Learning Low-Complexity Autoregressive Models via Proximal Alternating Minimization

Fu Lin and Jie Chen

***Abstract*—We consider the estimation of the state transition matrix in vector autoregressive models, when time sequence data is limited but nonsequence steady-state data is abundant. To leverage both sources of data, we formulate the least squares minimization problem regularized by a Lyapunov penalty. We impose cardinality or rank constraints to reduce the complexity of the autoregressive model. We solve the resulting nonconvex, nonsmooth problem by using the proximal alternating linearization method (PALM). We show that PALM is globally convergent to a critical point and that the estimation error monotonically decreases. Furthermore, we obtain explicit formulas for the proximal operators to facilitate the implementation of PALM. We demonstrate the effectiveness of the developed method on synthetic and real-world data. Our experiments show that PALM outperforms the gradient projection method in both computational efficiency and solution quality.**

Keywords: Autoregressive models, Lyapunov penalty, nonconvex nonsmooth problem, steady-state data, proximal alternating linearized minimization.

I. Introduction

Vector autoregressive (VAR) models are widely used in the analysis of linear interdependence in time series data. A key step in building the VAR model is the identification of the state transition matrix. When time sequence data is adequate, the standard approach is to solve a least-squares problem. In modern applications, however, the dimension of the model is significantly larger than the number of time sequence measurements, which makes the model unidentifiable through the standard least-squares approach. Such scenarios include, for example, tracking the progression of brain neurological diseases, because the number of comprehensive brain scans is limited due to cost or medical concerns [1]. In gene expression networks, the number of genes is typically much larger than the number of measurements, because of the intrusive nature of the measuring techniques [2], [3], [4].

In such situations, regularization is a typical rescue. For example, ridge regularization is a common approach for ensuring a unique solution. Other regularization approaches introduce additional structures to the solution. In particular, sparsity and low-rank structures are extensively studied. These regularization approaches are popular, in part because the resulting problem may be efficiently solved by using convex optimization techniques [5], [6], [7], [8], [3], [1], [9], [10], [11]. In [5], a sparse VAR model is found via Lasso for gene regulatory networks. In [10], the state transition matrix is decomposed into a sparse matrix and a low-rank matrix by using convex penalty functions. Other approaches based on convex optimization can be found in [6], [7], [8], [3], [1], [9], [11].

In a different vein, steady-state data provide opportunity for improving model accuracy. When the VAR model is stable and steady-state data are abundant, several authors show that the steady-state data can help reduce the estimation error [8], [12], [1], [3], [4], [13]. In [1], steady-state data are leveraged to form the Lyapunov regularization. In [3], the perturbed steady-state data is used to infer sparse, stable gene expression networks. In [4], both steady-state and temporal data are integrated in the estimation of the gene regulatory networks. Other work that employs steady-state data for system identification includes [8], [12], [13].

In this paper, we leverage both time sequence and steady-state nonsequence data for the model estimation. We propose a least-squares estimator regularized by the Lyapunov penalty subject to the cardinality or rank constraints on the state transition matrix. The identification problem is nonconvex due to the Lyapunov penalty and nonsmooth due to the low-complexity constraints. We solve the problem by using the proximal alternating linearization method (PALM). An advantage of PALM is that it converges to a critical point starting from any initial condition. We prove this global convergence property of PALM and show that the estimation error is monotonically decreasing with the PALM iterations. We obtain closed-form expressions for the proximal operators to facilitate implementation. We show that PALM can handle the stability constraints and also the convex low-complexity (e.g., the ℓ_1 or the nuclear-norm) constraints. We demonstrate that our approach outperforms the gradient projection method in both computational time and solution quality.

Our presentation is organized as follows. In Section II, we formulate the estimation problem for the low-complexity VAR model. In Section III, we present the PALM algorithm and derive explicit formulas for the proximal operators. In Section IV, we show the global convergence of PALM by establishing the Lipschitz conditions and the KL property of the estimation problem. In Section V, we demonstrate the effectiveness of PALM via numerical experiments. In Section VI, we summarize our contributions and discuss future directions.

II. Model Identification via Lyapunov Penalty

In this section, we formulate the model identification problem using both time-sequence data and steady-state data. The performance of the model is measured by the least-squares error for the time-sequence data and the Lyapunov penalty

F. Lin is with the Systems Department, United Technologies Research Center, 411 Silver Ln, East Hartford, CT 06118, USA. E-mail: linf@utrc.utc.com

J. Chen is with IBM Thomas J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA. E-mail: chenjie@us.ibm.com

for the steady-state data. We employ low-complexity penalty functions to promote sparsity and low-rank properties of the state transition matrix.

Consider a p-dimensional vector autoregressive model:

$$\phi(t+1) \;=\; A\phi(t) \;+\; \epsilon(t), \tag{1}$$

where $\phi(t) \in \mathbb{R}^p$ is the state vector, $A \in \mathbb{R}^{p\times p}$ is the state transition matrix, and $\epsilon(t) \in \mathbb{R}^p$ is a zero-mean white stochastic process. We assume that the autoregressive model (1) is asymptotically stable; that is, all eigenvalues of A have modulus less than one. The state vector $\phi(t)$ has a steady-state distribution, whose covariance matrix P is determined by the discrete-time Lyapunov equation

$$APA^T \;+\; Q \;=\; P,$$

where $Q \in \mathbb{R}^{p\times p}$ is the covariance matrix of $\epsilon(t)$. Linear systems theory says that P is positive definite if and only if A is asymptotically stable [14].

