

The minimisation problem is easily solved by writing the expression within the brackets in (9) as a sum of squares \( \| y^* - Z^*b^* \|^2 \), where \( y^* = (y', 0_q)' \) and \( Z^* = (Z', \sqrt{\lambda}I_q)' \) are augmented data matrices. This yields the estimator

\[ \hat{b} = (Z'Z + \lambda I_q)^{-1}Z'y. \]  

(10)

The ridge penalty shrinks \( \hat{b} \) towards \( 0_q \) as well, but instead of shrinking a couple of coefficients to exact zeros and not affecting much the others as lasso does, the ridge penalty shrinks...
all coefficients towards zeros more evenly and does not produce exact zeros. It is often used in situations with multicollinearity, where it reduces the variance of the estimator at the price of introducing a tolerable bias.

### 3.3 Elastic net

The elastic net (Zou & Hastie, 2005) combines the lasso and the ridge penalty as follows

\[
\hat{b} = \arg \min_{b \in \mathbb{R}^q} \left\{ \frac{1}{2} \| y - Zb \|^2_2 + \lambda \left( \frac{1}{2} (1 - \omega) \| b \|^2_2 + \omega \| b \|_1 \right) \right\}.
\]

The aim is to get the advantages of both lasso and ridge regression, that is, to achieve more similar shrinkage of the different elements of \( \hat{b} \), while allowing some of them to be exact zeros. Clearly, lasso and ridge regression are special cases of the elastic net, for \( \omega = 1 \) and \( \omega = 0 \), respectively. The estimate \( \hat{b} \) can be calculated by the cyclic coordinate descent algorithm with the update step

\[
\hat{b}_j^{(k+1)} = \frac{S_{\lambda\omega} \left( Z\cdot j^T \left( y - \sum_{i=1, i \neq j}^q Z\cdot i \hat{b}_i^{(k)} \right) \right)}{Z\cdot j^2 + \lambda (1 - \omega)}.
\]

Now, consider a multivariate regression

\[
Y = ZB + \varepsilon,
\]

where \( Y \in \mathbb{R}^{N \times p} \) is the response matrix, \( Z \in \mathbb{R}^{N \times q} \) is the design matrix, \( B \in \mathbb{R}^{q \times p} \) is the matrix of unknown parameters and \( \varepsilon \in \mathbb{R}^{N \times p} \) is a vector of i.i.d. errors. The multivariate elastic net estimator of \( B \),

\[
\hat{B} = \arg \min_{B \in \mathbb{R}^{q \times p}} \left\{ \frac{1}{2} \| Y - ZB \|^2_F + \lambda \left( \frac{1}{2} (1 - \omega) \| B \|^2_F + \omega \| B \|_1 \right) \right\},
\]

amounts to \( p \) independent univariate elastic net problems with the response vector \( Y\cdot j \) and vector of coefficients \( B\cdot j, j \in \{1, \ldots, p\} \). The general regression in (13) relates to VECM regression (4) by setting \( Y = \Delta X, Z = X \) and \( B = \Pi' \). Note, however, that \( \Pi \) has a restricted rank, which is not addressed in the elastic net.

### 3.4 Penalty to enforce row sparsity

The multivariate elastic net produces estimates with zero elements scattered anywhere in \( \hat{B} \). Sometimes it is desirable to have entire rows of \( \hat{B} \) with either zero, or nonzero elements. This can be achieved by solving the problem

\[
\hat{B} = \arg \min_{B \in \mathbb{R}^{q \times p}} \left\{ \frac{1}{2} \| Y - ZB \|^2_F + \lambda \sum_{j=1}^p \| B\cdot j \|_2 \right\}.
\]
\[
\hat{B}_j^{(k+1)} = \left( Z'_j Z_j + \frac{\lambda}{\|B_j^{(k)}\|_2} \right)^{-1} Z'_j (Y - Z_{-j} B_{-j}^{(k)}).
\]

### 3.5 Nuclear norm penalty

Matrix reconstruction problems gave origin to the use of a penalisation proportional to the nuclear norm (also called trace norm, Schatten norm or Ky Fan norm) \(\|B\|_* = \sum_{i=1}^{\min\{p, q\}} d_i\), where \(d_i\) are singular values of \(B\). The estimator with a penalised nuclear norm is defined as

\[
\hat{B} = \arg \min_{B \in \mathbb{R}^{q \times p}} \{ \|Y - ZB\|_F^2 + \lambda \|B\|_* \}. \tag{17}
\]

The nuclear norm is a convex relaxation of the rank of a matrix; therefore, a shrinkage of the nuclear norm frequently leads to reduction of the rank of \(B\). The estimator with the nuclear penalty is more complex to calculate. The solution can be found, for example, by a second-order cone program solver such as SDPT3 (Tütüncü et al., 2003). An alternative simpler algorithm is given in Section 4.4.1.

Apart from these commonly used penalties, some of the estimation methods presented in the following section use more specific penalties that will be explained together with the respective estimation methods.

### 4 Methods

In this section, we give an overview of ways of estimating parameters of the VEC model (4), with the main focus on penalisation methods. For simplicity, we consider model (4) without a drift term \(\mu\) in the form,

\[
Y = Z\Pi' + \epsilon, \tag{18}
\]

where \(Y, Z\) are either centered or standardised counterparts of \(\Delta X\) and \(X\), respectively. Centering yields \(Y = Y^{(c)} = \Delta X - \frac{1}{N} 1_N 1_N' \Delta X\) and \(Z = Z^{(c)} = X - \frac{1}{N} 1_N 1_N' X\). Standardising yields \(Y = Y^{(s)} = Y^{(c)} \text{diag}(\sigma_{\Delta X}, ..., \sigma_{\Delta X})^{-1}\) and \(Z = Z^{(s)} = Z^{(c)} \text{diag}(\sigma_X, ..., \sigma_X)^{-1}\), where \(\sigma_w\) denotes empirical standard deviation of a vector of observations \(w\). The estimates \(\hat{\Pi}\) and \(\hat{\mu}\) of the parameters in the original model (4) are related to the estimates \(\hat{\Pi}^{(c)}\) and \(\hat{\Pi}^{(s)}\) obtained with the centered and standardised data matrices, respectively, as

\[
\hat{\Pi} = \hat{\Pi}^{(c)} = \text{diag}(\sigma_{\Delta X}, ..., \sigma_{\Delta X}) \hat{\Pi}^{(s)} \text{diag}(\sigma_X, ..., \sigma_X)^{-1}, \quad \hat{\mu} = \Delta X - \frac{1}{N} 1_N 1_N' X\hat{\Pi}'. \tag{19}
\]

The methods can be grouped according to the type of factorisation of \(\Pi\) utilised:

1. **No factorisation**: The matrix \(\Pi\) is estimated directly. The penalisation can be imposed on the elements of \(\Pi\) or on rank(\(\Pi\)).
2. **Cointegration factorisation** \(\Pi = \alpha \beta'\): The penalisation is imposed either on \(\alpha\) or \(\beta\).
3. **QR factorisation** \(\Pi' = QR\): The penalisation is imposed on \(R\).
4. **SVD factorisation** \(\Pi = U D V'\): The penalisation can be imposed either on the singular values, or on both the singular values and the singular vectors simultaneously.
The estimation methods also differ in their approach to the cointegration rank. Some of the methods select an appropriate rank as part of their estimation procedure. However, most of the methods do not address the problem of the rank determination and require some preliminary estimate of $r$ to set the dimensions of the matrices under estimation. The rank of the final estimate $\hat{\Pi}$ might get further reduced below the initial value. We present two methods of rank selection in Section 4.5.

All the presented methods are summarised together with their main advantages and disadvantages in Table 1, where relevant references and software implementation are also listed. Details on each method are presented in the following. They are evaluated through a simulation study in Section 5.

4.1 Direct Estimation of $\Pi$

We start with estimation methods where no special structure of $\Pi$ is utilised. Many estimators can be expressed in terms of the moment matrices

\[
S_{yy} = Y' Y / N, \quad S_{yz} = Y' Z / N, \quad S_{zy} = Z' Y / N, \quad S_{zz} = Z' Z / N.
\]

(20)

The ordinary least squares (OLS) estimator of $\Pi$ minimises the sum of squared residuals with no restrictions imposed on the rank of $\Pi$. It is given by

\[
\hat{\Pi}_{OLS} = \arg \min_{\Pi \in \mathbb{R}^{p \times p}} \| Y - Z \Pi \|^2_F = S_{yz} S_{zz}^{-1}.
\]

(21)

In practice, the rows of $\hat{\Pi}_{OLS}$ can be obtained from OLS estimation in $p$ univariate regression models $Y_{i} = Z \Pi_{i} + \varepsilon_{i}, i = 1, \ldots, p$. This fact reveals the main drawback of the method, namely that possible relations between the response variables $Y_{i}, i = 1, \ldots, p$, are ignored. Moreover, the OLS estimator performs poorly when predictors are highly correlated, which can easily happen with a cointegrated process. On the other hand, the OLS estimator is consistent when $p$ is fixed (Lütkepohl, 2005).

4.1.1 Penalising $\Pi$ with unrestricted rank

The simplest approach to obtain a sparse estimator of $\Pi$ is to impose a lasso penalty, Equation (6), and thus force some elements of $\Pi$ to be exact zeros. A ridge penalty, Equation (9), can be added to make the estimator robust in case of collinearity. Thus, an estimator with an elastic net penalty, Equation (14), is given by

\[
\hat{\Pi} = \arg \min_{\Pi \in \mathbb{R}^{p \times p}} \left\{ \frac{1}{2}\| Y - Z \Pi \|^2_F + \lambda \left[ \omega \| \Pi \|_1 + \frac{1 - \omega}{2} \| \Pi \|_2^2 \right] \right\}.
\]

(22)

In practice, the multivariate elastic net regression (22) is performed as $p$ independent univariate elastic net regressions

\[
\min_{\Pi_i \in \mathbb{R}^{p}} \left\{ \frac{1}{2}\| Y_{i} - Z \Pi_{i} \|^2_F + \lambda \left[ \omega \| \Pi_{i} \|_1 + \frac{1 - \omega}{2} \| \Pi_{i} \|_2^2 \right] \right\}, \quad i = 1, \ldots, p.
\]

(23)

