Transition Matrix Estimation in High Dimensional Time Series

Fang Han FHAN@JHSPH.EDU
Johns Hopkins University, 615 N.Wolfe Street, Baltimore, MD 21205 USA

Han Liu HANLIU@PRINCETON.EDU
Princeton University, 98 Charlton Street, Princeton, NJ 08544 USA

Abstract

In this paper, we propose a new method in estimating transition matrices of high dimensional vector autoregressive (VAR) models. Here the data are assumed to come from a stationary Gaussian VAR time series. By formulating the problem as a linear program, we provide a new approach to conduct inference on such models. In theory, under a doubly asymptotic framework in which both the sample size T and dimensionality d of the time series can increase (with possibly $d \gg T$), we provide explicit rates of convergence between the estimator and the population transition matrix under different matrix norms. Our results show that the spectral norm of the transition matrix plays a pivotal role in determining the final rates of convergence. This is the first work analyzing the estimation of transition matrices under a high dimensional doubly asymptotic framework. Experiments are conducted on both synthetic and real-world stock data to demonstrate the effectiveness of the proposed method compared with the existing methods. The results of this paper have broad impact on different applications, including finance, genomics, and brain imaging.

1. Introduction

Vector autoregressive (VAR) models are an important class of models for analyzing multivariate time series data and have been used heavily in a number of domains such as finance, genomics and brain imaging data analysis (Tsay, 2005; Ledoit & Wolf, 2003; Bar-Joseph, 2004; Lozano et al., 2009; Andersson et al., 2001).

This paper aims at estimating the transition matrices of high dimensional stationary vector autoregressive (VAR) time series. In detail, let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_T$ be a time series. We assume that for $t = 1, \ldots, T$, $\boldsymbol{X}_t \in \mathbb{R}^d$ is a d-dimensional random vector and we have

$$\boldsymbol{X}_{t+1} = \mathbf{A}^\mathsf{T} \boldsymbol{X}_t + \boldsymbol{Z}_t, \quad \text{for } t = 1, 2, \ldots, T-1. \quad (1.1)$$

Here $\{\boldsymbol{X}_t\}_{t=1}^T$ is a stationary process with $\boldsymbol{X}_t \sim N_d(\mathbf{0}, \mathbf{\Sigma})$ for $t = 1, \ldots, T$, $\mathbf{A} \in \mathbb{R}^{d \times d}$ is called the *transition matrix*, and $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_T \sim^{\text{i.i.d.}} N_d(\mathbf{0}, \mathbf{\Psi})$ are independent multivariate Gaussian white noise with covariance matrix $\mathbf{\Psi}$. It is easy to observe that, in order to preserve the stationary property, we need to have $\mathbf{\Sigma} = \mathbf{A}^\mathsf{T} \mathbf{\Sigma} \mathbf{A} + \mathbf{\Psi}$.

Given the time series data $\{\boldsymbol{X}_t\}_{t=1}^T$, a common method for estimating $\mathbf{A}$ is the least-square estimator (see, for example, Hamilton (1994)). The estimator can be expressed as the optimum to the following optimization problem:

$$\widehat{\mathbf{A}}^{\text{LSE}} = \underset{\mathbf{M} \in \mathbb{R}^{d \times d}}{\operatorname{argmin}} ||\mathbf{Y} - \mathbf{M}^\mathsf{T} \mathbf{X}||_\mathsf{F}^2, \quad (1.2)$$

where $\mathbf{Y} := [\boldsymbol{X}_2, \boldsymbol{X}_3, \ldots, \boldsymbol{X}_T]$, $\mathbf{X} := [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{T-1}]$ and $||\cdot||_\mathsf{F}$ represents the matrix Frobenius norm. Different penalty terms, such as the ridge penalty $||\mathbf{M}||_F^2$, can be further posed on $\mathbf{M}$ in Equation (1.2). A more recent work proposed by Huang & Schneider (2011) utilizes a new penalty term called Lyapunov penalty and they accordingly obtain an estimator, which they claim to have good empirical performance. However, these estimators are no longer consistent in high dimensional settings when the dimensionality d is greater than the sample size T. Moreover, the optimization formulation in Huang & Schneider (2011) is not convex, making the estimator hard to compute and analyze. More recently, Wang et al. (2007); Hsu et al. (2008) propose to add sparsity penalty $\sum_{ij} |\mathbf{M}_{ij}|$ to Equation (1.2) to handle the

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

high dimensional case. The corresponding estimators' theoretical performance is further analyzed in Wang et al. (2007); Nardi & Rinaldo (2011) under the assumption that $\mathbf{A}$ is sparse, i.e., the number of nonzero entries in $\mathbf{A}$ is much less than the dimensionality d^2. However, they only consider fixed d setting in their analysis.

In this paper, we propose a new approach to estimate the transition matrix $\mathbf{A}$, where $\mathbf{A}$ can be both sparse and nonsparse matrices and without restricted to a specific sparsity pattern. We also consider recovering the support set of $\mathbf{A}$. We introduce a new method by directly estimating $\mathbf{A}$ utilizing the relationship between $\mathbf{A}$ and the marginal and lag one covariance matrices. The new method has several advantages. Firstly, the method is computational attractive because we can formulate the problem as d linear program and solve it either in sequence or in parallel. Similar convex formulations have been used in learning high dimensional graphical models (Candes & Tao, 2007; Yuan, 2010; Cai et al., 2011; Liu et al., 2012). Secondly, we can provide theoretical analysis for the propose method. Let a matrix be called s-sparse if there are at most s nonzero elements on each row. We show that if $\mathbf{A}$ is s-sparse, then under some mild conditions, the error between our estimator $\widehat{\mathbf{A}}$ and $\mathbf{A}$ satisfies that,

$$||\widehat{\mathbf{A}} - \mathbf{A}||_1 = O_P\left(s \cdot \frac{||\mathbf{A}||_1}{1 - ||\mathbf{A}||_2}\sqrt{\frac{\log d}{T}}\right),$$

$$\text{and } ||\widehat{\mathbf{A}} - \mathbf{A}||_{\max} = O_P\left(\frac{||\mathbf{A}||_1}{1 - ||\mathbf{A}||_2}\sqrt{\frac{\log d}{T}}\right).$$

Here for any square matrix $\mathbf{M}$, $||\mathbf{M}||_{\max}$ and $||\mathbf{M}||_1$ represent the matrix elementwise absolute maximum norm ($\ell_{\max}$ norm) and induced ℓ_1 norm (detailed definitions will be provided later). Utilizing the $\ell_{\max}$ convergence result, the estimators' performance in support recovery can also be established.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the background of this paper, especially the VAR time series model. In Section 3, we introduce the proposed method on inferring the VAR model. We prove the main theoretical results in Section 4. In Section 5, we apply the obtained new method on both synthetic and real-world stock data to illustrate its effectiveness. The conclusion is provided in the last section.