Our objective is to identify the state transition matrix A. Given a set of n time sequence measurements of $\phi(t)$, the standard least-squares estimation is given by

$$\underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}}\;\; \frac{1}{2}\|X\Phi \;-\; \Psi\|_F^2, \tag{2}$$

where $\Phi \;:=\; [\,\phi(1),\cdots,\phi(n\;-\;1)\,] \;\in\; \mathbb{R}^{p\times(n-1)}$, $\Psi \;:=\; [\,\phi(2),\cdots,\phi(n)\,] \in \mathbb{R}^{p\times(n-1)}$, and $\|\cdot\|_F$ denotes the Frobenius norm. We use X to denote the unknown state transition matrix for the convenience of developing optimization details. When the number of time sequence data is less than the dimension of the states (i.e., $p > n-1$), infinitely many solutions exist for (2) and the state transition matrix is unidentifiable.

We are interested in the scenario when the time sequence data is scarce but the steady-state nonsequence data is readily available [8], [12], [1], [7], [15]. In this case, Huang and Schneider [1] propose the Lyapunov penalty as a regularization term

$$\|XPX^T \;+\; Q \;-\; P\|_F^2. \tag{3}$$

They show that the Lyapunov penalty helps improve the accuracy of the estimation [1]. Since the covariance matrix P is unknown, we replace it by the sample covariance

$$S \;:=\; \frac{1}{N}\sum_{i=1}^{N}(z^i-\bar{z})(z^i-\bar{z})^T \text{ with } \bar{z} \;:=\; \frac{1}{N}\sum_{i=1}^{N} z^i,$$

where $\{z^i\}_{i=1}^N$ is the steady-state nonsequence data. The identification problem with the Lyapunov regularization can be expressed as

$$\underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}}\;\; \frac{1}{2}\|X\Phi \;-\; \Psi\|_F^2 \;+\; \frac{\rho}{2}\|XSX^T \;+\; Q \;-\; S\|_F^2, \tag{4}$$

where ρ is a positive coefficient that balances the estimation error between the sequence and the nonsequence data.

Huang and Schneider study (4) and show that the Lyapunov penalty improves the solution quality. However, there is no guarantee that the solution of (4) is stable (i.e., spectral radius of X is less than 1). We next incorporate stability constraint into (4).

A. Stability Constraint

Since stability is a necessary condition for the use of Lyapunov penalty (3), we impose a stability constraint in the identification problem (4). Let $\tau(X)$ denote the spectral radius of X, that is, $\tau(X) := \max\{|\lambda_i|\}_{i=1}^p$. A stable autoregressive model can be obtained by solving the following problem:

$$\begin{array}{ll}\underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\|X\Phi \;-\; \Psi\|_F^2 \;+\; \frac{\rho}{2}\|XSX^T \;+\; Q \;-\; S\|_F^2 \\ \text{subject to} & \tau(X) \;<\; 1.\end{array} \tag{5}$$

Dealing with τ directly is difficult because spectral radius is neither convex nor locally Lipschitz [16], [17]. Alternatively, one can employ a convex function as an upper bound [18]. Since $\tau(X) \le \|X\|_2 \le \|X\|_F$ (see [19, Chapter 5]), we can incorporate the stability constraint in the cost function

$$\underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}}\;\; \frac{1}{2}\|X\Phi-\Psi\|_F^2+\frac{\rho}{2}\|XSX^T+Q-S\|_F^2+\frac{\mu}{2}\|XX^T\|_F^2 \tag{6}$$

where μ is a positive constant. While one can employ the spectral norm $\|X\|_2$ as a less conservative proxy, we choose the Frobenius norm mainly because both the Lyapunov penalty (3) and the stability penalty $\|XX^T\|_F^2$ are then quadratic functions of X in Frobenius norm squared. Hence, the stability term $\|XX^T\|_F^2$ is inconsequential in the design of solution methods. For this reason and for the ease of presentation, in what follows we omit the stability penalty, but comment on the modification of the algorithm when appropriate to address stability. Detailed analysis of spectral radius and its relaxation in minimization problem can be found in [16], [20], [18], [17].

B. Low-Complexity Models

In several applications, it is desired to impose sparsity or low-rank structures on the state transition matrix [5], [6], [7], [8], [3], [1], [9], [10], [11]. In gene expression networks, for example, the nonzero elements of the state transition matrix determine the interaction graph of the expression network [5], [3]. A sparse state transition matrix is useful because one can construct a sparse network to explain experiment data.

One common approach to promoting sparsity is to impose the ℓ_1 constraint:

$$\|X\|_{\ell_1} := \sum_{i,j=1}^{p} |X_{ij}| \;\le\; l, \tag{7}$$

where l is a prescribed positive number. Since the ℓ_1 norm promotes sparsity *implicitly*, the actual number of nonzero elements in the solution is indirectly controlled by the threshold l. However, given a desired level of sparsity, the correct choice of l is typically unknown a priori. An *explicit* way to guarantee sparsity is to control the number of nonzero elements by the cardinality constraint:

$$\mathbf{card}(X) \;:=\; \text{number of nonzero entries of } X \;\le\; s, \tag{8}$$

where s is a given positive integer. Note that the cardinality constraint is harder to deal with than the ℓ_1 constraint, because cardinality is a nonconvex function.