Several forms of lasso-type penalties in single equation models with cointegrated variables have been further studied to illustrate how nonstationarity of the predictors affects the estimator. The asymptotic properties of the standard lasso are provided in Koo et al. (2020). In Smeekes & Wijler (2021), a single-equation penalised error correction selector is proposed. This procedure...
Table 1. Overview of the estimation methods. Some of the comments in the last column of pros and cons refer to results from the simulation study conducted in Section 5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Penalty type</th>
<th>References (implementation in R)</th>
<th>Effects</th>
<th>Pros/Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>None</td>
<td>Any textbook (Function lm)</td>
<td>$\Pi$ has full rank. $\Pi$ is not sparse.</td>
<td>+ Easy to compute.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ Unbiased.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Risk of wrong estimates by ignoring cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Risk of overfitting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Fails when $N &lt; p$.</td>
</tr>
<tr>
<td>Elastic net</td>
<td>$\lambda \left[ \omega | \Pi |_1 + \frac{1 - \omega \omega}{2} | \Pi |_2^2 \right]$</td>
<td>(Tibshirani, 1996), (Smeekes &amp; Wijler, 2018)</td>
<td>Sparse $\Pi$. $\Pi$ is usually of full rank.</td>
<td>+ No need to know the rank of $\Pi$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Package glmnet (Friedman et al., 2010))</td>
<td></td>
<td>+ Easy to implement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ Best detection of zeros for small samples.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Risk of wrong estimates by ignoring cointegration.</td>
</tr>
<tr>
<td>Elastic net + SVD</td>
<td>$\lambda \left[ \omega | \Pi |_1 + \frac{1 - \omega \omega}{2} | \Pi |_2^2 \right]$</td>
<td>(Package glmnet (Friedman et al., 2010))</td>
<td>$\Pi$ has reduced rank. Zeros obtained by the elastic net get lost after SVD.</td>
<td>+ Accounts for cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>only $r$ layers of SVD of $\Pi$ kept function svd</td>
<td></td>
<td>+ Easy to implement.</td>
</tr>
<tr>
<td>Rank penalty</td>
<td>$\lambda \omega \text{rank}(\Pi) + \frac{1 - \omega \omega}{2} | \Pi |_2^2$</td>
<td>(Bunea et al., 2011; Chen et al., 2013) (Supplement to this paper)</td>
<td>Reduces the rank of $\Pi$. $\Pi$ is not sparse.</td>
<td>+ No need to know the cointegration rank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- The algorithm works only for $N &gt; p$.</td>
</tr>
<tr>
<td>Lasso w. fixed rank</td>
<td>$\lambda | \Pi |_1$ s.t. rank($\Pi$) = $r$</td>
<td>(Lian et al., 2015) (Supplement to this paper)</td>
<td>$\Pi$ has reduced rank. $\Pi$ is not sparse.</td>
<td>+ Accounts for cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ Lowest MSPE, RD and $\Theta$ for $p = 50$, $N = p$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Issues with convergence of the algorithm.</td>
</tr>
<tr>
<td>Johansen method</td>
<td>rank($\Pi$) = $r$</td>
<td>(Johansen, 1995) (Supplement to this paper or package urca (Pfaff, 2008) if $p \leq 11$)</td>
<td>$\Pi$ has a reduced rank. $\Pi$ is not sparse.</td>
<td>+ Accounts for cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ Recovers the decomposition $\Pi = \alpha \beta'$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Large variability in small samples.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Fails when $N &lt; p$.</td>
</tr>
<tr>
<td>Lasso on $\beta$</td>
<td>$\lambda | \beta |_1$</td>
<td>(Wilms &amp; Croux, 2016) (Supplement to Wilms &amp; Croux, 2016)</td>
<td>$\Pi$ has reduced rank. $\beta$ is sparse. $\Pi$ is not sparse.</td>
<td>+ Accounts for cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ Recovers the decomposition $\Pi = \alpha \beta'$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Little control over the overall amount of shrinkage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Continues)</td>
</tr>
<tr>
<td>Method</td>
<td>Penalty type</td>
<td>References (implementation in R)</td>
<td>Effects</td>
<td>Pros/Cons</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Elastic net on $\alpha$</td>
<td>$\lambda \left[ \omega</td>
<td></td>
<td>\alpha</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Supplement to Levakova et al., 2022)</td>
<td>$\hat{\alpha}$ is sparse.</td>
<td>+ Recovers the decomposition $\Pi = \alpha \beta$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\hat{\Pi}$ is not sparse.</td>
<td>+ Lowest RD and $\Theta(\hat{\Pi}, \Pi)$ for $p = 50$, $N = 3p$ and $p = 150$, $N = 3p$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Fails when $N &lt; p$.</td>
</tr>
<tr>
<td>Lasso on QR</td>
<td>$\sum_{j=1}^p \frac{\lambda}{w_j}</td>
<td></td>
<td>R_{\cdot j}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Function qr, package glmnet (Friedman et al., 2010))</td>
<td>$\hat{\Pi}$ is not sparse.</td>
<td>+ Recovers the decomposition $\Pi = \alpha \beta$.</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>$\lambda \left[ \omega</td>
<td></td>
<td>\Pi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Supplement to this paper)</td>
<td>$\hat{\Pi}$ is not sparse.</td>
<td>+ Smoother penalisation than with the rank penalty.</td>
</tr>
<tr>
<td>Adaptive nuclear penalty</td>
<td>$\lambda \left[ \omega</td>
<td></td>
<td>Z\Pi'\Pi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Function rrr in the package rrpack (Chen, 2019))</td>
<td>$\Pi$ is not sparse.</td>
<td>- The algorithm works only for $N &gt; p$.</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>$\sum_{k=1}^p \sum_{j,k=1}^p w_{ik}[d_k U_{ik} V_{jk}]$</td>
<td>(Chen et al., 2012)</td>
<td>$\Pi$ has reduced rank.</td>
<td>+ Accounts for cointegration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Function rssvd in the package rrpack (Chen, 2019))</td>
<td>Sparse $\Pi$.</td>
<td>+ Best detection of zeros for large samples.</td>
</tr>
</tbody>
</table>
is close to elastic net penalisation (23), but the predictors $Z_\star_j$ are transformed to achieve that errors $\varepsilon_\star_j$ are orthogonal to $\varepsilon_\star_i$. In Masini & Medeiros (2021), a weighted lasso procedure is proposed, with weights determined according to the nature of the trend in each series. In Lee et al. (2022), a twin adaptive lasso is introduced. It consists of two-step weighted lasso that improve variable selection.

Note that the same tuning parameter $\lambda$ is used in (23) for all $i = 1, \ldots, p$, hence the choice of $\lambda$ must be carried out with respect to all the univariate models, for example, by choosing a value that gives the least crossvalidation error of the full multivariate model. The estimator can be further refined by introducing weights into the penalty term, which then becomes $\lambda \omega \sum_{i,j} w_{ij} |\Pi_{ij}| + (1 - \omega) \sum_{i,j} w_{ij} \Pi_{ij}^2$. A possible choice of the weights is $w_{ij} = 1/|\hat{\Pi}_{ij}^{OLS}|$. Penalised regression with the lasso-type penalty was investigated in the context of cointegrated VEC models in Smeekes & Wijler (2018), where the lasso outperformed OLS in a simulation study in terms of the forecast error.

4.1.2 Reducing the rank of $\Pi$

The OLS estimator and the estimator penalising the elements of $\Pi$ are likely to give an estimate with full or close to full rank. If we want an estimate of $\Pi$ among matrices with rank $r$. This is best achieved with the SVD decomposition of $\hat{\Pi}$, $\hat{\Pi} = UDV'$. The Eckart-Young theorem states that a matrix $\hat{\Pi}$ solving the problem $\min \{ \Pi \in \mathbb{R}^p \times p \| \Pi - \hat{\Pi} \|_F \}$ subject to $\text{rank}(\hat{\Pi}) = r$ can be found as

$$\hat{\Pi} = \sum_{i=1}^r d_i U_{\cdot i} V'_{\cdot i}, \quad (24)$$

where the singular values are in descending order, $d_1 \geq \ldots \geq d_p$, and $U_{\cdot i}, V_{\cdot i}$ are the corresponding left and right singular vectors. For a more concise notation, let $\mathcal{H}_\lambda(A) = \{ A_{ij} : |A_{ij}| \geq \lambda \}_{i,j=1}^p$ for $A \in \mathbb{R}^p \times p$ be the hard-thresholding operator, and $\mathcal{H}_\lambda^*$ the hard-thresholding operator for an SVD, $\mathcal{H}_\lambda^*(\Pi) = U \mathcal{H}_\lambda(D) V'$. Then $\hat{\Pi}_r = \mathcal{H}_\lambda^*(\hat{\Pi})$.

4.1.3 Penalising the rank of $\Pi$

The rank penalty was introduced in Bunea et al. (2011) for reduced rank regression. The estimator is defined as

$$\hat{\Pi} = \text{arg min} \{ \| Y - Z\hat{\Pi}^2_F + \lambda \text{rank}(\Pi) \}. \quad (25)$$

A simple computational procedure is derived in Chen et al. (2013). It can be shown that minimising the objective function within the brackets in (25) is equivalent to minimising $\| Z\hat{\Pi}'_{OLS} - Z\hat{\Pi}' \|_F^2 + \lambda \text{rank}(Z\hat{\Pi}')$. The minimum is achieved when $Z\hat{\Pi}'_{OLS}$ is the best approximation to $Z\hat{\Pi}'$ among matrices with a restricted rank, in particular, $(Z\hat{\Pi}') = \mathcal{H}_\lambda^*(Z\hat{\Pi}_{OLS}')$. This yields the estimator of $\Pi$ as

$$\hat{\Pi} = \hat{V} \mathcal{H}_\lambda^*(D) D^{-1} \hat{V}' \hat{\Pi}_{OLS}, \quad (26)$$

where $\hat{U} \hat{D} \hat{V}'$ is the SVD of $Z\hat{\Pi}'_{OLS}$. The method can be robustified by adding a ridge penalty (Mukherjee & Zhu, 2011), in which case the objective function becomes
\[ \| Y - Z\Pi \|^2_F + \lambda \left[ \omega \text{rank}(\Pi) + \frac{1 - \omega}{2} \| \Pi \|_F^2 \right]. \]  

(27)

The robustified estimator is found by constructing augmented data matrices \( Y^* = (Y', 0_p \times p)' \), \( Z^* = (Z', \sqrt{\lambda(1 - \omega)I_p})' \) and minimising \( \| Y^* - Z^*\Pi \|^2_F + \lambda \text{rank}(\Pi) \).

### 4.1.4 Reduced-rank lasso

A natural idea is to combine the rank restriction and lasso penalisation into one joint estimation problem (Lian et al., 2015) and to define the estimator of \( \Pi \) as

\[ \hat{\Pi} = \arg \min_{\Pi : \text{rank}(\Pi) \leq r} \left\{ \frac{1}{2} \| Y - Z\Pi' \|^2_F + \lambda \| \Pi \|_1 \right\}. \]  

(28)

The algorithm suggested in Lian et al. (2015) to solve the problem is the proximal gradient descent algorithm. This algorithm is designed for solving the problem \( \min_{x \in \mathcal{C}} [f(x) + g(x)] \), where \( f \) is a convex and differentiable function, \( g \) is a convex function and \( \mathcal{C} \) is a closed convex set. For a chosen decreasing sequence of positive numbers \( \alpha_k \), the general algorithm iterates two steps:

1. **Proximal step**: Minimising only \( g \) while keeping the auxiliary solution \( s^{(k+1)} \) close to the previous update \( x^{(k)} \). This is carried out by penalising the distance \( \| x^{(k)} - s \| \). The auxiliary update \( s^{(k+1)} \) is not restricted to the set \( G \). Altogether,

\[ s^{(k+1)} = \arg \min_s \left\{ \frac{1}{2\alpha_k} \| s - x^{(k)} \|^2 + g(s) \right\}. \]  

(29)

2. **Gradient step**: Minimising \( f \) by a gradient method and then projecting the minimum on \( G \),

\[ x^{(k+1)} = P_G \left( s^{(k+1)} - \alpha_k \nabla f(s^{(k+1)}) \right), \]  

(30)

where \( P_G(\cdot) \) is a projection on the set \( G \).