2. Background

In this section, we briefly introduce the background of this paper. We start with some notation: Let $\mathbf{M} = [M_{jk}] \in \mathbb{R}^{d\times d}$ and $\boldsymbol{v} = (v_1, ..., v_d)^T \in \mathbb{R}^d$. Let $\boldsymbol{v}$'s subvector with entries indexed by J be denoted by $\boldsymbol{v}_J$. Let $\mathbf{M}$'s submatrix with rows indexed by J and columns indexed by K be denoted by $\mathbf{M}_{JK}$. Let $\mathbf{M}_{J*}$ and $\mathbf{M}_{*K}$ be the submatrix of $\mathbf{M}$ with rows in J, and the submatrix of $\mathbf{M}$ with columns in K. For $0 < q < \infty$, we define the ℓ_0,ℓ_q, and ℓ_∞ vector norms as

$$||\boldsymbol{v}||_0 := \sum_j I(v_j \neq 0), \quad ||\boldsymbol{v}||_q := \left(\sum_{i=1}^d |v_i|^q\right)^{1/q}$$
$$\text{and } \quad ||\boldsymbol{v}||_\infty := \max_{1\le i\le d} |v_i|,$$

where $I(\cdot)$ is the indicator function. We use the following notation for matrix $\ell_q, \ell_{\max}$ and ℓ_{F} norms:

$$\|\mathbf{M}\|_q := \max_{\|v\|_q=1} \|\mathbf{M}\boldsymbol{v}\|_q, \|\mathbf{M}\|_{\max} := \max_{jk} |\mathbf{M}_{jk}|,$$
$$\text{and } \|\mathbf{M}\|_{\mathsf{F}} := \big(\sum_{j,k} |\mathbf{M}_{jk}|^2\big)^{1/2}.$$

Let $\Lambda_j(\mathbf{M})$ be the j-th largest eigenvalue of $\mathbf{M}$. In particular, $\Lambda_{\min}(\mathbf{M}) := \Lambda_d(\mathbf{M})$ and $\Lambda_{\max}(\mathbf{M}) := \Lambda_1(\mathbf{M})$ are the smallest and largest eigenvalues of $\mathbf{M}$. Let $\mathbf{1}_d = (1, \ldots, 1)^{\mathsf{T}} \in \mathbb{R}^d$.

The stationary Vector Autoregressive (VAR) time series model linear dependence between different movements. In particular, we assume that the T observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T$ can be modeled by a lag one autoregressive process:

$$\boldsymbol{X}_{t+1} = \mathbf{A}^{\mathsf{T}}\boldsymbol{X}_t + \boldsymbol{Z}_t, \quad \text{for } t = 1, 2, \ldots, T-1. \quad (2.1)$$

To secure the stationary of the above process, the transition matrix $\mathbf{A}$ must have bounded spectral norm $||\mathbf{A}||_2 < 1$. We assume the Gaussian colored noise $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots \sim^{i.i.d.} N_d(\mathbf{0}, \boldsymbol{\Psi})$. $\boldsymbol{Z}_t$ and $\boldsymbol{X}_t$ are independent. We have the following proposition, which states that $\mathbf{A}$ is relevant to the marginal and lag one covariance matrices of $\{\boldsymbol{X}_t\}_{t=1}^T$. This motivates the proposed method provided in the next section.

Proposition 2.1. *With the above notation, suppose that the VAR model in Equation* (2.1) *holds. For $i \geq 1$, let $\boldsymbol{\Sigma}_i := \mathrm{Cov}(\boldsymbol{X}_1, \boldsymbol{X}_{1+i})$. We have for any $1 \leq t \leq T - i$, $\boldsymbol{\Sigma}_i = \mathrm{Cov}(\boldsymbol{X}_t, \boldsymbol{X}_{t+i}) = (\mathbf{A}^{\mathsf{T}})^i\boldsymbol{\Sigma}$. Moreover,*

$$\mathbf{A} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\Sigma}_1)^{\mathsf{T}}. \quad (2.2)$$

Proof. Using Equation (2.1), for any t, we have

$$\mathrm{Cov}(\boldsymbol{X}_t, \boldsymbol{X}_{t+i}) = \mathrm{Cov}(\boldsymbol{X}_t, (\mathbf{A}^{\mathsf{T}})^i\boldsymbol{X}_t) = (\mathbf{A}^{\mathsf{T}})^i\boldsymbol{\Sigma},$$

which is not relevant to t. Moreover, $\boldsymbol{\Sigma}_1 = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}$, implying that $\mathbf{A} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\Sigma}_1)^{\mathsf{T}}$. □

VAR model is widely used in the analysis of economic time series (Hatemi-J, 2004; Briiggemann & Liitkepohl, 2001), signal processing (de Waele & Broersen, 2003) and brain fMRI (Goebel et al., 2003; Roebroeck et al., 2005).

3. Methods and Algorithms

In this section, we provide an new optimization formulation of the proposed method to achieve the final estimator. We then provide the detailed algorithm to calculate this estimator.

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_T$ be a sequence satisfying the VAR model described in Equation (2.1). Let $\mathbf{S}$ and $\mathbf{S}_1$ be the marginal and lag one sample covariance matrices of $\{\boldsymbol{X}_t\}_{t=1}^T$:

$$\mathbf{S} := \frac{1}{T}\sum_{t=1}^{\mathsf{T}} \boldsymbol{X}_t \boldsymbol{X}_t^{\mathsf{T}} \quad \text{and} \quad \mathbf{S}_1 := \frac{1}{T-1}\sum_{t=1}^{T-1} \boldsymbol{X}_t \boldsymbol{X}_{t+1}^{\mathsf{T}}.$$