Another approach to obtaining low-complexity models is to impose the low-rank constraint. A low-rank state transition

matrix is useful because it implies that the data can be explained by a model with lower dimensions. An implicit way to promote low-rank solutions is to use the nuclear norm constraint [21], [22], [23], [24]

$$\|X\|_* := \sum_{i=1}^{p} \sigma_i(X) \leq \nu, \tag{9}$$

where ν is a prescribed positive number and the σ_is are the singular values. Similar to the sparsity case, the threshold ν is not known a priori. We impose a low-rank constraint by controlling the rank of the state transition matrix:

$$\mathbf{rank}(X) := \text{ number of nonzero singular values of } X \leq r, \tag{10}$$

where r is a given positive integer.

Hence we consider the following estimation problem:

$$\widehat{A} = \underset{X \in \mathbb{R}^{p\times p}}{\operatorname{argmin}} \ \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{\rho}{2}\|XSX^T + Q - S\|_F^2$$
$$\text{subject to constraint (7) or (8) or (9) or (10).} \tag{11}$$

For the convex constraints (7) and (9), one may employ gradient projection methods; namely, taking a descent direction of the objective function and projecting it onto the convex constraint sets. A gradient projection method is proposed in [1] to solve (11) with the ℓ_1 constraint (7). For the nonconvex constraints (8) and (10), on the other hand, we develop the PALM algorithm in the subsequent section.

III. Proximal Alternating Linearized Method

In this section, we develop the PALM algorithm for the identification problem of low-complexity models. This approach decomposes the problem into a sequence of smaller problems that can be solved efficiently. Furthermore, we show that PALM is global convergence to a critical point for both convex and nonconvex constraints in (11).

We begin with a reformulation of the low-complexity autoregressive models (11)

$$\begin{array}{ll} \underset{X,Y\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{\rho}{2}\|YSX^T + Q - S\|_F^2 \\ \text{subject to} & Y - X = 0, \\ & \text{(7) or (8) or (9) or (10)} \end{array}$$

where we replace one of the two Xs in the Lyapunov penalty by a new variable Y. Let f denote the least-squares term

$$f(X) = \frac{1}{2}\|X\Phi - \Psi\|_F^2, \tag{12}$$

and let g denote the indicator function of the individual constraints in (7)-(10), for example,

$$g(Y) = \begin{cases} 0, & \mathbf{card}(Y) \leq s \\ \infty, & \text{otherwise} \end{cases} \tag{13}$$

for the cardinality constraint (8) and

$$g(Y) = \begin{cases} 0, & \mathbf{rank}(Y) \leq r \\ \infty, & \text{otherwise} \end{cases} \tag{14}$$

for the rank constraint (10). Then we have

$$\underset{X,Y\in\mathbb{R}^{p\times p}}{\text{minimize}} \ \omega(X,Y) := f(X) + g(Y) + h(X,Y), \tag{15}$$

where h denotes the coupling term

$$h(X,Y) = \frac{\rho_1}{2}\|YSX^T + Q - S\|_F^2 + \frac{\rho_2}{2}\|X - Y\|_F^2. \tag{16}$$

Here, the penalty parameter $\rho_1 > 0$ resumes the role of ρ in (11) and $\rho_2 > 0$ is sufficiently large to penalize the discrepancy between X and Y. It is worth mentioning that the convergence of PALM does not depend on the choice of ρ_1 and ρ_2. This is in contrast to ADMM that may require sufficiently large quadratic term to ensure convergence when it is applied to nonconvex problems [25], [26].

A. Generic PALM Method

PALM computes the proximal operators of the *uncoupled* functions f and g, around the linearization of the *coupling* function h at the previous iterate, hence the name [27], [28], [29], [30]. It is instructive to put PALM in the context of other alternating methods. Suppose for the moment that $\omega(X,Y)$ is a strictly convex function. One approach to minimizing ω is the Gauss-Seidel iteration (also known as the coordinate descent):

$$\begin{aligned} X^{k+1} &\in \underset{X}{\operatorname{argmin}} \ \omega(X, Y^k) \\ Y^{k+1} &\in \underset{Y}{\operatorname{argmin}} \ \omega(X^{k+1}, Y). \end{aligned}$$

Convergence of the iteration requires a unique solution in each minimization step; otherwise, Gauss-Seidel may cycle indefinitely [31]. When ω is convex but *not strictly* convex, uniqueness can be achieved by including a quadratic proximal term

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{\omega(X, Y^k) + \frac{c_k}{2}\|X - X^k\|_F^2\right\} \tag{18a}$$

$$Y^{k+1} \in \underset{Y}{\operatorname{argmin}} \left\{\omega(X^{k+1}, Y) + \frac{d_k}{2}\|Y - Y^k\|_F^2\right\}, \tag{18b}$$

where c_k and d_k are positive coefficients. This class of proximal methods is well studied; see [29] for a recent survey.

When ω is nonconvex, as in our case (15), we need to modify the proximal terms to ensure convergence. Instead of taking the proximal term around X^k as in (18a), we take the term around X^k modified with a scaled partial gradient of h:

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{f(X) + \frac{c_k}{2}\|X - U^k\|_F^2\right\}, \tag{19}$$

where $U^k = X^k - \frac{1}{c_k}\nabla_X h(X^k, Y^k)$. The parameter c_k is chosen to be greater than the Lipschitz constant of $\nabla_X h$; in particular, $c_k = \gamma_1 L_1(Y^k)$ for some $\gamma_1 > 1$ where L_1 is the Lipschitz constant of $\nabla_X h$.