To solve (28), let \( f(\Pi) = (1/2) \| Y - Z\Pi' \|^2_F \), \( g(\Pi) = \lambda \| \Pi \|_1 \) and \( G = \{ \Pi \in \mathbb{R}^p \times p : \text{rank}(\Pi) = r \} \). The two steps of the proximal gradient descent algorithm become

1. **Soft-thresholding**: The proximal step is \( S^{(k+1)} = \text{argmin}_S \left\{ \frac{1}{2\alpha_k} \| S - \hat{\Pi}^{(k)} \|^2_F + \lambda \| S \|_1 \right\} \) and the solution is

\[ S^{(k+1)} = S_\lambda \left( \hat{\Pi}^{(k)} \right), \]  

(31)

where \( S_\lambda(\cdot) \) is defined in Equation (7).

2. **Minimising the sum of squared errors and projecting the update onto the space of matrices with rank \( r \)

\[ \hat{\Pi}^{(k+1)} = \mathcal{H}_{d_r}^* \left( S^{(k)} + \alpha_k \left( Y - ZS^{(k)} \right)'Z \right), \]  

(32)

where \( d_r \) is the \( r \)th largest singular value of the matrix in the parentheses, and \( \mathcal{H}_{d_r}^* \) is the hard-thresholding operator for an SVD.
The implementation of the proximal gradient descent algorithm suffers from two issues. First, the proximal gradient method is proven to converge when the set $G$ is closed and convex. However, the set of all matrices of rank at most $r$ is not convex. Therefore, the convergence is not guaranteed, although convergence in simulations was reported in Lian et al. (2015). Second, we found no recommendation regarding how to choose the sequence $\alpha_k$. The sequences that we tried lead usually to satisfactory convergence when the dimension $p$ is low, but often lead to diverging behavior for high $p$.

### 4.2 Estimation Through Factorisation $\Pi = \alpha \beta'$

This subsection deals with the methods that take cointegration directly into account by estimating the factorisation $\Pi = \alpha \beta'$, $\alpha, \beta \in \mathbb{R}^p \times r$. Before presenting the penalisation methods, we recall the Johansen procedure (Johansen, 1995), which is a popular method of fitting cointegrated VEC models using the maximum-likelihood principle.

The log-likelihood function has the form

$$
\ell(\alpha, \beta; Y, Z) \propto -N \log |\Sigma| - \frac{1}{2} \text{tr}(Y - Z \beta \alpha') \Sigma^{-1}(Y - Z \beta \alpha')'.
$$

For a given cointegration rank $r$, the columns of $\hat{\beta}$ can be found as the eigenvectors $v_1, \ldots, v_r$ corresponding to the $r$ largest eigenvalues $\lambda_i$ of the eigenvalue problem

$$
\lambda_i S_{zz} v_i = S_{zy} S_{yy}^{-1} S_{yz} v_i, \quad i = 1, \ldots, p,
$$

satisfying the normalisation condition

$$
v_i' S_{zz} v_j = \begin{cases} 
1 & \text{for } i = j \\
0 & \text{for } i \neq j
\end{cases}
$$

where the moment matrices $S_{zz}$ are defined in (20). Any basis of $\text{sp}(\hat{\beta})$ can serve as $\hat{\beta}$, because any linear combination of the cointegration vectors yields a cointegration vector.

Once $\hat{\beta}$ is determined, we construct new stationary covariates $Z \hat{\beta} = U \in \mathbb{R}^N \times r$ and turn the VEC model (2) into a standard linear regression model $Y = U \alpha' + \varepsilon$ with no restrictions on the rank of the parameter matrix $\alpha$. The maximum likelihood estimator (MLE) of $\alpha$ and $\Sigma$ are

$$
\hat{\alpha} = S_{yz} (\beta' S_{zz} \beta)^{-1},
$$

$$
\hat{\Sigma} = S_{yy} - S_{yz} \beta' (\beta' S_{zz} \beta)^{-1} \beta S_{zy}.
$$

The chosen form of $\hat{\beta}$ affects $\hat{\alpha}$, however, $\text{sp}(\hat{\alpha})$ is invariant. Furthermore, $\hat{\Pi} = \hat{\alpha} \hat{\beta}'$ is always unique. The presented estimation procedure gives some flexibility for choosing $\hat{\beta}$, but not for $\hat{\alpha}$.

An alternative estimation procedure that uses the least squares method identifies $\hat{\alpha}$ first and then $\hat{\beta}$ is identified for given $\hat{\alpha}$ (Reinsel & Velu, 1998). The estimators are $\hat{\alpha} = \Gamma^{-1/2}(v_1, \ldots, v_r), \hat{\beta} = S_{zz}^{-1} S_{zy} \Gamma \hat{\alpha}$, where $\Gamma$ is an arbitrary positive-definite matrix, $(v_1, \ldots, v_r)$ are eigenvectors corresponding to the $r$ largest eigenvalues $(\lambda_1, \ldots, \lambda_r)$ of the matrix $\Gamma^{1/2} S_{yz} S_{zz}^{-1} S_{zy} \Gamma^{1/2}$ and normalised as $||v_i||_2 = 1$, for $i = 1, \ldots, r$. These estimators satisfy $\beta' S_{zz} \beta = \text{diag}(\lambda_1, \ldots, \lambda_r)$ and $\alpha' \Gamma \alpha = \mathbf{I}_r$. The choice $\Gamma = \hat{\Sigma}^{-1}$ leads to the ML estimates that are asymptotically efficient.
4.2.1 Penalising $\beta$

A procedure with a lasso penalty on $\beta$ was introduced in Wilms & Croux (2016). The original method also incorporates additional penalties on $\mu$ and $\Omega = \Sigma^{-1}$, but we ignore those here for simplicity. The estimators are derived from the penalised negative log-likelihood

$$\min_{\alpha, \beta, \Sigma} \left\{ \frac{1}{N} \text{tr}\left( (Y - Z\beta^*)\Sigma^{-1}(Y - Z\beta^*)' \right) + \log|\Sigma| + \lambda ||\beta||_1 \right\}. \quad (38)$$

For identifiability reasons, the estimator $\hat{\alpha}$ is constrained by condition $\hat{\alpha}'\Sigma^{-1}\hat{\alpha} = I_r$. Restricting $\hat{\alpha}$ instead of $\hat{\beta}$ is convenient for calculations. The solution of (38) is found by an algorithm that iterates three steps:

1. **Solving for a conditional on $\beta$ and $\Sigma$:** The minimisation problem reduces to

   $$\hat{\alpha}|(\beta, \Sigma) = \arg \min_{\alpha} \frac{1}{N} \text{tr}\left( (Y - Z\beta^*)\Sigma^{-1}(Y - Z\beta^*)' \right), \quad \text{subject to } \alpha^2 = I_r. \quad (39)$$

   This is a weighted Procrustes problem and the solution is $\hat{\alpha} = \Sigma^{1/2}VU'$, where $U$ and $V$ are left and right singular vector matrices from the SVD of $\beta'Z\Sigma^{-1/2} = UDV'$.

2. **Solving for $\beta$ conditional on $\alpha$ and $\Sigma$:** The minimisation problem reduces to

   $$\hat{\beta}|(\alpha, \Sigma) = \arg \min_{\beta} \frac{1}{N} \text{tr}\left( (Y\Sigma^{-1/2}\alpha^* - Z\beta)(Y\Sigma^{-1/2}\alpha^* - Z\beta)' \right) + \lambda ||\beta||_1, \quad (40)$$

   with $\alpha^* = \Sigma^{-1/2}\alpha$. This is a multivariate lasso regression of a response matrix $Y\Sigma^{-1/2}\alpha^*$ on $Z$.

3. **Solving for $\Sigma$ conditional on $\alpha$ and $\beta$:** The estimator $\hat{\Sigma}$ is the usual maximum likelihood estimator (37).

The starting values for $\beta^{(0)}$ can be chosen as the first $r$ eigenvectors of the matrix diag($S_{xz}$)$^{-1}$ diag($S_{yy}$)$^{-1}$ $S_{yz}$ and $\Sigma^{(0)} = I_p$ (Wilms & Croux, 2016). If it is not important to have a sparse $\Sigma$, we propose instead to determine $\alpha^{(0)}$ by regressing $Y$ on $Z\beta^{(0)}$ and set $\Sigma^{(0)} = (Y - Z\beta^{(0)}\alpha^{(0)})'(Y - Z\beta^{(0)}\alpha^{(0)})/N$. Note that an appropriate choice of starting values can be more important in high-dimensional settings (Wilms & Croux, 2016).

There is no proof of convergence of the algorithm, only reports of good performance in simulations. Implementation of the iterative algorithm in R is provided in the supplementary material to Wilms & Croux (2016). We discovered a minor discrepancy between the theoretical form of the algorithm and the actual implementation. Namely, the implemented algorithm selects the optimal value of the tuning parameter $\lambda$ by crossvalidation in each iteration of step 2; therefore, $\lambda$ varies over the iterations. The resulting estimators cannot be associated with a solution of (38) for a specified value of $\lambda$. In practice, we cannot track the amount of shrinkage of $\hat{\Pi}$ as a function of $\lambda$. It would be possible to correct the implementation, so that the same $\lambda$ is used in all iterations. However, it would substantially increase the running time and make computations with larger dimensions nearly prohibitive. Therefore, we stick to the original implementation.

Imposing a lasso penalty on $\beta$ leads to biased estimation of the cointegration space. A sparse $\hat{\beta}$ can only induce sparsity in $\hat{\Pi} = \hat{\alpha}\hat{\beta}'$ if it contains a large portion of zeros that make some of the rows identically zero. This can be achieved directly by a penalty that enforces row sparsity of $\beta$.
(Chen & Huang, 2012). The row sparsity means that some variables are forced not to enter any of the cointegration relationships, which seems to be a too strict restriction for most cointegrated processes.

### 4.2.2 Penalising \( \alpha \)

If we prioritise unbiased estimation of \( \beta \) and thus of the cointegration space, it is more meaningful to impose penalty on \( \alpha \) instead of \( \beta \). The following simple procedure was proposed in Levakova et al. (2022). We first estimate \( \beta \) as in the Johansen procedure (34). Then we find \( \hat{\alpha} \) by minimising the penalised sum of squared residuals

\[
\hat{\alpha} = \arg \min_{\alpha \in \mathbb{R}^p \times r} \left\{ \frac{1}{2} ||Y - Z\hat{\beta}\alpha||_F^2 + \lambda \left[ \alpha ||\alpha||_1 + \frac{1 - \alpha}{2} ||\alpha||_F^2 \right] \right\}. \tag{41}
\]

It is still not guaranteed that \( \hat{\Pi} = \hat{\alpha}\hat{\beta}' \) is a sparse matrix. Zeros in \( \alpha \) mean that some of the univariate subprocesses of \( \{x_n\} \) are not adjusting the disequilibria in some of the cointegration relationships.