Using Proposition 2.1, we propose to estimate $\mathbf{A}$ by plugging the marginal and lag one sample covariance matrices $\mathbf{S}$ and $\mathbf{S}_1$ into the following convex optimization problem:

$$\widehat{\mathbf{A}} = \underset{\mathbf{M}\in\mathbb{R}^{d\times d}}{\operatorname{argmin}} \sum_{jk} |\mathbf{M}_{jk}|$$
$$\text{subject to} \quad ||\mathbf{S}\mathbf{M} - \mathbf{S}_1^{\mathsf{T}}||_{\max} \le \lambda_0, \tag{3.1}$$

where $\lambda_0 > 0$ is a tuning parameter. When a suitable sparsity assumption on the transition matrix $\mathbf{A}$ is added, we will see that $\widehat{\mathbf{A}}$ is a consistent estimator of $\mathbf{A}$. It can be further shown that this is equivalent to calculate $\mathbf{A}$ in column by column. In detail, letting $\widehat{\boldsymbol{\beta}}_j$ be the solution to the following optimization problem:

$$\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{v}\in\mathbb{R}^d}{\operatorname{argmin}} ||\boldsymbol{v}||_1$$
$$\text{subject to} \quad ||\mathbf{S}\boldsymbol{v} - [\mathbf{S}_1^{\mathsf{T}}]_{*j}||_\infty \le \lambda_0, \tag{3.2}$$

we have $\widehat{\mathbf{A}}_{*j} := \widehat{\boldsymbol{\beta}}_j$ for $j = 1, \ldots, d$.

Recall that any real number a takes the decomposition $a = a^+ - a^-$, where $a^+ = a \cdot I(a \ge 0)$ and $a^- = -a \cdot I(a < 0)$. For any vector $\boldsymbol{v} = (v_1, \ldots, v_d)^{\mathsf{T}} \in \mathbb{R}^d$, let $\boldsymbol{v}^+ := (v_1^+, \ldots, v_d^+)^{\mathsf{T}}$ and $\boldsymbol{v}^- := (v_1^-, \ldots, v_d^-)^{\mathsf{T}}$. We say that $\boldsymbol{v} \ge 0$ if $v_1, \ldots, v_d \ge 0$ and $\boldsymbol{v} < 0$ if $v_1, \ldots, v_d < 0$. We say that $\boldsymbol{v}_1 \ge \boldsymbol{v}_2$ if $\boldsymbol{v}_1 - \boldsymbol{v}_2 \ge 0$, and $\boldsymbol{v}_1 - \boldsymbol{v}_2 < 0$ if $\boldsymbol{v}_1 - \boldsymbol{v}_2 < 0$. Letting $\boldsymbol{v} = (v_1, \ldots, v_d)^{\mathsf{T}}$, Equation (3.2) can be further relaxed to the following problem:

$$\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{v}^+,\boldsymbol{v}^-}{\operatorname{argmin}} \mathbf{1}_d^{\mathsf{T}}(\boldsymbol{v}^+ + \boldsymbol{v}^-)$$
$$\text{subject to } ||\mathbf{S}\boldsymbol{v}^+ - \mathbf{S}\boldsymbol{v}^- - [\mathbf{S}_1^{\mathsf{T}}]_{*j}||_\infty \le \lambda_0,$$
$$\text{and } \boldsymbol{v}^+ \ge 0, \boldsymbol{v}^- \ge 0. \tag{3.3}$$

Equation (3.4) can be written as

$$\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{v}^+,\boldsymbol{v}^-}{\operatorname{argmin}} \mathbf{1}_d^{\mathsf{T}}(\boldsymbol{v}^+ + \boldsymbol{v}^-)$$
$$\text{subject to } \mathbf{S}\boldsymbol{v}^+ - \mathbf{S}\boldsymbol{v}^- - [\mathbf{S}_1^{\mathsf{T}}]_{*j} \le \lambda_0 \mathbf{1}_d$$
$$-\mathbf{S}\boldsymbol{v}^+ + \mathbf{S}\boldsymbol{v}^- + [\mathbf{S}_1^{\mathsf{T}}]_{*j} \le \lambda_0 \mathbf{1}_d$$
$$\text{and } \boldsymbol{v}^+ \ge 0, \boldsymbol{v}^- \ge 0. \tag{3.4}$$

This is equivalent to

$$\widehat{\boldsymbol{\beta}}_j = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \mathbf{1}_{2d}^{\mathsf{T}}\boldsymbol{\omega}$$
$$\text{subject to } \boldsymbol{\theta} + \mathbf{W}\boldsymbol{\omega} \ge 0, \text{and } \boldsymbol{\omega} \ge 0, \tag{3.5}$$

where

$$\boldsymbol{\omega} = \begin{pmatrix} \boldsymbol{v}^+ \\ \boldsymbol{v}^- \end{pmatrix}, \boldsymbol{\theta} = \begin{pmatrix} [\mathbf{S}_1^{\mathsf{T}}]_{*j} + \lambda_0 \mathbf{1}_d \\ -[\mathbf{S}_1^{\mathsf{T}}]_{*j} + \lambda_0 \mathbf{1}_d \end{pmatrix},$$
$$\text{and } \mathbf{W} = \begin{bmatrix} -\mathbf{S} & \mathbf{S} \\ \mathbf{S} & -\mathbf{S} \end{bmatrix}. \tag{3.6}$$

Equation (3.5) is a linear programming problem. In this paper, we use the simplex algorithm to compute $\widehat{\mathbf{A}}$.

4. Theoretical Properties

In this section we analyze the theoretical properties of the proposed method. We provide the nonasymptotic upper bound of the rate of convergence in parameter estimation under the matrix ℓ_1 and $\ell_{\max}$ norms. To our knowledge, this is the first work analyzing the estimation of transition matrix under a high dimensional doubly asymptotic framework. In the sequel, we assume that $d > T$.

4.1. Main Result

The main result states that under the VAR model, the estimator $\widehat{\mathbf{A}}$ obtained by Equation (3.1) can approximate $\mathbf{A}$ consistently even when d grows exponentially fast with respect to T, and the upper bound of the convergence is also related to the sparsity level of the transition matrix $\mathbf{A}$.

We start with some additional notation. Let M_d be a quantity which may scale with the dimensionality d. For any matrix $\mathbf{M} = [M_{ij}] \in \mathbb{R}^{d\times d}$, we define

$$\mathcal{M}(q, s, M_d) := \left\{ \mathbf{M} : \max_{1\le i\le d} \sum_{j=1}^{d} |M_{ij}|^q \le s, ||\mathbf{M}||_1 \le M_d \right\}.$$

For $q = 0$, the class $\mathcal{M}(0, s, M_d)$ contains all the s-sparse matrices as defined in Section 1.