Similarly, we take the proximal term around Y^k modified with a scaled partial gradient of h:

$$Y^{k+1} \in \underset{Y}{\operatorname{argmin}} \left\{g(Y) + \frac{d_k}{2}\|Y - V^k\|_F^2\right\}, \tag{20}$$

where $V^k = Y^k - \frac{1}{d_k}\nabla_Y h(X^{k+1}, Y^k)$. The parameter d_k is determined by $d_k = \gamma_2 L_2(X^{k+1})$ for some $\gamma_2 > 1$ where L_2

is the Lipschitz constant of $\nabla_Y h$. PALM alternates between updating (X, Y) by using the iterations (19)-(20).

B. Formulas for Lipschitz Constants and Solutions to (19)-(20)

To implement (19)-(20), one needs the Lipschitz constants L_1 and L_2 in order to determine the coefficients c_k and d_k, respectively. Taking the partial gradients of h yields

$$\begin{aligned}\nabla_X h &= \rho_1(XS^TY^TYS + (Q - S)^TYS) + \rho_2(X - Y)\\ \nabla_Y h &= \rho_1(YSX^TXS^T + (Q - S)XS^T) + \rho_2(Y - X).\end{aligned}$$

Since $\nabla_X h$ is linear in X and $\nabla_Y h$ is linear in Y, we obtain explicit formulas for the Lipschitz constants

$$\begin{aligned}L_1(Y) &= \|\rho_1 S^TY^TYS + \rho_2 I\|_2\\ L_2(X) &= \|\rho_1 SX^TXS^T + \rho_2 I\|_2\end{aligned} \tag{21}$$

where $\|\cdot\|_2$ denotes the largest singular value of a matrix.

We next show that the proximal operators (19)-(20) can be computed efficiently. The proximal operator (19) can be expressed as

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{ \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{c_k}{2}\|X - U^k\|_F^2 \right\}.$$

Solving this least-squares problem yields

$$X^{k+1} = (\Psi\Phi^T + c_k U^k)(\Phi\Phi^T + c_k I)^{-1},$$

where I denotes the identity matrix. When the number of states is no less than the number of time sequence data (i.e., $p \geq n$), one can reduce the computational cost by inverting $\Phi^T\Phi + c_k I$ instead of $\Phi\Phi^T + c_k I$, since the Woodbury formula gives

$$X^{k+1} = (c_k^{-1}\Psi\Phi^T + U^k)(I - \Phi(c_k I + \Phi^T\Phi)^{-1}\Phi^T).$$

The proximal operator (20) can be expressed as

$$\begin{aligned}\underset{Y}{\text{minimize}} \quad & \frac{d_k}{2}\|Y - V^k\|_F^2\\ \text{subject to} \quad & \text{(7) or (8) or (9) or (10).}\end{aligned}$$

For the cardinality constraint (8), the solution is obtained by keeping the s largest elements of V^k in magnitude and zero out the rest of the elements in V^k. This is because the squared Frobenius norm is the sum of the squared elements of $Y - V^k$. For the rank constraint (10), by the Eckart–Young theorem, the solution is the best rank-r approximation of V^k obtained by the truncated SVD; that is, keeping the r-largest singular value and setting the remaining singular values of V^k to zero.

For the ℓ_1 constraint (7), the projection onto the ℓ_1-ball can be computed by an algorithm developed in [32]. For the nuclear-norm constraint (9), the optimal solution Y can be computed by performing the singular value decomposition of V^k and then projecting the singular values of V^k onto the ℓ_1-ball.

We summarize the computational steps in Algorithm 1, focusing on only the constraints (8) and (10).

We conclude this section with a remark on stability.

Remark 1 (Stability)**.** *As discussed in Section II-A, we can incorporate the stability constraint by penalizing $\|XX^T\|_F^2$ in the cost function. In this case, the coupling term becomes*

$$\begin{aligned}h(X,Y) &= \frac{\rho_1}{2}\|YSX^T + Q - S\|_F^2 + \frac{\rho_2}{2}\|X - Y\|_F^2\\ &\quad + \frac{\mu}{2}\|YX^T\|_F^2.\end{aligned}$$

Its partial gradients are given by

$$\begin{aligned}\nabla_X h &= \rho_1(XS^TY^TYS + (Q - S)^TYS) + \rho_2(X - Y)\\ &\quad + \mu XY^TY\\ \nabla_Y h &= \rho_1(YSX^TXS^T + (Q - S)XS^T) + \rho_2(Y - X)\\ &\quad + \mu YX^TX,\end{aligned}$$

whose Lipschitz constants are given by

$$\begin{aligned}L_1(Y) &= \|\rho_1 S^TY^TYS + \mu Y^TY + \rho_2 I\|_2\\ L_2(X) &= \|\rho_1 SX^TXS^T + \mu X^TX + \rho_2 I\|_2.\end{aligned}$$

Therefore, Algorithm 1 applies by modifying the computation of the Lipschitz constants.