### 4.2.3 Penalisation in a process transformed into stationary components and stochastic trends

The decomposition \( \Pi = \alpha\beta' \) is used in Liao & Phillips (2015) to transform the process \( \{x_n\}_{n=0}^N \) into stationary and nonstationary components, and the penalised estimation is performed on the transformed process. Let \( \alpha_1 \) and \( \beta_1 \) be full rank \( p \times (p - r) \) matrices of orthogonal complements, satisfying \( \alpha_1'\alpha = 0_{(p - r) \times r} \) and \( \beta_1'\beta = 0_{(p - r) \times r} \), respectively. Without loss of generality, we assume that \( \alpha_1'\alpha_1 = I_{p - r} \) and \( \beta_1'\beta_1 = I_{p - r} \). The matrices \( \alpha_1 \) and \( \beta_1 \) can be constructed from the normalised left and right eigenvectors that correspond to the zero singular values of \( \Pi \). Define a matrix \( F = [\beta, \alpha_1]' \). Then

\[
F\Pi = \begin{pmatrix} \beta'\alpha' & 0 \\ 0_{(p - r) \times p} \end{pmatrix} \quad \text{and} \quad F\Pi F^{-1} = \begin{pmatrix} \beta'\alpha & 0_{r \times (p - r)} \\ 0_{(p - r) \times r} & 0_{(p - r) \times (p - r)} \end{pmatrix}. \tag{42}
\]

Thus, the transformed process \( w_n = Fx_n \) satisfies the VEC model

\[
\Delta w_n = \begin{pmatrix} \beta'\alpha & 0 \\ 0 & 0 \end{pmatrix} w_{n-1} + \xi_n, \tag{43}
\]

where \( \xi_n = F\epsilon_n \). It means that the transformation decomposes the process into \( r \) stationary components and \( p - r \) random walks.

The estimator of \( \Pi \) is defined as

\[
\hat{\Pi} = \arg \min_{\Pi \in \mathbb{R}^p \times p} \left\{ ||Y - Z\Pi||^2_F + N \sum_{j=1}^p \lambda_j \||\hat{F}\Pi_j||_2 \right\}, \tag{44}
\]

where \( \hat{F} \) is a matrix consisting of the normalised left eigenvectors of a preliminary estimate \( \hat{\Pi}_0 \). The procedure does not require a prespecified cointegration rank, but tries to reduce the rank to the optimal number by identifying the zero rows of \( F\Pi \). Theoretical properties of this estimator are derived in Liao & Phillips (2015). Unfortunately, no algorithm to solve (44) is provided and we have not found any information regarding the practical implementation in the literature.
4.3 Estimation Through QR Decomposition

The following method was introduced in Liang & Schienle (2019) and makes use of the QR decomposition $\Pi' = QR$. The matrix $Q \in \mathbb{R}^{p \times p}$ is an orthonormal matrix and $R \in \mathbb{R}^{p \times p}$ is an upper triangular matrix. The matrix $\Pi$ can be written in a block-matrix form as

\[ \Pi = R'Q' = \begin{pmatrix} R'_1 & R'_2 \end{pmatrix} \begin{pmatrix} Q'_1 & 0 \\ 0 & Q'_2 \end{pmatrix} = \begin{pmatrix} R'_{11} & R'_{12} \\ R'_{12} & R'_{22} \end{pmatrix} \begin{pmatrix} Q'_1 & 0 \\ 0 & Q'_2 \end{pmatrix}, \]  

(45)

where $Q'_1 \in \mathbb{R}^{r \times p}$, $Q'_2 \in \mathbb{R}^{(p-r) \times p}$ are orthogonal matrices; $R'_{11} \in \mathbb{R}^{r \times r}$, $R'_{12} \in \mathbb{R}^{(p-r) \times r}$ and $R'_{22} \in \mathbb{R}^{(p-r) \times (p-r)}$. The matrix $Q$ can be seen as a $p \times p$ extended form of $\beta$, where the first $r$ columns span the cointegration space, while $R'$ represents an extended form of $\alpha$ that has zeros in the last $p - r$ columns, so $R_{22} = 0_{(p-r) \times (p-r)}$. Therefore, we expect that $R'_{22}$ has lower diagonal elements close to zero.

This motivates the idea to impose penalty on the columns of $R'$ to shrink the block $R_{22}$ towards 0. The objective function to be minimised is

\[ \min_{R \in \mathbb{R}^{p \times p}} \left\{ \| Y - ZQ\|_F^2 + \sum_{j=1}^{p} \lambda_j w_j \| R'_{jj} \|_2^2 \right\}, \text{ with } w_j = \| R'_{jj} \|_2, \]  

(46)

where $\lambda$ is the tuning parameter and $\gamma > 0$ is a constant that is fixed in advance. In Liang & Schienle (2019), they use $\gamma \in \{2, 3, 4\}$ and suggest to opt for a lower value when the sample size is small. The matrix $Q$ is an orthogonal matrix obtained from a QR decomposition of a preliminary estimate $\hat{\Pi}_0$, for example, the OLS estimate. Similarly, the weights $w_j$ are calculated from the QR decomposition of $\hat{\Pi}_0$. Alternative elementwise penalisation is used in Chen & Schienle (2022),

\[ \min_{R \in \mathbb{R}^{p \times p}} \left\{ \text{tr}\left[ (Y - ZQ)^{-1}(Y - ZQ) \right] + \sum_{i,j=1}^{p} \lambda_{ij} |R_{ij}| \right\}. \]  

(47)

4.4 Estimation Through Singular Value Decomposition

4.4.1 Penalising the nuclear norm

This estimation procedure was first introduced in Yuan et al. (2007) in the context of reduced-rank regression problems. A practical summary of the method with focus on cointegrated VAR models is provided in Signoretto & Suykens (2012). The argument to use the nuclear norm penalty in fitting a cointegrated VAR model is as follows. The rows $\Pi_i, i = 1, \ldots, p$, belong to a linear subspace $B$ of dimension less than $p$. Let $\{\eta_1, \ldots, \eta_p\}$ be a set of basis vectors of $\mathbb{R}^p$ and $A$ a subset of $\{1, \ldots, p\}$ such that $B \subseteq \text{span}\{\eta_i; i \in A\}$. Assume the basis vectors $\eta_i$’s are known and consider the problem of simultaneous selection of the basis vectors and parameter estimation in a model

\[ Y = F\Psi + E, \]  

(48)

where $F_{ij} = Z\eta_i$ and $\Psi \in \mathbb{R}^{p \times p}$ such that $(\eta_1, \ldots, \eta_p)\Psi = \Pi'$. This is carried out by minimising
adding a ridge penalty term to the objective function (50) is recommended, which leads to
utilise an SVD, $\Pi' = UDV'$, from which it follows that $\Psi = DV'$ and $\|\Psi_i\cdot\| = \|d_i(\Pi)V_i'\| = d_i(\Pi)$. The minimisation problem (49) is therefore equivalent to

$$\min_{\Psi} \left\{ \| Y - F\Psi \|_F^2 + \lambda \sum_{i=1}^p \| \Psi_i \|_2 \right\}. \quad (49)$$

It turns out that we do not need to know the basis vectors $\eta_i$, $i = 1, \ldots, p$, because we can utilise an SVD, $\Pi' = UDV'$, from which it follows that $\Psi = DV'$ and $\|\Psi_i\cdot\| = \|d_i(\Pi)V_i'\| = d_i(\Pi)$. The minimisation problem (49) is therefore equivalent to

$$\min_{\Pi} \left\{ \| Y - \Pi\Psi' \|_F^2 + \lambda\|\Pi\|_* \right\}. \quad (50)$$

It can be shown that the nuclear norm $\|\Pi\|_*$ is the convex envelope of rank($\Pi$), which makes (50) the tightest convex relaxation to the problem $\min_{\Pi} \{\| Y - Z\Pi' \|_F^2 + \lambda\|\Pi\|_* \}$, discussed in Section 4.1.3. The convexity property implies that any local solution found by a convergent algorithm is globally optimal. One possibility to minimise (50) is to use a second-order cone program solver such as SDPT3 (Tütüncü et al., 2003). An alternative is to apply a special case of the fast iterative shrinkage-thresholding algorithm (FISTA):

- Iteration $k = 0$: $\hat{\Pi}^{(0)} = \hat{\Pi}^{(1)}$, $t_1 = 1$, $L = N\|S_{zz}\|_F$.
- Iteration $k \geq 1$:

$$\text{Update } \hat{\Pi}: \hat{\Pi}^{(k)} = \frac{2}{L} \left( \hat{\Pi}^{(k)} - \frac{1}{L} (Z\hat{\Pi}^{(k)'})' - Y)'Z \right). \quad (51)$$

$$\text{Update auxiliary variables}: t_{k+1} = \frac{1}{2} \left( 1 + \sqrt{1 + 4t_k^2} \right), \quad r_{k+1} = \frac{t_k - 1}{t_{k+1}}, \quad \hat{\Pi}^{(k+1)} = \hat{\Pi}^{(k)} + r_{k+1} \left( \hat{\Pi}^{(k)} - \hat{\Pi}^{(k-1)} \right). \quad (52)$$

The sequence $\hat{\Pi}^{(k)}$ is an auxiliary sequence of linear combinations of the previous two updates $\hat{\Pi}^{(k-1)}$ and $\hat{\Pi}^{(k-2)}$. The convergence properties of this procedure are documented in Beck & Teboulle (2009).

To make the estimator more robust and to prevent numerical issues when $\lambda$ is close to 0, adding a ridge penalty term to the objective function (50) is recommended, which leads to the minimisation problem

$$\hat{\Pi} = \arg \min_{\Pi} \left\{ \| Y - Z\Pi' \|_F^2 + \lambda \left[ \omega \|\Pi\|_* + \frac{1}{2} \omega \|\Pi\|_F^2 \right] \right\}. \quad (53)$$

The estimator $\hat{\Pi}$ is found by using augmented data matrices $Y' = (Y', 0_p \times p)'$, $Z' = (Z', \sqrt{\lambda(1 - \omega)}/21_p)'$ and performing the FISTA algorithm with a tuning parameter $\lambda_0$.

### 4.4.2 Adaptive nuclear norm penalisation

The previous method was modified in two aspects in Chen et al. (2013), where the estimator $\hat{\Pi}$ is defined as
\[ \hat{\Pi} = \arg \min_{\Pi} \left\{ \frac{1}{2} \| \mathbf{Y} - \mathbf{Z}\Pi' \|^2_F + \lambda \| \mathbf{Z}\Pi' \|_{sw} \right\}. \tag{54} \]

First, the objective function contains the norm of \(\mathbf{Z}\Pi'\) instead of the norm of \(\Pi\). This is inspired by the rank penalty from Section 4.1.3, where the minimum of \(\| \mathbf{Y} - \mathbf{Z}\Pi' \|^2_F + \lambda \text{rank}(\mathbf{Z}\Pi')\) was found to be the same as the minimum of \(\| \mathbf{Y} - \mathbf{Z}\Pi' \|^2_F + \lambda \text{rank}(\mathbf{Z}\Pi')\). The solution is expected to yield a low-rank estimator of \(\mathbf{Z}\Pi'\) and thus possibly also a sparser form of \(\Pi\). Second, the nuclear norm \(\| \Pi \|_*\) is generalised by allowing weights \(w = (w_1, \ldots, w_p)'\) satisfying \(0 \leq w_1 \leq \ldots \leq w_p\) and defining a weighted nuclear norm \(\| \Pi \|_{sw} = \sum_{i=1}^{p} w_i d_i(\Pi)\).