Theorem 4.1. *Suppose that* $\{\boldsymbol{X}_t\}_{t=1}^T$ *follows the VAR model described in Equation* (2.1) *with transition matrix* $\mathbf{A}$. *Suppose that* $\mathbf{A} \in \mathcal{M}(q, s, M_d)$ *for some* $0 \leq q < 1$. *Let* $\widehat{\mathbf{A}}$ *be the optimum to Equation* (3.1) *with tuning parameter*

$$\lambda_0 = \frac{16||\mathbf{\Sigma}||_2 \max_j(\mathbf{\Sigma}_{jj})}{\min_j(\mathbf{\Sigma}_{jj})(1-||\mathbf{A}||_2)} \cdot (2M_d+5)\sqrt{\frac{\log d}{T}}.$$

Then when $T \geq 6\log d$ *and* $d \geq 8$, *with probability larger than* $1-14d^{-1}$,

$$||\widehat{\mathbf{A}} - \mathbf{A}||_1 \leq$$

$$4s\left(\frac{32||\mathbf{\Sigma}^{-1}||_1 \max_j(\mathbf{\Sigma}_{jj})||\mathbf{\Sigma}||_2}{\min_j(\mathbf{\Sigma}_{jj})(1-||\mathbf{A}||_2)} \cdot (2M_d+5)\sqrt{\frac{\log d}{T}}\right)^{1-q}.$$

Moreover, with probability larger than $1-14d^{-1}$,

$$||\widehat{\mathbf{A}} - \mathbf{A}||_{\max} \leq \frac{32||\mathbf{\Sigma}^{-1}||_1 \max_j(\mathbf{\Sigma}_{jj})||\mathbf{\Sigma}||_2}{\min_j(\mathbf{\Sigma}_{jj})(1-||\mathbf{A}||_2)} \cdot (2M_d+5)\sqrt{\frac{\log d}{T}}.$$

Remark 4.2. *The bound obtained in Theorem 4.1 depends on both* $\mathbf{\Sigma}$ *and* $\mathbf{A}$. *Here* $\mathbf{A}$ *characterized the data dependence degree. When both* $||\mathbf{\Sigma}^{-1}||_1$ *and* $||\mathbf{\Sigma}||_2$ *do not scale with* (n, d, s), *the rate can be further simplified as:*

$$||\widehat{\mathbf{A}} - \mathbf{A}||_1 = O_P\left(s \cdot \frac{M_d}{1-||\mathbf{A}||_2}\sqrt{\frac{\log d}{T}}\right),$$

$$\text{and} \quad ||\widehat{\mathbf{A}} - \mathbf{A}||_{\max} = O_P\left(\frac{M_d}{1-||\mathbf{A}||_2}\sqrt{\frac{\log d}{T}}\right).$$

Moreover, if $\mathbf{A} \in \mathcal{M}(0, s, M_d)$, utilizing the result on elementwise $\ell_{\max}$ norm convergence, a support recovery result can be easily derived. In detail, let $\widetilde{\mathbf{A}}$ be the truncated version of $\widehat{\mathbf{A}}$ with level γ, i.e.,

$$\widetilde{\mathbf{A}}_{ij} = \widehat{\mathbf{A}}_{ij} I(|\widehat{\mathbf{A}}_{ij}| \geq \gamma).$$

We then have the following corollary, claiming that we can recover the support set of $\mathbf{A}$ with large probability.

Corollary 4.3. *Suppose that the assumptions in Theorem 4.1 hold and* $\mathbf{A} \in \mathcal{M}(0, s, M_d)$. *Then choose the truncation level*

$$\gamma = \frac{32||\mathbf{\Sigma}^{-1}||_1 \max_j(\mathbf{\Sigma}_{jj})||\mathbf{\Sigma}||_2}{\min_j(\mathbf{\Sigma}_{jj})(1-||\mathbf{A}||_2)} \cdot (2M_d+5)\sqrt{\frac{\log d}{T}}.$$

Provided that $\min_{j,k}|\mathbf{A}_{jk}| \geq 2\gamma$, *with probability larger than* $1-14d^{-1}$, *we have* $\text{sign}(\mathbf{A}) = \text{sign}(\widetilde{\mathbf{A}})$. *Here for any matrix* $\mathbf{M}$, $\text{sign}(\mathbf{M})$ *determines the sign of each entry in* $\mathbf{M}$.

Detailed proofs of Theorem 4.1 and Corollary 4.3 can be found in the long version of this paper (Han & Liu, 2013).

5. Experiments

In this section we show the numerical results on both synthetic and real-world data to illustrate the effectiveness and empirical usefulness of the proposed method compared with the existing methods. In detail, the following three methods are considered:

Ridge: the method implemented by adding a ridge penalty $||\mathbf{M}||_F^2$ on Equation (1.2);

Lasso: the method implemented by adding a sparsity penalty $\sum_{ij}|\mathbf{M}_{ij}|$ on Equation (1.2);

LP: the proposed method by formulating the problem as a linear program (see Equation (3.1)).

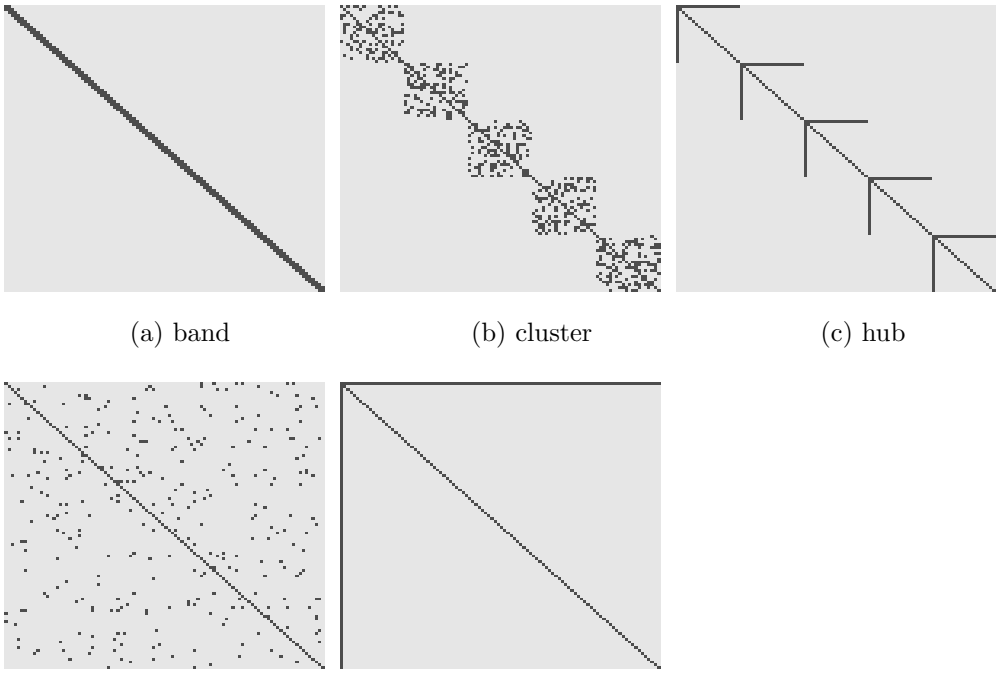
We use package *glmnet* in computing Lasso and the simplex algorithm in computing LP.