Remark 2 (Comparison with ADMM)**.** *The alternating direction method of multipliers (ADMM) has been a very powerful tool in distributed control and optimization [33], [25], [26], [34]. Since ADMM is a class of proximal algorithms [29], it is closely related to PALM. It is worth mentioning that ADMM is most useful for minimizing the sum of convex functions. For certain classes of nonconvex problems, the convergence of ADMM has been established in [25], [26], [34]. For the cardinality* (8) *and the rank function* (10)*, ADMM may not converge for* (11)*. The solution to which ADMM converges may also depend on the value of ρ; see [25]. Furthermore, efficient methods for subproblems in ADMM that deal with the Lyapunov penalty are yet to be developed.*

IV. Convergence Analysis

In this section, we show that Algorithm 1 globally converges to a critical point of the nonconvex, nonsmooth problem (15). Furthermore, the objective value is monotonically decreasing throughout the PALM iterations. We build upon the seminal work on the convergence of PALM for generic problems [30]. Our contributions are the establishments of the required Lipschitz conditions and the KL property.

We begin with a technical lemma on the Lipschitz conditions of the objective function ω.

Lemma 1. *The objective function ω in* (15) *satisfies the following properties:*

1) $\inf_{X,Y} \omega(X,Y) > -\infty$, $\inf_X f(X) > -\infty$, *and* $\inf_Y g(Y) > -\infty$.
2) *For a fixed Y, the partial gradient $\nabla_X h(X,Y)$ is globally Lipschitz; that is, there exists $L_1(Y)$ such that $\|\nabla_X h(X_1,Y) - \nabla_X h(X_2,Y)\|_F \leq L_1(Y)\|X_1 - X_2\|_F$ for all X_1 and X_2. Likewise, for a fixed X, the partial gradient $\nabla_Y h(X,Y)$ is globally Lipschitz; that is, there exists $L_2(X)$ such that $\|\nabla_Y h(X,Y_1) - \nabla_Y h(X,Y_2)\|_F \leq L_2(X)\|Y_1 - Y_2\|_F$ for all Y_1 and Y_2.*

Algorithm 1 Proximal Alternating Linearization Method for (15)

Initialization: Start with any (X^0, Y^0).
for $k = 0, 1, 2, \ldots$ until convergence **do**
▷ The following section computes X^{k+1}
Compute the Lipschitz constant $L_1(Y^k) = \|\rho_1 S^T Y^{kT} Y^k S + \rho_2 I\|_2$.
Compute $c_k = \gamma_1 L_1(Y^k)$ for some $\gamma_1 > 1$.
Compute the partial gradient
$\nabla_X h(X^k, Y^k) = \rho_1(X^k S^T Y^{kT} Y^k S + (Q - S)^T Y^k S) + \rho_2(X^k - Y^k)$.
Update the proximal point $U^k = X^k - \frac{1}{c_k}\nabla_X h(X^k, Y^k)$.
if $p < n$ **then**
$X^{k+1} = (\Psi\Phi^T + c_k U^k)(\Phi\Phi^T + c_k I)^{-1}$
else
$X^{k+1} = (c_k^{-1}\Psi\Phi^T + U^k)(I - \Phi(c_k I + \Phi^T\Phi)^{-1}\Phi^T)$.
end if
▷ The following section computes Y^{k+1}
Compute the Lipschitz constant
$L_2(X^{k+1}) = \|\rho_1 S X^{(k+1)T} X^{k+1} S^T + \rho_2 I\|_2$.
Compute $d_k = \gamma_2 L_2(X^{k+1})$ for some $\gamma_2 > 1$.
Compute the partial gradient
$\nabla_Y h(X^{k+1}, Y^k) = \rho_1(Y^k S(X^{k+1})^T X^{k+1} S^T + (Q - S)X^{k+1} S^T) + \rho_2(Y^k - X^{k+1})$.
Update the proximal point $V^k = Y^k - \frac{1}{d_k}\nabla_Y h(X^{k+1}, Y^k)$.
if g is the cardinality constraint (8) **then**
$Y^{k+1} = \mathcal{I}_s \circ V^k$, where $(\mathcal{I}_s)_{ij} = 1$ if $(|V^k|)_{ij} \geq s$-th largest element of $|V^k|$, and $(\mathcal{I}_s)_{ij} = 0$ otherwise.
else if g is the rank constraint (10) **then**
Y^{k+1} is the rank-r truncated SVD of V^k.
end if
end for

3) *There exist bounded constants q_1^-, q_1^+, q_2^-, $q_2^+ > 0$ such that*

$$\begin{aligned} \inf_k\{L_1(Y^k)\} \geq q_1^- \quad \textit{and} \quad \inf_k\{L_2(X^k)\} \geq q_2^- \\ \sup_k\{L_1(Y^k)\} \leq q_1^+ \quad \textit{and} \quad \sup_k\{L_2(X^k)\} \leq q_2^+. \end{aligned} \tag{22}$$

4) *The entire gradient $\nabla h(X, Y)$ is Lipschitz continuous on the bounded subsets of $\mathbb{R}^{p\times p} \times \mathbb{R}^{p\times p}$.*

Proof. Property 1 is a direct consequence of the nonnegativity of f in (12), h in (16), and the indicator function g in (13) and (14). Property 2 follows from the Lipschitz constants derived in (21). To show property 3, note that $L_1(Y)$ in (21) is clearly bounded below for all Y. In particular,

$$L_1^2(Y) = \rho_1^2\|S^T Y^T Y S\|_F^2 + 2\rho_1\rho_2\|YS\|_F^2 + \rho_2^2 \geq \rho_2^2 > 0.$$

On the other hand, since Y^k is the minimizer of a feasible problem over a bounded set, it is bounded for all k and hence $L_1(Y^k)$ is bounded above. Thus, the entire sequence $L_1(Y^k)$ satisfies the upper and lower bounds in (22). An analogous argument shows that the Lipschitz constant $L_2(X)$ satisfies (22). Property 4 is a direct consequence of the twice continuous differentiability of h and the mean value theorem. □

A few comments are in order. Property 1 ensures that each proximal operator in PALM is well defined, as well as the minimization of ω. Property 2 on the boundedness of the Lipschitz constants is critical for convergence. Note that the block-Lipschitz property in X and Y is weaker than standard assumptions in proximal methods that require ω to be globally Lipschitz in *joint* variables (X, Y). Property 3 guarantees that the Lipschitz constants for the partial gradients are lower and upper bounded by finite numbers. Property 4 is a technical condition for controlling the distance between two consecutive steps in the sequence (X^k, Y^k).