Due to penalising \(\mathbf{Z}\Pi'\), an explicit solution of (54) can be found using the soft-thresholding operator \(S_w^*\), Equation (7),

\[ \mathbf{Z}\hat{\Pi}' = S_w^*(\mathbf{Z}\Pi'_{OLS}) = \hat{U}\text{diag}(S_{iw}(\hat{D}))\hat{V}', \quad \hat{\Pi} = \hat{V}\text{diag}(S_{iw}(\hat{D}))\hat{D} - \hat{V}'\hat{\Pi}_{OLS}, \tag{55} \]

where \(\hat{U}\hat{D}\hat{V}'\) is the SVD of \(\mathbf{Z}\Pi'\). The only difference from the rank penalised estimator (26) is that the hard-thresholding operator is replaced with a soft-thresholding operator.

The adaptive nuclear norm penalisation method can be robustified in a similar way as the penalised rank regression, by incorporating a ridge penalty term. The ridge penalty is again imposed on \(\mathbf{Z}\Pi'\) instead on \(\Pi\), so the objective function to be minimised is

\[ \frac{1}{2} \| \mathbf{Y} - \mathbf{Z}\Pi' \|^2_F + \frac{\lambda}{2} \| \mathbf{Z}\Pi' \|_{sw} + \frac{\omega}{2} \| \mathbf{Z}\Pi' \|_2^2 \]. \tag{56} \]

The minimiser is \(\hat{\Pi} = \hat{\Pi}^{(\omega)}/[1 + \lambda(1 - \omega)]\), where \(\hat{\Pi}^{(\omega)}\) is the adaptive nuclear penalty estimator with the tuning parameter \(\lambda\omega\).

4.4.3 Penalisation of all components of singular value decomposition of \(\Pi\)

An algorithm to estimate a sparse form of SVD of \(\Pi\) has been introduced in Chen et al. (2012). The SVD splits \(\Pi\) into a sum of \(r\) unit-rank matrices, \(\Pi' = UDV' = \sum_{k=1}^{r} d_k U \cdot k V' \cdot k = \sum_{k=1}^{r} \Pi^{(k)}\). The algorithm relaxes the orthogonality of the left and right singular vectors and minimises the following objective function with respect to the triplets \((d_k, U \cdot k, V \cdot k), k = 1, \ldots, r:\]

\[ \frac{1}{2} \| \mathbf{Y} - \mathbf{Z} \sum_{k=1}^{r} d_k U \cdot k V' \cdot k \|^2_F + \sum_{k=1}^{r} \lambda_k \sum_{i=1}^{p} \sum_{j=1}^{p} w_{ijk} |d_k U_{ik} V_{jk}| = \]

\[ = \frac{1}{2} \| \mathbf{Y} - \mathbf{Z} \sum_{k=1}^{r} d_k U \cdot k V' \cdot k \|^2_F + \sum_{k=1}^{r} \lambda_k \left( \sum_{i=1}^{p} w_{ik}^{(U)} |U_{ik}| \right) \left( \sum_{j=1}^{p} w_{jk}^{(V)} |V_{jk}| \right). \tag{57} \]

The multiplicative form of the penalty not only promotes sparsity in the singular vectors, but also shrinks the singular values of each SVD layer towards 0.

The objective function is non-convex. The optimisation algorithm optimises the unit rank layers \(\Pi^{(k)} = d_k U \cdot k V' \cdot k\) in parallel. Therefore, we first present the algorithm for \(r = 1\).

For \(\text{rank}(\Pi) = 1\): The problem reduces to minimising

\[ d \in \mathbb{R}, \ U \in \mathbb{R}^p, \ V \in \mathbb{R}^p \left\{ \frac{1}{2} \| \mathbf{Y} - \mathbf{Z}dUV' \|^2_F + \lambda \sum_{i=1}^{p} \sum_{j=1}^{p} w_{ij} |d_{U} V_{f_j}| \right\}, \tag{58} \]
where $\|U\|_2 = \|V\|_2 = 1$. The weights are decomposed as $w_{ij} = w^{(d)}_{ij}w^{(U)}_{ij}w^{(V)}_{ij}$ and chosen in a data-driven manner using an SVD of an initial estimate $\Pi'_0 = \hat{d}_0\hat{U}\hat{V}'_0$ as

$$w^{(d)} = |\hat{d}_0|^{-\gamma}, \quad w^{(U)} = (w^{(U)}_1, ..., w^{(U)}_p)' = |\hat{U}_i|^{-\gamma}, \quad w^{(V)} = (w^{(V)}_1, ..., w^{(V)}_p)' = |\hat{V}_i|^{-\gamma}, \quad (59)$$

where $\gamma$ is a prespecified non-negative parameter ($\gamma = 2$ is used in Chen et al. (2012)) and the absolute value is applied element-wise. The algorithm alternates between estimating $(d, V)$ for fixed $U$ and estimating $(d, U)$ for fixed $V$ until convergence.

1. **Fix $U$, minimise with respect to $(d, V)$**: The problem (58) can be re-formulated as

$$\min_{v^*} \left\{ \frac{1}{2}\|y - Zv^*\|_2^2 + \lambda^{(V)}\|v^*\|_1 \right\}, \quad (60)$$

where $v^* = \text{diag}(dw^{(V)})V$, $y = \text{vec}(Y)$, $Z^{(V)} = \text{diag}(w^{(V)})^{-1}\otimes(ZU)$ and $\lambda^{(V)} = \lambda w^{(d)}(\sum_{i=1}^p |U_i|)$. It is a standard lasso problem with unknown parameters $v^*$.

2. **Fix $V$, minimise with respect to $(d, U)$**: The problem (58) can be re-formulated as

$$\min_{u^*} \left\{ \frac{1}{2}\|y - ZUu^*\|_2^2 + \lambda^{(U)}\|u^*\|_1 \right\}, \quad (61)$$

where $u^* = \text{diag}(dw^{(U)})U$, $Z^{(U)} = V\otimes Z\text{diag}(w^{(U)})^{-1}$ and $\lambda^{(U)} = \lambda w^{(d)}(\sum_{i=1}^p |V_i|)$. It is a standard lasso problem with unknown parameters $u^*$.

For rank$(\Pi) = r > 1$: The problem is decomposed into $r$ parallel unit-rank problems by splitting the response matrix $Y$ into $r$ additive layers $Y = \sum_{k=1}^r Y_k$ according to the SVD of the initial estimate $\Pi'_0 = \sum_{k=1}^r \Pi^{(k)}_0 = \sum_{k=1}^r \hat{d}_{k,0}\hat{U}_{k,0}\hat{V}'_{k,0}$.

1. For each $k \in \{1, ..., r\}$:
   a. Construct adaptive weights $w^{(d)}_k = |\hat{d}_{k,0}|^{-\gamma}$, $w^{(U)}_k = |\hat{U}_{k,0}|^{-\gamma}$ and $w^{(V)}_k = |\hat{V}_{k,0}|^{-\gamma}$.
   b. Construct the layer $Y_k = Y - Z(\Pi'_0 - \Pi^{(k)}_0)$.
   c. Find an updated estimate $(\hat{d}_{k,0}, \hat{U}_{k,0}, \hat{V}_{k,0})$ by running the unit-rank algorithm with $Y_k$ and $Z$.

2. The final estimator of $\Pi$ is given by $\hat{\Pi} = (\sum_{k=1}^r \hat{d}_{k,0}\hat{U}_{k,0}\hat{V}'_{k,0})'$.

The algorithm can be further improved by iterating step 1 until convergence.

### 4.5 Determination of the Cointegration Rank

If a particular method of estimating $\Pi$ requires to set $r = \text{rank}(\hat{\Pi})$, a popular way of selecting $r$ is the Johansen's trace and maximum eigenvalue test (Johansen, 1995). It can be shown that the maximum of the loglikelihood $\ell(a, \beta, \Sigma|r)$ given a cointegration rank $r$ is $\ell_{\max}(r) = \log|S_{\Pi}| + \sum_{i=1}^r \log(1 - \lambda_i)$, where $\lambda_i, i = 1, ..., p$ are eigenvalues of the eigenvalue problem (34) in descending order. The trace test and the maximum eigenvalue test are...
likelihood-ratio tests of a null hypothesis $H_0: r = r_0$ against the alternative $H_a: r = p$ (trace test) or $H_a: r = r_0 + 1$ (maximum eigenvalue test). The test statistics are

- trace test: $-2\ell_{\text{max}}(r_0) - \ell_{\text{max}}(p) = -N \sum_{i=r_0 + 1}^{p} \log(1 - \lambda_i)$ (62)

- maximum eigenvalue test: $-2\ell_{\text{max}}(r_0) - \ell_{\text{max}}(r_0 + 1) = -N \log(1 - \lambda_{r_0 + 1})$ (63)

Both test statistics follow nontrivial distributions; the critical values of these distributions were calculated numerically for $p \leq 11$. To determine the cointegration rank, the trace test or maximum eigenvalue test is performed with $r_0 \in \{0, \ldots, p - 1\}$, starting with $r_0 = 0$ and increasing $r_0$ by one until the null hypothesis is rejected. The lowest $r_0$, for which the null hypothesis is not rejected, is the estimate of the cointegration rank $\hat{r}$.

Determination of the rank by the trace or maximum eigenvalue tests is not viable for large $p$, due to unknown critical values of the tests. The critical values can possibly be determined by bootstrap methods (Cavaliere et al., 2012, 2014), however, computational times become prohibitive for large $p$. A more general way of estimating $r$ is the rank selection criterion (RSC) (Bun et al., 2011). The method is based on the rank penalisation from Section 4.1.3, where the tuning parameter is set as

$$\hat{\lambda} = \frac{2\|Y - Z(Z'Z)^{-1}Z'Y\|_F^2}{p(N - \text{rank}(Z))(p + \text{rank}(Z))}. \quad (64)$$

The rank estimated by RSC corresponds to the number of singular values of $Z\hat{\Pi}_i^{\text{OLS}}$ that are larger or equal to $\sqrt{\hat{\lambda}}$. The value of $\hat{\lambda}$ was derived as an optimal threshold to distinguish significant singular values from those induced by noise.