5.1. Synthetic Data

In this section we show the effectiveness of LP compared with Ridge and Lasso on several synthetic data. In our numerical simulations, we consider the setting where the sample size T varies from 50 to 800 and the dimensionality d varies from 50 to 200. Each data are distributed according to a VAR model described in Equation (2.1). To generate such data, we first determine the sparse matrix $\mathbf{A}$ according to several patterns. We adopt the following five models for $\mathbf{A}$: *band, cluster, hub, random* and *scale-free*. Typical patterns of the generated $\mathbf{A}$ are illustrated in Figure 1. Here the white points represent the zero entries and the black points represent the nonzero entries. Note that in each pattern, $\mathbf{A}$ is not symmetric.

We then rescale $\mathbf{A}$ such that the operator norm of $\mathbf{A}$ is set to be $||\mathbf{A}||_2 = \alpha$, where α is set to be 0.5. Given $\mathbf{A}$, $\mathbf{\Sigma}$ is generated such that the operator norm of $\mathbf{\Sigma}$ is $||\mathbf{\Sigma}||_2 = 2||\mathbf{A}||_2$. According to the stationary property, the covariance matrix of the noise vector $\boldsymbol{Z}_t$ is $\mathbf{\Psi} = \mathbf{\Sigma} - \mathbf{A}^\intercal\mathbf{\Sigma}\mathbf{A}$, where $\mathbf{\Psi}$ must be a positive definite matrix. Using the generating model in Equation (2.1), we can then obtain a sequence $\mathbf{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T]^\intercal \in \mathbb{R}^{T\times d}$. We repeat this procedure for 1,000 times in each setting. We apply the three methods on each dataset $\mathbf{X}$, the averaged distance between the estimators and the true transition matrix with respect to the matrix Frobenius, operator and ℓ_1 norms are illustrated in Tables 1 to 5, with standard deviations provided in the brackets. Here the tuning parameter is selected by cross validation.

There are several conclusions which can be drawn from the results shown in Tables 1 to 5: (i) It can be observed that, across all these settings, LP outperforms Ridge and Lasso significantly in terms of the ℓ_1 norm,



(a) band (b) cluster (c) hub

(d) random (e) scale-free

Figure 1. Four different transition matrix patterns. Here white points represent the zero entries and black points represent nonzero entries. Here $d = 100$.

indicating that LP is more effect in estimating the transition matrix than the existing methods; (ii) When the sample size increases, the error in estimating the transition matrix increases for Lasso and LP, coinciding with the theoretical result in the last section. On the other hand, Ridge has a very poor performance in each setting, indicating that it can not handle the high dimensional data.

5.2. Equity Data

We compare different methods on the stock price data from Yahoo! Finance (`finance.yahoo.com`). We collect the daily closing prices for 452 stocks that are consistently in the S&P 500 index between January 1, 2003 through January 1, 2008. This gives us altogether 1,257 data points, each data point corresponding to the vector of closing prices on a trading day. Let $St = [St_{t,j}]$ with $St_{t,j}$ denoting the closing price of stock j on day t. Let $\mathbf{X}$ be the standardized version of St. We model the variables $\mathbf{X}_{t,j}$ using the VAR model shown in Equation (2.1) and estimate the transition matrix $\mathbf{A}$. For any estimator $\widehat{\mathbf{A}}_s$ with the number of nonzero entries to be s, we calculate the prediction error as:

$$\epsilon_s = \frac{1}{T-1}\sum_{t=2}^{T} ||\mathbf{X}_{t*} - \widehat{\mathbf{A}}_s^{\top}\mathbf{X}_{(t-1)*}||_2.$$

We plot (s, ϵ_s) for methods Lasso and LP in Figure 6. It can be observed that LP constantly outperform Lasso with respect to any given sparsity level s. This illustrates the empirical usefulness of the proposed method compared with the existing methods.

6. Conclusion

In this paper a new method in estimating the transition matrix under the stationary vector autoregressive model is proposed. The contribution of the paper includes: (i) With respect to methodology, we propose a new method utilizing the power of linear programming; (ii) With respect to theory, under the doubly asymptotic framework, the nonasymptotic bound of convergence with respect to different matrix norms are explicitly provided; (iii) With respect to empirical usefulness, numerical experiments on both synthetic and real-world stock data are conducted, demonstrating the effectiveness of the proposed method compared with the existing methods. To our knowledge, this is the first work analyzing the performance of estimators on inferring the VAR model in high dimensional set-

Table 1. Comparison of averaged matrix losses for three methods over 1,000 replications. Here **A**'s pattern is "band", ℓ_F, ℓ_2 and ℓ_1 represent the Frobenius, induced ℓ_2 and ℓ_1 matrix norms respectively.