Proposition 1. *Let $Z^k := (X^k, Y^k)$ be a sequence generated by Algorithm 1. Then,*

$$\frac{\delta}{2}\|Z^{k+1} - Z^k\|_F^2 < \omega(Z^k) - \omega(Z^{k+1}), \quad \forall k \geq 0$$

where $\delta = \min\{(\gamma_1 - 1)q_1^-, (\gamma_2 - 1)q_2^-\}$. Furthermore, $\lim_{k\to\infty}\|Z^{k+1} - Z^k\|_F^2 = 0$.

Proof. Consider the proximal operator

$$\mathbf{u}^{k+1} \in \operatorname{argmin}\left\{\eta(\mathbf{u}) + \frac{\tau}{2}\|\mathbf{u} - (\mathbf{u}^k - \frac{1}{\tau}\nabla\mathbf{h}(\mathbf{u}^k))\|^2\right\}$$

where $\mathbf{h}$ is a continuously differentiable function with Lipschitz constant $L_{\mathbf{h}}$ and η is a proper, bounded, lower semicontinuous function. Recall the sufficient decrease property of the proximal map [30, Lemma 3.2]

$$\mathbf{h}(\mathbf{u}^{k+1}) + \eta(\mathbf{u}^{k+1}) \leq \mathbf{h}(\mathbf{u}^k) + \eta(\mathbf{u}^k) - \frac{\tau - L_{\mathbf{h}}}{2}\|\mathbf{u}^{k+1} - \mathbf{u}^k\|^2. \tag{23}$$

Applying (23) to (19) and (20) yields

$$h(X^{k+1},Y^k)+f(X^{k+1}) \le h(X^k,Y^k)+f(X^k) - \frac{c_k-L_1}{2}\|X^{k+1}-X^k\|_F^2$$
$$h(X^{k+1},Y^{k+1})+g(Y^{k+1}) \le h(X^{k+1},Y^k)+g(X^k) - \frac{d_k-L_2}{2}\|Y^{k+1}-Y^k\|_F^2.$$

Adding these two inequalities leads to

$$\omega(Z^{k+1}) \le \omega(Z^k) - \frac{c_k-L_1}{2}\|X^{k+1}-X^k\|_F^2 - \frac{d_k-L_2}{2}\|Y^{k+1}-Y^k\|_F^2.$$

Since $c_k=\gamma_1 L_1$ and $d_k=\gamma_2 L_2$, we obtain

$$\begin{aligned}\omega(Z^k)-\omega(Z^{k+1}) &\ge \frac{(\gamma_1-1)L_1}{2}\|X^{k+1}-X^k\|_F^2 + \frac{(\gamma_2-1)L_2}{2}\|Y^{k+1}-Y^k\|_F^2 \\ &\ge \frac{\delta}{2}\|Z^{k+1}-Z^k\|_F^2\end{aligned}$$

where $\delta := \min\{(\gamma_1-1)q_1^-,(\gamma_2-1)q_2^-\}$ and q_1^-, q_2^- are the lower bounds of Lipschitz constants defined in (22). Since ω is bounded below and δ is strictly positive, it follows that $\lim_{k\to\infty}\|Z^{k+1}-Z^k\|_F^2=0$. This completes the proof. □

Proposition 1 guarantees that the objective value is monotonically decreasing and the PALM algorithm is globally convergent. Note that $\delta>0$ throughout iterations because $\gamma_1,\gamma_2>1$ (see Algorithm 1) and $q_1^-, q_2^->0$ (see Lemma 1). The convergence of the decision variable Z^k can be measured by the convergence of the objective value. The numerical experiments in Section V verify this convergence behavior.

We next show that Algorithm 1 converges to a critical point of ω.[1] The key step is to establish the KL property of ω.

Definition 1 (KL property [30]). *Let $\mathbf{f}:\mathbb{R}^d\to(-\infty,+\infty]$ be proper and lower semicontinuous. The function $\mathbf{f}$ is said to have the* Kurdyka-Lojasiewicz (KL) *property at $\bar{\mathbf{u}}\in dom\,\partial\mathbf{f}:=\{\mathbf{u}\in\mathbb{R}^d:\partial\mathbf{f}(\mathbf{u})\neq\emptyset\}$ if there exist $\eta\in(0,+\infty]$, a neighborhood $\mathcal{N}$ of $\bar{\mathbf{u}}$, and a scalar-valued function ψ such that for all $\mathbf{u}\in\mathcal{N}\cap\{\mathbf{f}(\bar{\mathbf{u}})<\mathbf{f}(\mathbf{u})<\mathbf{f}(\bar{\mathbf{u}})+\eta\}$, the following inequality holds: $\psi'(\mathbf{f}(\mathbf{u})-\mathbf{f}(\bar{\mathbf{u}}))\cdot dist(0,\partial\mathbf{f}(\mathbf{u}))\ge 1$, where $()'$ denotes the derivative function and $dist(x,s):=\inf\{\|y-x\|:y\in\mathbf{s}\}$ denotes the distance from a point $x\in\mathbb{R}^d$ to a set $\mathbf{s}\subset\mathbb{R}^d$. A function $\mathbf{f}$ is called a KL function if $\mathbf{f}$ satisfies the KL property at each point of the domain of the gradient $\partial\mathbf{f}$.*

While KL property is a technical condition, it is shown in [30] that a large class of nonsmooth problems that arise in modern applications satisfy the KL property. For the low-complexity autoregressive model (15), the concept of semi-algebraic function is instrumental in establishing the KL property.