Another option is the procedure of Pedroni et al. (Pedroni et al., 2015). It starts with projecting the series $\{x_n\}_{n=1}^{N}$ on the deterministic trend, which is a linear trend for the model assumed here. This provides a new detrended series of the residuals $\{\hat{u}_n\}_{n=1}^{N}$, which are estimates of the stochastic component of $\{x_n\}_{n=1}^{N}$, and a cumulative series $\{\hat{s}_n\}_{n=1}^{N}$, where $\hat{s}_n = \sum_{i=1}^{n} \hat{u}_i$. These time series are used to calculate an estimate of contemporaneous variance $\hat{\Omega}_N = 1/N \sum_{i=1}^{N} \hat{u}_i\hat{u}_i'$ and an estimate of a long-run variance matrix $\hat{\Omega}_N = 2/N^2 \sum_{i=1}^{N} \hat{s}_i\hat{s}_i'$. The test statistic for testing $H_0: r = r_0$ is then

$$2N \sum_{i=r_0 + 1}^{p} \hat{\lambda}_i, \quad (65)$$

where $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_p$ are the ordered eigenvalues of $\hat{\Omega}_N^{\text{OLS}}\hat{\Omega}_N^{-1}$. Under the null hypothesis, the distribution of the test statistics converges weakly to $\text{tr}\left(\int_0^1 W_2(s)W_2(s)' ds\left(\int_0^1 Q_2(s)Q_2(s)' ds\right)^{-1}\right)$, where $W_2(s)$ is an orthogonal complement to a projection of a $(p - r)$-dimensional Brownian motion on a linear trend and $Q_2(s) = \int_0^s W_2(r) dr$. The critical values are obtained numerically from simulations for a particular dimension $p$. The test statistic can be used in the same recursive manner as in the trace and maximum eigenvalue tests.
4.6 Illustration of Shrinkage Effects

We first conduct a simulation study to illustrate how the methods shrink $\hat{\Pi}$ in different manners as a function of the size of the tuning hyperparameter. In the next section, we conduct a larger simulation study for fixed penalisation (selected by cross-validation) to evaluate and compare the methods by different error measures.

We simulated a cointegrated process of dimension $p = 30$ with cointegration rank $r = 4$. The true $\Pi$ has 114 nonzero elements out of 900, which are located in 4 rows and the remaining rows contain only zeros. The norm $\|\Pi\|_F$ is approximately 0.9. The simulated data set consists of $N = 30,000$ observations of the vector process to showcase asymptotic behavior of the estimation methods. We show the full regularisation paths, that is, characteristics of the estimates of $\Pi$ (and some other relevant parameters) as a function of the tuning parameters, from zero penalisation to full penalisation (where $\Pi$ is shrunk to a zero matrix) in Figure 1. To achieve the same scales in the plots, the $x$-axis do not show the actual tuning parameter values, but the norm $\|\hat{\Pi}\|_F$ obtained with a given value of the tuning parameter. Thus, a norm of zero corresponds to the largest tuning parameter value giving the largest shrinkage to an estimate of zero, and a norm of 0.9 corresponds to tuning parameters equal to 0 giving no shrinking of $\hat{\Pi}$ compared with nonpenalised estimators (OLS estimator for a nonrestricted rank and Johansen estimator for a restricted rank).

The elastic net estimator tends to shrink the smallest elements of $\Pi$ in absolute value. The elements outside of the four nonzero rows are typically shrunk to 0 first and smaller elements in those four columns are shrunk next, as the tuning parameter is increased, even though the method does not penalise the rank of $\hat{\Pi}$. The regularisation path has a region around $\|\hat{\Pi}\|_F = 0.3$, where the proportion of zeros and the estimated rank change rapidly, while in other regions the proportion of zeros change only little (Figure 1A).

The elastic net penalty imposed on $\alpha$ has the same impact on $\hat{\alpha}$ as the elastic net penalty has on $\hat{\Pi}$, that is, shrinks the smallest elements of $\hat{\alpha}$. The impact of shrinking elements of $\alpha$ on $\hat{\Pi}$ is that with increasing penalisation the four nonzero rows are correctly identified while the other rows are shrunk to zero (Figure 1C). The elements in these rows are shrunk evenly to 0, that is, a whole row is eliminated at a time. This gives in fact a more appropriate form of sparsity to $\hat{\Pi}$ than the standard elastic net penalisation of $\Pi$, because a relatively small shrinkage is needed to pick the most important rows and eliminate the less important rows. Therefore, only little bias is introduced (Figure 1H, red lines).

The penalisation of $\hat{R}$ in the QR decomposition of $\hat{\Pi}$ shrinks whole columns of $R$ (Figure 1D). The last $p - r$ columns corresponding to the non-existing cointegration relationships are among the columns that are shrunk first and $\hat{R}$ has around four columns during the main part of the regularisation path. It means that the rank of $\hat{\Pi}$ is close to the true cointegration rank for a large part of the regularisation path (Figure 1H, middle panel, green line). The shrinkage of $\hat{\Pi}$ is nevertheless gradual and no exact zero elements are created.

The rank penalty, the nuclear penalty and the adaptive nuclear penalty shrink $\hat{\Pi}$ in a similar manner (Figure 1B,E,F). The shrinkage of the singular values does not push any of the elements of $\hat{\Pi}$ to exact zeros and the elements are shrunk approximately evenly. The rank penalty gives a step-wise shrinkage of the singular values, meaning that a singular value is either kept intact or shrunk to 0, when the chosen tuning parameter becomes larger. The regularisation path thus shows ‘break points’, whenever some singular value is dropped out. The nuclear penalty provides a similar, but smoother regularisation path. The adaptive nuclear penalty yields more even shrinkage of $\hat{\Pi}$ than the rank penalty and the nuclear penalty.
FIGURE 1. Illustration of the shrinkage of $\hat{\Pi}$ and other relevant parameters by the penalisation methods. A-G: Regularisation paths of $\hat{\Pi}$ and other parameters being shrunk in the respective estimation methods as a function of $\|\hat{\Pi}\|_F$. The grey lines show the relationship between individual elements of the estimated parameters and the amount of shrinkage corresponding to $\|\hat{\Pi}\|_F$. The dotted lines mark zero level corresponding to the maximum possible shrinkage.

H: The proportion of zeros $p_0$ in $\hat{\Pi}$ as a function of $\|\hat{\Pi}\|_F^2$ (left), rank($\hat{\Pi}$) as a function of $\|\hat{\Pi}\|_F^2$ (middle), the norms $\|\hat{\Pi}\|_F$ resulting from the used sequences of the tuning parameters for each method. The colored lines indicate different methods as indicated in the legend. The dotted lines indicate the norm, the proportion of zeros $p_0$ and the rank of the true $\Pi$.
The lasso penalisation of the SVD of $\Pi$ picks up the most prominent rows and columns of $\Pi$ and shrinks the less important elements in those rows and columns (Figure 1G). Because the rank is set to exactly 4 throughout the regularisation path, at least one element in each nonzero row is always kept.

5 Simulation Study and Comparison of Methods

We performed a simulation study to compare the methods summarised in this paper. The methods were implemented in the R programming environment, version 4.2.0 (R Core Team, 2020). The underlying R code is provided in an R package available in the Supplementary material. If an earlier implementation of a method was available (references in Table 1), we incorporated the existing code into the package with minimum adjustments.

We explored 2 choices of a dimension $p$ combined with 2 choices of the sample size $N$:

1. $p = 50$, with 4 possible choices of the cointegration rank $r \in \{10, 20, 30, 40\}$;
2. $p = 150$, with 4 possible choices of the cointegration rank $r \in \{30, 60, 90, 120\}$;
   a. Small sample: $N = p$;
   b. Large sample: $N = 3p$.

For high $p$ and small $N$ the benefits of the penalisation methods should stand out more clearly. Conversely, the penalisation methods are expected to have less advantage over the standard methods for low $p$ and large $N$. We simulated 20 datasets for each combination of $p$, $N$ and $r$. The matrix $\Pi$ was randomly generated in each simulation run through the Jordan normal form $\Pi = PJP^{-1}$. The matrix $J$ was set as a diagonal matrix with $r$ diagonal elements drawn from a uniform distribution on $(-1, 0)$ and the remaining $p - r$ diagonal elements were set to 0. Elements of $P$ were generated independently from a $N(0, 1)$ distribution.

We performed each estimation method with up to 4 options of the initial estimate of the cointegration rank $\hat{r}^{(0)}$:

1. Full rank $p$ (for the methods that have a rank selecting property)
2. True value $r$
3. Rank $\hat{r}_{RSC}$ estimated by the RSC method
4. Rank $\hat{r}_{P}$ estimated by the method of Pedroni et al. (only for $N > p$)

The tuning parameters for controlling the amount of penalisation were selected using AIC. If a method allows the ridge penalty, the proportion of the non-ridge penalty $\omega$ was set to 0.9. We used $\gamma = 2$ for the estimation through QR decomposition.

The performance of the estimators is quantified by the following measures:

- **Relative sum of squared errors**: $\text{RSSE}(\hat{\Pi}) = ||Y - Z\hat{\Pi}||_F^2 / ||Y||_F^2$
- **Mean squared prediction error**: $\text{MSPE}(\hat{\Pi}) = \frac{1}{MP}\sum_{i=1}^{M} ||x_i - \hat{x}_i||_2$.

For this measure, we simulated $M = 200$ additional values of the process $\{x_i\}_{i=1}^{200}$ not used for fitting the VEC model. The values $\{\hat{x}_i\}_{i=1}^{200}$ are one step predictions based on the knowledge of the previous value $x_{i-1}$.

- **Relative deviation of $\hat{\Pi}$**: $\text{RD}(\hat{\Pi}) = ||\hat{\Pi} - \Pi||_F / ||\Pi||_F$. 
• Angle between \( sp(\Pi) \) and \( sp(\tilde{\Pi}) \): \( \Theta(\Pi, \tilde{\Pi}) = \arccos (\langle \Pi, \tilde{\Pi} \rangle \| \Pi \| \| \tilde{\Pi} \|) \). The angle is defined as a generalised version of the vector angle. It ranges between 0 and \( \pi/2 \). For \( \Pi = 0 \) or \( \tilde{\Pi} = 0 \) we define \( \Theta(\Pi, \tilde{\Pi}) = \pi/2 \).

• Relative shrinkage of \( \tilde{\Pi} \): \( \text{Shrink}(\tilde{\Pi}) = \| \tilde{\Pi} \| / \| \Pi \| \)

• Distance between the true and estimated cointegration rank: \( d(\hat{r}, r) = |\text{rank}(\tilde{\Pi}) - r| \)

• Proportion of zeros in \( \tilde{\Pi}p_0 = \{ (i, j) : \tilde{\Pi}_{ij} = 0 \} / p^2 \)

• Specificity (true positive): \( TP = \frac{|\{ (i, j) : \Pi_{ij} = 0 \land \tilde{\Pi}_{ij} = 0 \}|}{|\{ (i, j) : \Pi_{ij} = 0 \}|} \)

• Sensitivity (true negative): \( TN = \frac{|\{ (i, j) : \Pi_{ij} \neq 0 \land \tilde{\Pi}_{ij} \neq 0 \}|}{|\{ (i, j) : \Pi_{ij} \neq 0 \}|} \)