		Ridge			Lasso			LP		
d	n	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1
50	50	11.8(0.33)	1.17(0.04)	12.54(1.27)	3.53(0.09)	0.66(0.03)	2.46(0.24)	2.21(0.06)	0.50(0.01)	0.86(0.09)
	100	6.96(0.15)	0.91(0.04)	7.65(0.40)	2.34(0.06)	0.50(0.02)	1.48(0.13)	2.09(0.03)	0.49(0.00)	0.57(0.03)
	200	4.03(0.04)	0.60(0.03)	4.16(0.17)	1.56(0.05)	0.39(0.03)	0.82(0.07)	2.13(0.06)	0.49(0.01)	0.50(0.01)
	400	2.59(0.05)	0.41(0.02)	2.67(0.20)	1.25(0.05)	0.34(0.02)	0.49(0.04)	1.34(0.18)	0.36(0.04)	0.45(0.03)
	800	1.74(0.03)	0.26(0.01)	1.78(0.08)	0.98(0.10)	0.29(0.03)	0.36(0.03)	0.98(0.09)	0.27(0.03)	0.37(0.03)
100	50	9.83(0.17)	1.24(0.02)	10.53(0.85)	5.52(0.07)	0.75(0.03)	3.06(0.19)	3.27(0.04)	0.53(0.02)	1.06(0.27)
	100	20.3(0.29)	1.23(0.03)	21.87(1.18)	3.96(0.06)	0.58(0.02)	2.18(0.17)	3.00(0.03)	0.49(0.00)	0.62(0.05)
	200	10.01(0.09)	0.92(0.03)	9.96(0.38)	2.49(0.06)	0.43(0.02)	1.12(0.07)	3.01(0.05)	0.49(0.00)	0.50(0.01)
	400	5.71(0.05)	0.62(0.03)	5.67(0.18)	1.85(0.04)	0.37(0.02)	0.58(0.05)	2.18(0.51)	0.42(0.05)	0.50(0.01)
	800	3.68(0.02)	0.42(0.02)	3.58(0.11)	1.61(0.07)	0.31(0.02)	0.39(0.03)	1.59(0.10)	0.31(0.02)	0.40(0.01)
200	50	9.08(0.04)	1.23(0.02)	8.70(0.35)	7.97(0.06)	0.79(0.02)	3.34(0.16)	4.88(0.08)	0.55(0.02)	1.31(0.34)
	100	14.31(0.11)	1.26(0.01)	14.17(0.39)	6.38(0.05)	0.65(0.02)	2.75(0.18)	4.26(0.03)	0.50(0.00)	0.69(0.05)
	200	34.65(0.32)	1.27(0.03)	36.13(1.35)	4.12(0.05)	0.47(0.01)	1.59(0.12)	4.28(0.04)	0.49(0.00)	0.52(0.01)
	400	14.22(0.08)	0.93(0.02)	13.92(0.33)	2.72(0.04)	0.37(0.01)	0.70(0.07)	4.37(0.56)	0.49(0.04)	0.50(0.00)
	800	8.11(0.04)	0.63(0.02)	7.60(0.13)	2.35(0.08)	0.33(0.01)	0.43(0.02)	2.41(0.12)	0.34(0.02)	0.42(0.01)

Table 2. Comparison of averaged matrix losses for three methods over 1,000 replications. Here **A**'s pattern is "cluster", ℓ_F, ℓ_2 and ℓ_1 represent the Frobenius, induced ℓ_2 and ℓ_1 matrix norms respectively.

		Ridge			Lasso			LP		
d	n	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1
50	50	12.07(0.38)	1.09(0.06)	13.38(0.88)	3.34(0.09)	0.57(0.03)	2.61(0.24)	1.60(0.04)	0.48(0.02)	0.84(0.07)
	100	7.00(0.19)	0.83(0.04)	7.66(0.52)	2.14(0.06)	0.44(0.04)	1.50(0.17)	1.48(0.02)	0.49(0.01)	0.68(0.02)
	200	4.07(0.07)	0.57(0.03)	4.21(0.23)	1.39(0.04)	0.40(0.02)	0.88(0.07)	1.43(0.08)	0.47(0.02)	0.65(0.02)
	400	2.64(0.03)	0.39(0.02)	2.67(0.09)	1.11(0.03)	0.39(0.02)	0.60(0.03)	1.20(0.10)	0.42(0.03)	0.58(0.02)
	800	1.79(0.03)	0.26(0.01)	1.84(0.08)	0.91(0.06)	0.35(0.03)	0.50(0.03)	0.92(0.05)	0.35(0.02)	0.51(0.03)
100	50	9.61(0.14)	1.13(0.02)	10.36(0.92)	5.22(0.07)	0.65(0.03)	3.12(0.20)	2.29(0.12)	0.48(0.02)	1.00(0.15)
	100	20.8(0.32)	1.12(0.02)	22.29(0.96)	3.55(0.06)	0.49(0.02)	2.12(0.13)	1.97(0.02)	0.49(0.01)	0.70(0.03)
	200	9.93(0.11)	0.84(0.03)	10.12(0.36)	2.20(0.04)	0.40(0.03)	1.17(0.06)	1.92(0.05)	0.48(0.02)	0.68(0.02)
	400	5.74(0.05)	0.58(0.02)	5.67(0.16)	1.56(0.03)	0.40(0.02)	0.69(0.05)	1.63(0.12)	0.43(0.03)	0.64(0.04)
	800	3.74(0.03)	0.38(0.02)	3.56(0.10)	1.37(0.07)	0.38(0.02)	0.59(0.04)	1.30(0.07)	0.36(0.02)	0.58(0.04)
200	50	8.44(0.04)	1.14(0.02)	8.69(0.35)	7.37(0.08)	0.70(0.02)	3.30(0.18)	3.48(0.26)	0.49(0.01)	1.23(0.21)
	100	13.98(0.13)	1.16(0.01)	14.53(0.58)	5.78(0.05)	0.57(0.02)	2.82(0.17)	2.81(0.01)	0.49(0.01)	0.75(0.05)
	200	35.69(0.39)	1.16(0.02)	36.68(1.51)	3.66(0.03)	0.44(0.02)	1.63(0.07)	2.80(0.06)	0.49(0.01)	0.70(0.02)
	400	14.1(0.08)	0.84(0.02)	13.63(0.39)	2.34(0.03)	0.41(0.01)	0.82(0.03)	2.54(0.15)	0.47(0.02)	0.64(0.03)
	800	8.13(0.03)	0.58(0.03)	7.55(0.16)	2.00(0.05)	0.40(0.01)	0.60(0.03)	1.99(0.08)	0.40(0.02)	0.60(0.03)

Table 3. Comparison of averaged matrix losses for three methods over 1,000 replications. Here **A**'s pattern is "hub", ℓ_F, ℓ_2 and ℓ_1 represent the Frobenius, induced ℓ_2 and ℓ_1 matrix norms respectively.