Definition 2 (Semi-algebraic function [30]). *A subset $\mathcal{S}$ of $\mathbb{R}^d$ is a real* semi-algebraic set *if there exists a finite number of real polynomial functions $\mathbf{g}_{ij}$ and $\mathbf{h}_{ij}:\mathbb{R}^d\to\mathbb{R}$ such that $\mathcal{S}=\bigcup_{j=1}^p\bigcap_{i=1}^q\{\mathbf{u}\in\mathbb{R}^d:\mathbf{g}_{ij}(\mathbf{u})=0$ and $\mathbf{h}_{ij}(\mathbf{u})<0\}$. A function $\mathbf{h}:\mathbb{R}^d\to(-\infty,+\infty]$ is called semi-algebraic function if its graph $\{(\mathbf{u},v)\in\mathbb{R}^{d+1}:\mathbf{h}(\mathbf{u})=v\}$ is a semi-algebraic subset of $\mathbb{R}^{d+1}$.*

A proper, lower semicontinuous, and semi-algebraic function satisfies the KL property; see [30, Theorem 5.1]. Based on this result, we now show the KL property of ω.

Lemma 2. *The objective function ω in* (15) *satisfies the KL property.*

Proof. Since ω is the summation of smooth functions f, h and the indicator function g that is lower semicontinuous, it follows that ω is a proper and lower semicontinuous function. To show that it is a semi-algebraic function, we examine each term in ω. Clearly, f and h are semi-algebraic because they are real-valued polynomials. Moreover, the indicator function of the semi-algebraic set $\{Y\,|\,\mathbf{card}(Y)\le s\}$ is semi-algebraic, and the indicator function of the semi-algebraic set $\{Y\,|\,\mathbf{rank}(Y)\le r\}$ is also semi-algebraic; see [30]. A finite sum of semi-algebraic functions is semi-algebraic. This completes the proof. □

We conclude this section by invoking the convergence result [30, Theorem 3.1] of PALM for KL functions.

Proposition 2. *Let $Z^k=(X^k,Y^k)$ be a sequence generated by the PALM algorithm. Suppose that ω is a KL function that satisfies the properties in Lemma 1. Then the sequence $\{Z^k\}$ converges to a critical point $Z^*=(X^*,Y^*)$ of ω.*

V. Numerical Experiments

In this section, we evaluate the performance of Algorithm 1 on both synthetic and real-world data. We demonstrate that the solution converges to a matrix with the prescribed level of nonzero elements or matrix rank. Furthermore, the objective value (i.e., estimation error) decreases monotonically as predicted by the convergence analysis.

We also compare the estimation errors of the autoregressive models obtained from nonconvex and convex constraints; in particular, we focus on the cardinality constraint versus the ℓ_1 constraint. Our numerical results show that the cardinality constraint achieves a smaller error than the ℓ_1 constraint on a variety of systems drawn from the COMPLeib library [35], [36]. Moreover, we show that PALM outperforms with gradient projection method when handling the ℓ_1 constraint.

In our experiments, we assume that the covariance matrix of the noise $\epsilon(t)$ is $Q=\sigma^2 I$. We set $\gamma_1=\gamma_2=2$ in Algorithm 1. The hyperparameters ρ_1 and σ are determined through cross validation.

A. Synthetic Data

We test the performance of the proposed method on a sparse example and a low-rank example with synthetic transition matrices of size 200×200. In both examples, we use time series of length $n=50$ for training and $m=800$ for testing. For steady-state data, we set the length $N=1600$. The

[1] For nonconvex, nonsmooth functions, the critical point is understood as the points whose Frechet subdifferential contains 0.

performance of the identified autoregressive model is evaluated by using the normalized error and the cosine score proposed in [1]

$$\text{Normalized error: } \frac{1}{m-1}\sum_{t=1}^{m-1}\frac{\|\phi(t+1)-\widehat{A}\phi(t)\|}{\|\phi(t+1)-\phi(t)\|}$$

$$\text{Cosine score: } \frac{1}{m-1}\sum_{t=1}^{m-1}\frac{|(\phi(t+1)-\phi(t))^T(\phi(t)-\widehat{A}\phi(t))|}{\|\phi(t+1)-\phi(t)\|\|\phi(t)-\widehat{A}\phi(t)\|}.$$

A smaller normalized error (lower bounded by 0) and a higher cosine score (upper bounded by 1) imply better performance.

1) Sparse Example: The sparse matrix is generated by using the rule $A = (0.95M)/\max_k(|\lambda_k(M)|)$, where M has 5000 normally distributed nonzero elements and $\lambda_k(M)$ denotes the eigenvalues of M. We set $s = 5000$ in the cardinality constraint (8).