Table 2. Simulation results for \( p = 50, N = 50 \) (low dimension, small sample size). The results are averages over 100 simulation runs. See main text for definition of estimation methods and error measures. Bold numbers indicate the best results within a column, after excluding those where true cointegration rank was used.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{r}(0) )</th>
<th>RSSE</th>
<th>MSPE</th>
<th>RD</th>
<th>( \Theta(\Pi) )</th>
<th>( d(\hat{r}, r) )</th>
<th>( p_0 )</th>
<th>TP</th>
<th>TN</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic net</td>
<td>( p )</td>
<td>0.01</td>
<td>92.82</td>
<td>3.54</td>
<td>1.26</td>
<td>3.70</td>
<td>25.06</td>
<td>0.01</td>
<td>0.01</td>
<td>0.99</td>
</tr>
<tr>
<td>Lasso + SVD</td>
<td>( r )</td>
<td>0.21</td>
<td>86.44</td>
<td>3.33</td>
<td>1.27</td>
<td>3.48</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso w. fixed</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.01</td>
<td>91.01</td>
<td>3.42</td>
<td>1.25</td>
<td>3.58</td>
<td>23.36</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( \beta )</td>
<td>( r )</td>
<td>0.27</td>
<td>5.27</td>
<td>0.96</td>
<td>1.06</td>
<td>0.87</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( r )</td>
<td>0.27</td>
<td>5.71</td>
<td>0.98</td>
<td>1.05</td>
<td>0.93</td>
<td>22.78</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.71</td>
<td>139.89</td>
<td>3.37</td>
<td>1.35</td>
<td>3.17</td>
<td>22.60</td>
<td>0.13</td>
<td>0.07</td>
<td>0.93</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>( r )</td>
<td>0.94</td>
<td>5.79</td>
<td>1.12</td>
<td>1.52</td>
<td>0.27</td>
<td>23.75</td>
<td>0.86</td>
<td>0.86</td>
<td>0.14</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( r )</td>
<td>0.45</td>
<td>9.39</td>
<td>1.10</td>
<td>1.07</td>
<td>1.14</td>
<td>9.96</td>
<td>0.05</td>
<td>0.07</td>
<td>0.97</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.45</td>
<td>8.86</td>
<td>1.08</td>
<td>1.06</td>
<td>1.13</td>
<td>10.75</td>
<td>0.05</td>
<td>0.06</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 3. Simulation results for \( p = 50, N = 150 \) (low dimension, large sample size). The results are averages over 100 simulation runs. See main text for definition of estimation methods and error measures. Bold numbers indicate the best results within a column, after excluding those where true cointegration rank was used.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{r}(0) )</th>
<th>RSSE</th>
<th>MSPE</th>
<th>RD</th>
<th>( \Theta(\Pi) )</th>
<th>( d(\hat{r}, r) )</th>
<th>( p_0 )</th>
<th>TP</th>
<th>TN</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>( p )</td>
<td>0.21</td>
<td>9.77</td>
<td>0.66</td>
<td>0.54</td>
<td>1.24</td>
<td>22.33</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Elastic net</td>
<td>( p )</td>
<td>0.21</td>
<td>8.99</td>
<td>0.62</td>
<td>0.54</td>
<td>1.20</td>
<td>22.48</td>
<td>0.02</td>
<td>0.02</td>
<td>0.99</td>
</tr>
<tr>
<td>El. net + SVD</td>
<td>( r )</td>
<td>0.33</td>
<td>7.76</td>
<td>0.64</td>
<td>0.56</td>
<td>1.19</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Rank penalty</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.41</td>
<td>5.87</td>
<td>0.60</td>
<td>0.55</td>
<td>1.13</td>
<td>12.95</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso w. fixed</td>
<td>( r )</td>
<td>0.23</td>
<td>11.10</td>
<td>0.76</td>
<td>0.59</td>
<td>1.32</td>
<td>25.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Johansen</td>
<td>( r )</td>
<td>0.43</td>
<td>5.31</td>
<td>0.72</td>
<td>0.79</td>
<td>0.78</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( \beta )</td>
<td>( r )</td>
<td>0.43</td>
<td>5.67</td>
<td>0.72</td>
<td>0.53</td>
<td>1.19</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( r )</td>
<td>0.33</td>
<td>5.22</td>
<td>0.49</td>
<td>0.47</td>
<td>1.07</td>
<td>16.79</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.54</td>
<td>4.55</td>
<td>0.79</td>
<td>0.92</td>
<td>0.64</td>
<td>12.95</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.33</td>
<td>5.31</td>
<td>0.72</td>
<td>0.79</td>
<td>0.78</td>
<td>12.95</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.32</td>
<td>5.22</td>
<td>0.49</td>
<td>0.47</td>
<td>1.07</td>
<td>16.79</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on ( RSC )</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.25</td>
<td>9.58</td>
<td>0.71</td>
<td>0.57</td>
<td>1.27</td>
<td>9.74</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>( r )</td>
<td>0.25</td>
<td>9.58</td>
<td>0.71</td>
<td>0.57</td>
<td>1.27</td>
<td>9.74</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Adapt. nuclear p.</td>
<td>( p )</td>
<td>0.23</td>
<td>11.10</td>
<td>0.76</td>
<td>0.59</td>
<td>1.32</td>
<td>25.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( r )</td>
<td>0.43</td>
<td>7.60</td>
<td>0.65</td>
<td>0.65</td>
<td>1.01</td>
<td>3.94</td>
<td>0.01</td>
<td>0.02</td>
<td>0.99</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( \hat{r}_{RSC} )</td>
<td>0.41</td>
<td>7.48</td>
<td>0.63</td>
<td>0.64</td>
<td>1.01</td>
<td>1.91</td>
<td>0.02</td>
<td>0.03</td>
<td>0.99</td>
</tr>
</tbody>
</table>

International Statistical Review (2023)
• **Time**: Elapsed time (in seconds) on a cluster computer with 2x Intel(R) Xeon(R) Gold 6130 CPU@2.10 GHz processor and 384 GB of memory. Because the R code comes from different sources and it is not guaranteed that the implementations are computationally optimised, the numbers should be interpreted with caution.

Averages over all simulation runs for each combination of \( N \) and \( p \) are shown in Tables 2–5. The average prediction errors MSPE, angles \( \Theta(\bar{\Pi}, \Pi) \) and ranks of \( \bar{\Pi} \) are illustrated as functions of the true rank \( r \) in Figure 2. The tables and figures contain the lasso penalisation of \( \Pi \) and ranks of \( \bar{\Pi} \); figures contain the lasso penalisation of \( \Pi \) and ranks of \( \bar{\Pi} \); figures contain the lasso penalisation of \( \Pi \) and ranks of \( \bar{\Pi} \); figures contain the lasso penalisation of \( \Pi \) and ranks of \( \bar{\Pi} \); figures contain the lasso penalisation of \( \Pi \) and ranks of \( \bar{\Pi} \).

### Table 4. Simulation results for \( p = 150, N = 150 \) (high dimension, small sample size). The results are averages over 100 simulation runs. See main text for definition of estimation methods and error measures. Bold numbers indicate the best results within a column, after excluding those where true cointegration rank was used.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{\rho}^{(0)} )</th>
<th>RSSE</th>
<th>MSPE</th>
<th>RD</th>
<th>( \Theta )</th>
<th>( d(\hat{\rho}, \rho) )</th>
<th>( p_0 )</th>
<th>TP</th>
<th>TN</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic net</td>
<td>( p )</td>
<td>0.09</td>
<td>38.73</td>
<td>1.07</td>
<td>1.02</td>
<td>1.07</td>
<td>76.62</td>
<td>0.35</td>
<td>0.36</td>
<td>0.66</td>
</tr>
<tr>
<td>El. net + SVD</td>
<td>( r )</td>
<td>0.20</td>
<td>34.78</td>
<td>0.94</td>
<td>1.00</td>
<td>0.94</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>5704.20</td>
</tr>
<tr>
<td>Lasso w. fixed ( r )</td>
<td>( \hat{\rho} )</td>
<td>0.23</td>
<td>31.70</td>
<td>1.11</td>
<td>1.03</td>
<td>1.11</td>
<td>57.68</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on ( \beta )</td>
<td>( \hat{\rho} )</td>
<td>0.21</td>
<td>22.26</td>
<td>0.87</td>
<td>1.00</td>
<td>0.77</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>5218.42</td>
</tr>
<tr>
<td>Lasso on QR</td>
<td>( \hat{\rho} )</td>
<td>0.26</td>
<td>17.43</td>
<td>0.93</td>
<td>1.03</td>
<td>0.82</td>
<td>57.68</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>( p )</td>
<td>1.00</td>
<td>5.39</td>
<td>1.00</td>
<td>1.57</td>
<td>0.00</td>
<td>74.17</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( r )</td>
<td>0.40</td>
<td>23.12</td>
<td>1.16</td>
<td>1.18</td>
<td>0.97</td>
<td>71.32</td>
<td>0.32</td>
<td>0.32</td>
<td>0.68</td>
</tr>
<tr>
<td>( \hat{\rho} )</td>
<td></td>
<td>0.29</td>
<td>33.90</td>
<td>0.95</td>
<td>1.00</td>
<td>0.98</td>
<td>37.16</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### Table 5. Simulation results for \( p = 150, N = 450 \) (high dimension, large sample size). The results are averages over 100 simulation runs. See main text for definition of estimation methods and error measures. Bold numbers indicate the best results within a column, after excluding those where true cointegration rank was used.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{\rho}^{(0)} )</th>
<th>RSSE</th>
<th>MSPE</th>
<th>RD</th>
<th>( \Theta )</th>
<th>( d(\hat{\rho}, \rho) )</th>
<th>( p_0 )</th>
<th>TP</th>
<th>TN</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>( p )</td>
<td>0.13</td>
<td>29.38</td>
<td>0.47</td>
<td>0.41</td>
<td>1.13</td>
<td>75.20</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Elastic net</td>
<td>( p )</td>
<td>0.13</td>
<td>27.33</td>
<td>0.45</td>
<td>0.42</td>
<td>1.10</td>
<td>75.34</td>
<td>0.01</td>
<td>0.01</td>
<td>0.99</td>
</tr>
<tr>
<td>El. net + SVD</td>
<td>( r )</td>
<td>0.21</td>
<td>21.56</td>
<td>0.40</td>
<td>0.38</td>
<td>1.06</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>5376.05</td>
</tr>
<tr>
<td>Lasso w. fixed ( r )</td>
<td>( \hat{\rho} )</td>
<td>0.27</td>
<td>15.81</td>
<td>0.39</td>
<td>0.38</td>
<td>1.04</td>
<td>36.64</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Rank penalty</td>
<td>( p )</td>
<td>0.15</td>
<td>32.66</td>
<td>0.52</td>
<td>0.46</td>
<td>1.16</td>
<td>82.50</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on ( \beta )</td>
<td>( \hat{\rho} )</td>
<td>0.37</td>
<td>25.90</td>
<td>0.65</td>
<td>0.70</td>
<td>0.69</td>
<td>34.24</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Johansen</td>
<td>( r )</td>
<td>0.19</td>
<td>21.64</td>
<td>0.40</td>
<td>0.38</td>
<td>1.09</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>776.65</td>
</tr>
<tr>
<td>Lasso on QR</td>
<td>( \hat{\rho} )</td>
<td>0.23</td>
<td>15.62</td>
<td>0.38</td>
<td>0.37</td>
<td>1.06</td>
<td>37.71</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on ( \alpha )</td>
<td>( \hat{\rho} )</td>
<td>0.31</td>
<td>29.73</td>
<td>0.64</td>
<td>0.69</td>
<td>0.72</td>
<td>33.92</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( \hat{\rho} )</td>
<td>0.19</td>
<td>19.72</td>
<td>0.38</td>
<td>0.36</td>
<td>1.07</td>
<td>35.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>( p )</td>
<td>0.22</td>
<td>14.09</td>
<td>0.36</td>
<td>0.35</td>
<td>1.05</td>
<td>39.56</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( r )</td>
<td>0.84</td>
<td>9.22</td>
<td>0.91</td>
<td>1.35</td>
<td>0.24</td>
<td>71.36</td>
<td>0.79</td>
<td>0.79</td>
<td>0.21</td>
</tr>
<tr>
<td>Nuclear penalty</td>
<td>( \hat{\rho} )</td>
<td>0.55</td>
<td>10.52</td>
<td>0.86</td>
<td>1.04</td>
<td>0.53</td>
<td>33.92</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Adapt. nuclear p.</td>
<td>( p )</td>
<td>0.49</td>
<td>31.61</td>
<td>0.71</td>
<td>0.79</td>
<td>0.76</td>
<td>11.65</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Lasso on SVD</td>
<td>( r )</td>
<td>0.37</td>
<td>44.53</td>
<td>0.69</td>
<td>0.76</td>
<td>0.79</td>
<td>2.14</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \hat{\rho} )</td>
<td></td>
<td>0.40</td>
<td>42.56</td>
<td>0.69</td>
<td>0.76</td>
<td>0.78</td>
<td>34.09</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>
FIGURE 2. Average error measures (angle $\theta(\Pi, \hat{\Pi})$, mean squared prediction error $\text{MSPE}(\hat{\Pi})$ and estimated rank$(\hat{\Pi})$) as functions of the true rank, calculated on 100 simulated data sets. Colored lines represent different methods. Averages are based on the estimates obtained with the estimated rank by $\hat{r}_{RSC}$ for those methods that require a pre-estimate of the rank. The ranks estimated by the Johansen method, elastic net followed by SVD and lasso with a fixed rank are identical to the initial value $\hat{r}^{(0)} = \hat{r}_{RSC}$ and overlap.
for Equation (29). The results may therefore not reflect the theoretical properties of the estimator.