		Ridge			Lasso			LP		
d	n	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1
50	50	12.08(0.38)	1.04(0.02)	13.37(1.24)	3.21(0.09)	0.53(0.02)	2.40(0.23)	1.25(0.08)	0.49(0.07)	1.18(0.11)
	100	6.94(0.15)	0.78(0.04)	7.51(0.42)	1.93(0.07)	0.39(0.05)	1.47(0.13)	1.13(0.15)	0.38(0.06)	1.04(0.07)
	200	4.04(0.05)	0.55(0.03)	4.21(0.14)	1.17(0.05)	0.32(0.05)	1.03(0.07)	1.05(0.11)	0.32(0.05)	0.92(0.09)
	400	2.64(0.05)	0.39(0.02)	2.71(0.22)	0.91(0.07)	0.28(0.03)	0.86(0.05)	0.89(0.06)	0.28(0.04)	0.86(0.07)
	800	1.79(0.03)	0.26(0.01)	1.85(0.10)	0.77(0.06)	0.20(0.03)	0.74(0.06)	0.74(0.06)	0.21(0.02)	0.74(0.06)
100	50	9.59(0.16)	1.10(0.01)	10.47(0.52)	5.06(0.09)	0.64(0.03)	2.96(0.15)	1.90(0.05)	0.50(0.01)	1.40(0.12)
	100	21.02(0.31)	1.08(0.02)	22.46(1.39)	3.37(0.06)	0.45(0.03)	2.15(0.15)	1.53(0.07)	0.50(0.05)	1.23(0.05)
	200	9.88(0.11)	0.80(0.03)	10.12(0.33)	1.92(0.04)	0.35(0.04)	1.35(0.09)	1.39(0.05)	0.39(0.04)	1.09(0.07)
	400	5.75(0.05)	0.56(0.02)	5.65(0.18)	1.26(0.03)	0.32(0.02)	1.03(0.05)	1.25(0.05)	0.32(0.03)	1.01(0.05)
	800	3.75(0.02)	0.38(0.02)	3.69(0.11)	1.06(0.04)	0.27(0.02)	0.91(0.03)	1.05(0.05)	0.26(0.02)	0.91(0.04)
200	50	8.28(0.05)	1.10(0.01)	8.66(0.35)	7.23(0.07)	0.68(0.03)	3.29(0.15)	2.95(0.17)	0.50(0.01)	1.55(0.14)
	100	13.90(0.10)	1.12(0.01)	14.22(0.48)	5.49(0.05)	0.52(0.02)	2.69(0.18)	2.10(0.04)	0.50(0.00)	1.24(0.03)
	200	35.83(0.27)	1.11(0.02)	37.13(0.84)	3.31(0.03)	0.37(0.02)	1.64(0.11)	1.98(0.05)	0.43(0.03)	1.14(0.04)
	400	14.07(0.09)	0.81(0.01)	13.60(0.48)	1.93(0.03)	0.33(0.02)	1.11(0.07)	1.82(0.03)	0.35(0.03)	1.04(0.05)
	800	8.15(0.04)	0.56(0.02)	7.69(0.16)	1.56(0.02)	0.31(0.02)	0.97(0.03)	1.57(0.05)	0.29(0.02)	0.97(0.03)

Table 4. Comparison of averaged matrix losses for three methods over 1,000 replications. Here **A**'s pattern is "random", ℓ_F, ℓ_2 and ℓ_1 represent the Frobenius, induced ℓ_2 and ℓ_1 matrix norms respectively.

		Ridge			Lasso			LP		
d	n	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1
50	50	12.10(0.32)	1.11(0.05)	13.35(0.71)	3.36(0.09)	0.60(0.04)	2.48(0.23)	1.72(0.03)	0.47(0.02)	0.88(0.10)
	100	6.95(0.15)	0.87(0.05)	7.31(0.51)	2.18(0.06)	0.44(0.03)	1.46(0.13)	1.63(0.04)	0.48(0.02)	0.67(0.05)
	200	4.02(0.06)	0.59(0.03)	4.13(0.20)	1.40(0.05)	0.37(0.03)	0.84(0.08)	1.40(0.14)	0.43(0.04)	0.64(0.03)
	400	2.62(0.04)	0.39(0.03)	2.67(0.11)	1.09(0.05)	0.34(0.02)	0.54(0.05)	1.12(0.06)	0.35(0.04)	0.55(0.03)
	800	1.79(0.02)	0.27(0.02)	1.78(0.07)	0.87(0.05)	0.27(0.03)	0.47(0.03)	0.89(0.05)	0.29(0.03)	0.46(0.04)
100	50	9.73(0.16)	1.17(0.01)	10.90(0.50)	5.18(0.08)	0.67(0.03)	3.05(0.19)	2.43(0.06)	0.48(0.02)	0.96(0.11)
	100	20.92(0.42)	1.14(0.02)	21.91(1.01)	3.65(0.05)	0.50(0.01)	2.14(0.14)	2.16(0.03)	0.49(0.02)	0.74(0.04)
	200	9.96(0.09)	0.85(0.03)	10.24(0.46)	2.23(0.04)	0.39(0.03)	1.12(0.09)	1.97(0.12)	0.44(0.03)	0.68(0.04)
	400	5.75(0.05)	0.58(0.03)	5.65(0.19)	1.55(0.05)	0.35(0.02)	0.65(0.05)	1.61(0.12)	0.37(0.03)	0.60(0.03)
	800	3.73(0.03)	0.40(0.02)	3.60(0.08)	1.27(0.06)	0.31(0.02)	0.53(0.03)	1.22(0.07)	0.30(0.03)	0.53(0.04)
200	50	8.43(0.04)	1.15(0.01)	8.82(0.32)	7.38(0.07)	0.71(0.02)	3.39(0.20)	3.48(0.04)	0.49(0.03)	1.18(0.13)
	100	13.99(0.11)	1.16(0.01)	14.31(0.40)	5.75(0.05)	0.54(0.01)	2.69(0.16)	2.78(0.04)	0.47(0.03)	0.89(0.08)
	200	35.39(0.25)	1.15(0.02)	36.35(1.00)	3.60(0.04)	0.39(0.02)	1.53(0.07)	2.63(0.07)	0.41(0.03)	0.75(0.08)
	400	14.13(0.09)	0.85(0.02)	13.60(0.37)	2.23(0.04)	0.33(0.02)	0.79(0.06)	2.28(0.08)	0.36(0.02)	0.69(0.07)
	800	8.14(0.03)	0.58(0.02)	7.68(0.13)	1.73(0.06)	0.29(0.02)	0.59(0.04)	1.73(0.12)	0.29(0.02)	0.62(0.04)

Table 5. Comparison of averaged matrix losses for three methods over 1,000 replications. Here **A**'s pattern is "scale-free", ℓ_F, ℓ_2 and ℓ_1 represent the Frobenius, induced ℓ_2 and ℓ_1 matrix norms respectively.