The cointegration rank is estimated meaningfully only by RSC, the method by Pedroni et al. always returned 0 in our simulation study, due to the too low sample size (results not shown). The RSC method underestimates the true rank. Average \( \hat{r}_{RSC} \) can be found in Figure 2 for those methods that require a pre-estimated rank and do not apply any penalisation leading to possible reduction of the rank, for example, the Johansen procedure and elastic net penalty of \( \Pi \) followed by SVD. Most of the methods not requiring a pre-specified rank of \( \Pi \) (rank penalty, nuclear penalty, adaptive nuclear penalty) are not very good in identifying the true rank and give estimates with the full rank or very close to the full rank. It is not surprising, because the tuning parameters in these methods are selected to achieve optimal fit, not an optimal rank. The method with penalty in the QR decomposition provides a more substantial reduction of the rank. The best results in all studied scenarios are achieved with the penalisation of the whole SVD of \( \hat{\Pi} \), when the starting value \( \hat{r}(0) \) is the full rank \( p \).

The impact of the rank misspecification on the estimation of \( \Pi \) is shown in Figure 3, where we display the average MSPE, RD and \( \Theta \) of the methods requiring a prespecified value of \( r \) as a function of the difference \( \text{rank}(\hat{\Pi}) - r \). MSPE and RD are less sensitive to underestimated \( r \) than to overestimated \( r \), where underestimated \( r \) is sometimes associated with even lower values of these measures than the true \( r \). The angle \( \Theta \) is sensitive to underestimated and overestimated \( r \) in the same way. Because the cointegration ranks tend to be underestimated by the RSC method, we conclude that using \( \hat{r}_{RSC} \) instead of the (unknown) true rank \( r \) has only mildly negative or even slightly positive effect on the estimation of \( \Pi \) in our simulation study. However, to make a general conclusion, one needs to estimate \( \hat{\Pi} \) with a range of fixed values of \( r \). In the following,
we comment on the results obtained with $\hat{r}^{(0)} = \hat{r}_{RSC}$ or with $\hat{r}^{(0)} = p$, if the initial estimate is not required.

For small $N$, lower MSPE is usually achieved with methods that lead to a reduced rank of $\Pi$ (Figure 4), such as lasso with a fixed rank, lasso on QR decomposition or lasso imposed on the whole SVD structure, which documents the benefits of a reduced rank of $\Pi$ for model fitting. The lowest MSPE is achieved with nuclear penalty, but note that $\Pi$ is estimated with a rank of 0 with this method. If $p$ and $N$ are large, the most reliable methods are those that estimate $\Pi$ through the product $\hat{\Pi} = \hat{\alpha} \hat{\beta}'$, that is, the Johansen procedure, lasso on $\beta$, elastic net on $\alpha$ and lasso on QR decomposition. Finally, if $p$ is low and $N$ is large, nuclear penalty provides good results. The proximity of $\hat{\Pi}$ to the true $\Pi$ measured by $\text{RD}(\hat{\Pi})$ and by $\Theta(\hat{\Pi}, \Pi)$ reveal a similar ranking of the methods (Figures 5 and 6). If $N$ is low compared with $p$, the lasso with fixed rank and the lasso imposed on SVD give the smallest error on average. The penalisation of $\alpha$ perform best on large samples. Only a few methods are able to identify zeros in $\hat{\Pi}$ (Figure 7). The unrestricted penalisation of $\Pi$ performs best in large samples and penalisation of $\beta$, QR and SVD in small samples.

**FIGURE 4.** Distribution of MSPE($\Pi$) of each estimation method. Boxplots are based on estimates obtained with $\hat{r}^{(0)} = \hat{r}_{RSC}$ or with $\hat{r}^{(0)} = p$ for the methods not requiring a pre-estimate of the rank.
FIGURE 5. Distribution of the relative deviance $RD(\Pi)$ of each estimation method. Boxplots are based on estimates obtained with $\hat{r}(0) = \hat{r}_{RSC}$, or with $\hat{r}(0) = p$ for the methods not requiring a pre-estimate of the rank.

FIGURE 6. Distribution of $\Theta(\Pi)$ of each estimation method. Boxplots are based on estimates obtained with $\hat{r}(0) = \hat{r}_{RSC}$, or with $\hat{r}(0) = p$ for the methods not requiring a pre-estimate of the rank.
The OLS and the Johansen method are always fastest with respect to computing time. The fastest penalisation methods are the rank penalty and the adaptive nuclear penalty, which use similar algorithms based on hard- and soft-thresholding of singular values. The most computationally heavy methods are the penalisation in the QR decomposition, the nuclear penalty and the penalisation of the whole SVD decomposition.

Altogether, the best performing methods appear to be: lasso with fixed rank for small $N$ and low $p$, lasso on SVD for small $N$ and high $p$, and the penalisation of $\alpha$ for large $N$.

6 Discussion

In this paper, we have reviewed regularisation methods for estimation in co-integrated VEC models proposed in the literature, highlighting pros and cons of the different approaches, and evaluating them through a simulation study.

The simulation study confirmed that the OLS method leads to one of the largest prediction errors in all investigated setups, where it is feasible. On the contrary, lasso on SVD and lasso with a fixed rank give the best fit in the most challenging setup with high $p$ and small $N$, even if the rank was underestimated. This reveals that rank reduction, possibly with some additional regularisation, is most beneficial for fitting a cointegrated VEC model and that the reduced rank induced by cointegration should always be taken into account.

The best method for high $p$ and $N > p$ is the penalisation of $\alpha$. The good performance may be due to several reasons. First, the method always starts with a reduced rank, which is a form of pre-regularisation. If the rank is estimated well (such as by the well established RSC method), it has an advantage over the nuclear penalty, lasso on SVD or penalisation of the QR

![Diagram showing distribution of true positives for each estimation method.](https://onlinelibrary.wiley.com/doi/10.1111/insr.12553)
decomposition, because those methods have to identify the rank as part of the procedure often leading to overestimation of the rank. Second, the penalisation of $\alpha$ is most similar to the Johansen procedure and thus inherits some beneficial properties, namely the unbiased estimation of the cointegration space spanned by $\beta$. In contrast, the algorithm used in the penalisation of $\beta$ updates the estimates of $\beta$ and $\alpha$ conditioned on the other in an iterative process, which makes both of them biased. Similarly, the penalisation of the whole SVD structure distorts both the left and the right singular vectors. The penalisation of the QR decomposition also distorts the estimated cointegration space, because $\hat{Q}$, which is the counterpart of $\beta$, does not satisfy the normalisation condition (35). It means that subsets of columns of $\hat{Q}$ span different subspaces than subsets of the normalised eigenvectors in $\hat{\beta}$ estimated with the Johansen procedure. Third, we did not experience any issues with convergence due to the simplicity of the algorithm.

If the aim is to identify the zeros in $\Pi$, most of the methods in this review are not suitable. The only options are to use the standard penalisation of $\Pi$ that does not account for the reduced rank, or the penalisation imposed on QR decomposition or on the whole SVD structure of $\Pi$.

Most methods were originally designed for the reduced rank regression problem, a larger class of models than cointegrated VEC models. Many theoretical properties of the estimators were derived under the assumption that $X$ has finite moments. Unfortunately, the moment conditions are not satisfied if $\{x_n\}$ is an $I(1)$ process. It is not clear how much the violation of the assumption affects the properties of these estimators.

We presented most of the methods with only one tuning parameter $\lambda$. However, it is often straightforward to generalise the methods by introducing multiple tuning parameters, for example, specific values for columns or rows of the matrix under estimation. The algorithms to solve the unrestricted penalisation of $\Pi$, the penalisation of $\alpha$ and the penalisation of $\beta$ contain a step where a penalised multivariate regression model is fitted. This problem can be split into univariate regressions, each with a different tuning parameter. If the variables are on the same scale or have been standardised, it is most natural to use the same tuning parameter. The penalisation of $\beta$ is performed with a transformed response variable $Y\hat{\Sigma}^{-1}$ and the penalisation of $\alpha$ is performed with a transformed predictor matrix $Z\hat{\beta}'$, where the same scale is not guaranteed, so multiple tuning parameters can be reasonable.

We opted for one tuning parameter in the simulation study to make the comparison of the methods as fair as possible. The only method where we used multiple tuning parameters is the penalisation of the SVD. Note that this method is usually one of the worst performing methods despite multiple tuning parameters.

The presented methods can be extended to fit a VAR model of higher order than 1 in terms of the number of lags in the model. The additional lags are always accompanied with full-rank matrices $\Gamma_1, \ldots, \Gamma_k$, so a possible strategy could be to penalise them with one of the common types of penalties, such as lasso (used in Wilms & Croux, 2016), weighted lasso (used in Liang & Schienle, 2019; Chen & Schienle, 2022) or elastic net. However, a VAR model that has order of integration larger than 1 requires more complex treatment.

Finally, we hope this review is useful for deciding which penalisation method to use in a specific application according to the purpose of the analysis.

Funding Information

EU Horizon 2020 research and innovation programme, Marie Sklodowska-Curie Action, Grant Number: 887784; Novo Nordisk Foundation, Grant Number: NNF20OC0062958; Independent Research Fund Denmark, Grant Number: 9040-00215B.
Supporting Information

The estimation methods described in the paper are implemented in an R package penCA, which is available at a GitHub repository https://github.com/MarieLevakova/penCA. All files documenting the simulation study are available in an archive https://doi.org/10.17894/ucph.cb5ee5dd-b83c-4136-a5b1-fb8018c2cba6.

References


International Statistical Review (2023)


[Received August 2022; accepted August 2023]