		Ridge			Lasso			LP		
d	n	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1	ℓ_F	ℓ_2	ℓ_1
50	50	12.07(0.4)	1.06(0.02)	13.29(1.06)	3.24(0.09)	0.54(0.03)	2.41(0.25)	1.31(0.14)	0.41(0.07)	0.96(0.13)
	100	6.90(0.13)	0.80(0.03)	7.44(0.42)	1.99(0.07)	0.39(0.05)	1.43(0.14)	1.16(0.08)	0.37(0.09)	0.89(0.16)
	200	4.06(0.05)	0.56(0.03)	4.19(0.18)	1.19(0.05)	0.29(0.04)	0.88(0.10)	1.06(0.05)	0.33(0.04)	0.77(0.08)
	400	2.64(0.05)	0.39(0.02)	2.71(0.21)	0.91(0.07)	0.26(0.04)	0.74(0.09)	0.91(0.05)	0.26(0.04)	0.71(0.07)
	800	1.79(0.03)	0.26(0.02)	1.83(0.10)	0.73(0.05)	0.19(0.02)	0.60(0.07)	0.71(0.05)	0.19(0.02)	0.58(0.06)
100	50	9.59(0.16)	1.10(0.01)	10.64(0.7)	5.05(0.08)	0.61(0.03)	2.97(0.16)	1.78(0.16)	0.39(0.08)	1.15(0.23)
	100	21.09(0.34)	1.06(0.02)	22.41(1.23)	3.36(0.06)	0.41(0.02)	2.10(0.17)	1.42(0.16)	0.40(0.08)	1.04(0.13)
	200	9.90(0.10)	0.79(0.02)	9.90(0.34)	1.89(0.05)	0.28(0.03)	1.10(0.08)	1.31(0.14)	0.34(0.04)	0.94(0.08)
	400	5.77(0.05)	0.56(0.02)	5.63(0.13)	1.25(0.11)	0.29(0.03)	0.89(0.07)	1.20(0.06)	0.31(0.03)	0.91(0.06)
	800	3.76(0.02)	0.38(0.02)	3.66(0.09)	0.99(0.05)	0.25(0.03)	0.83(0.07)	1.02(0.11)	0.23(0.03)	0.79(0.06)
200	50	8.17(0.05)	1.08(0.01)	8.64(0.32)	7.16(0.07)	0.67(0.02)	3.24(0.25)	2.64(0.07)	0.45(0.08)	1.42(0.16)
	100	13.89(0.11)	1.10(0.01)	14.19(0.42)	5.41(0.05)	0.48(0.02)	2.64(0.18)	1.71(0.15)	0.36(0.07)	1.18(0.11)
	200	35.96(0.33)	1.08(0.03)	37.59(1.25)	3.19(0.03)	0.31(0.03)	1.54(0.10)	1.69(0.16)	0.33(0.05)	1.07(0.10)
	400	14.06(0.08)	0.79(0.02)	13.62(0.5)	1.74(0.03)	0.28(0.03)	1.03(0.08)	1.58(0.07)	0.30(0.03)	1.02(0.06)
	800	8.17(0.04)	0.55(0.02)	7.73(0.17)	1.38(0.03)	0.27(0.02)	0.99(0.04)	1.36(0.10)	0.25(0.02)	0.95(0.04)

tings. The proof techniques we used have the separate interest for analyzing a variety of other multivariate statistical methods on time series data. The results of this paper have broad impact on different application, including finance, genomics and brain imaging.

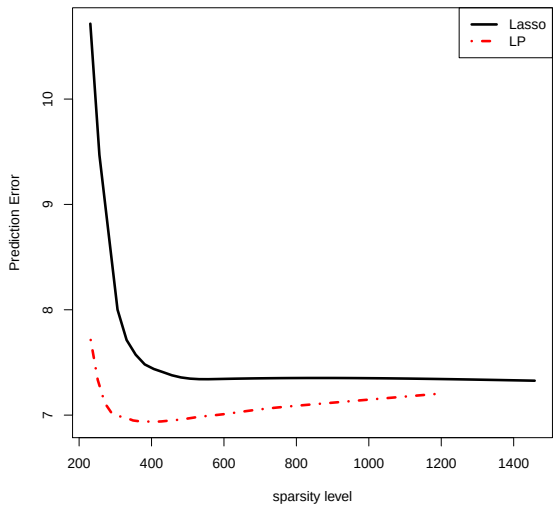


Figure 2. The figure illustrating the prediction error versus the sparsity level of the transition matrix. Here the x-lab represents the number of nonzero entries in the estimated matrix, y-lab represents the averaged prediction error.

References

Andersson, J.L.R., Hutton, C., Ashburner, J., Turner, R., and Friston, K. Modeling geometric deformations in epi time series. *Neuroimage*, 13(5):903–919, 2001.

Bar-Joseph, Z. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–2503, 2004.

Briiggemann, R. and Liitkepohl, H. Lag selection in subset var models with an application to a us monetary system. *Econometric Studies: A Festschrift in Honour of Joachim Frohn, LIT-Verlag, Miinster*, pp. 107–28, 2001.

Cai, T., Liu, W., and Luo, X. A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

Candes, E. and Tao, T. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

de Waele, S. and Broersen, P.M.T. Order selection for vector autoregressive models. *Signal Processing, IEEE Transactions on*, 51(2):427–433, 2003.

Goebel, R., Roebroeck, A., Kim, D.S., and Formisano, E. Investigating directed cortical interactions in time-resolved fmri data using vector autoregressive modeling and granger causality mapping. *Magnetic resonance imaging*, 21(10):1251–1261, 2003.

Hamilton, J.D. *Time series analysis*, volume 2. Cambridge Univ Press, 1994.

Han, F. and Liu, H. Transition matrix estimation in high dimensional time series data. *Technical Report*, 2013.

Hatemi-J, A. Multivariate tests for autocorrelation in the stable and unstable var models. *Economic Modelling*, 21(4):661–683, 2004.

Hsu, N.J., Hung, H.L., and Chang, Y.M. Subset selection for vector autoregressive processes using lasso. *Computational Statistics & Data Analysis*, 52(7): 3645–3657, 2008.

Huang, T.K. and Schneider, J. Learning autoregressive models from sequence and non-sequence data. *Advances in Neural Information Processing Systems*, 25, 2011.

Ledoit, O. and Wolf, M. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. High dimensional semiparametric gaussian copula graphical models. *Annals of Statistics*, 2012.

Lozano, A.C., Abe, N., Liu, Y., and Rosset, S. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–i118, 2009.

Nardi, Y. and Rinaldo, A. Autoregressive process modeling via the lasso procedure. *Journal of Multivariate Analysis*, 102(3):528–549, 2011.

Roebroeck, A., Formisano, E., Goebel, R., et al. Mapping directed influence over the brain using granger causality and fmri. *Neuroimage*, 25(1):230–242, 2005.

Tsay, R.S. *Analysis of financial time series*, volume 543. Wiley-Interscience, 2005.

Wang, H., Li, G., and Tsai, C.L. Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1): 63–78, 2007.

Yuan, M. High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, 99:2261–2286, 2010.