Figure 1 shows the convergence results. The objective value monotonically decreases, as Proposition 1 indicates. The errors in two consecutive steps, namely, $e_X^k = \|X^{k+1} - X^k\|_F$, $e_Y^k = \|Y^{k+1} - Y^k\|_F$, $e_{XY}^k = \|X^k - Y^k\|_F$, all decrease quickly. It takes fewer than 30 iterations to reach $e_X, e_Y \leq 10^{-4}$ and $e_{XY} \leq 3.5 \times 10^{-4}$. Note that the solution has exactly 5000 nonzero elements as required by the cardinality constraint. For the estimated matrix $\widehat{A}$, the normalized error is 0.2848 and the cosine score is 0.9582.

2) Low-Rank Example: The low-rank matrix is generated by using the rule $A = \mathcal{U}\Sigma\mathcal{V}$, where $\Sigma \in \mathbb{R}^{25\times 25}$ is a diagonal matrix with random diagonal entries uniformly distributed in $[0, 1)$, and $\mathcal{U} \in \mathbb{R}^{200\times 25}$ and $\mathcal{V} \in \mathbb{R}^{25\times 200}$ are random orthonormal matrices. By construction $A \in \mathbb{R}^{200\times 200}$ is stable with $\mathbf{rank}(A) = 25$. We set $r = 25$ in the rank constraint (10).

Figure 2 shows the convergence results. Similar to those for the sparse example in Figure 1, we observe that the objective value ω monotonically decreases and the errors in two consecutive steps decrease quickly. It takes fewer than 30 iterations to reach $e_X, e_Y \leq 3 \times 10^{-5}$ and $e_{XY} \leq 2 \times 10^{-4}$. The solution has a numerical rank 25, as required by the rank constraint. For the estimated matrix $\widehat{A}$, the normalized error is 0.6949 and the cosine score is 0.7189.

B. Electricity Load Data

We explore the utility of the low-complexity models on an electricity load data set from the UCI repository.[2] The data set consists of 15-minute interval load readings of clients over 1461 days. To investigate the daily dynamics, we aggregate the data in every 24-hour interval. Because over half of the clients are not registered in the first year, we start from the second-year data and collect clients whose time series data are uninterrupted. Interruptions may arise from late registration of clients, missing data, or a period of low electricity consumption due to inactivity. Such a preprocessing results in 272 clients and 1095 daily readings per client. We further subtract each time series by its seasonal mean, that is, the mean of the same day along all the years, and normalize it by the standard deviation.

[2]http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

We first use the least squares estimator (2) to obtain a reference model. To this end, the first 995 days are used for training and the last 100 days are used for testing. Note that the reference model provides an upper bound on the performance, because the number n of measurements is sufficiently greater than the data dimension p.

Next, we test the low-complexity models (11) with different thresholds for the cardinality and the rank constraint. In particular, we set $s \in \{100p, 125p, 150p, 175p, 200p, p^2\}$ and similarly $r \in \{100, 125, 150, 175, 200, p\}$. We take the first n days with $n \in \{25, 50, 75, 100\}$ as the training data, the last 100 days as the testing data, and 600 randomly sampled days in the remaining dataset as the steady-state data. We repeat the experiment five times for each set of s, r, and n.

Figure 3 shows the performance measures as the number n of training data and the sparsity level s vary. Three observations can be made. First, the normalized error and the cosine score are not sensitive to the length of the training data, because the performance varies slightly with n. This fact indicates that the Lyapunov penalty as a regularization is effective. Second, as the complexity of the transition matrix increases (e.g., a larger s), the performance gets closer to that of the least squares estimator. Third, in the case of no constraints (i.e., $s = p^2$), the Lyapunov-penalized VAR model (4) performs as well as the least squares estimator (2). In other words, the Lyapunov-penalized VAR model with a small number of time sequence data and a large number of nonsequence data is as competitive as the least squares estimator with a large amount of time sequence data. This result demonstrates the utility of the proposed method when time sequence data is limited.

C. Comparison different penalties and different methods

We test on a variety of dynamical systems from the *COMPleib* library [35], [36]. This set of systems is drawn from aircraft, helicopter, jet engine, reactor, decentralized interconnected systems, and wind energy systems. The set consists of 40 continuous-time systems with the dimension of the state matrix $A_c \in \mathbb{R}^{p\times p}$ ranging from $p = 3$ to $p = 40$. For each model, we generate $n = p/2$ sequence data and $N = 5n$ nonsequence data for training, and $m = p$ sequence data for testing.

We use PALM to solve the problem with cardinality constraint (8) and ℓ_1 constraint (7). For a fair comparison, the number of nonzero elements of the solution $\widehat{A}$ must be the same in both cases. For the cardinality constraint, we set the desired number of nonzeros to be $s = \alpha p^2$ for $\alpha \in \{1/2, 1/4, 1/8\}$. For the ℓ_1 constraint $\|X\|_{\ell_1} \leq l$, the upper bound l that yields the desired number of nonzero elements is unknown a priori. To find the matching l, we use a bisection method: Starting from an interval $[l_{\text{low}}, l_{\text{up}}]$ that contains the unknown l, repeatedly solve (11) and divide the interval by half, until l for the desired number of nonzero elements is found or the interval is sufficiently small. We also use gradient projection (GP) to solve (11) with the ℓ_1 constraint.

Table I shows the performance of PALM-card, PALM-ℓ_1, and GP-ℓ_1 methods. PALM-card outperforms the other two

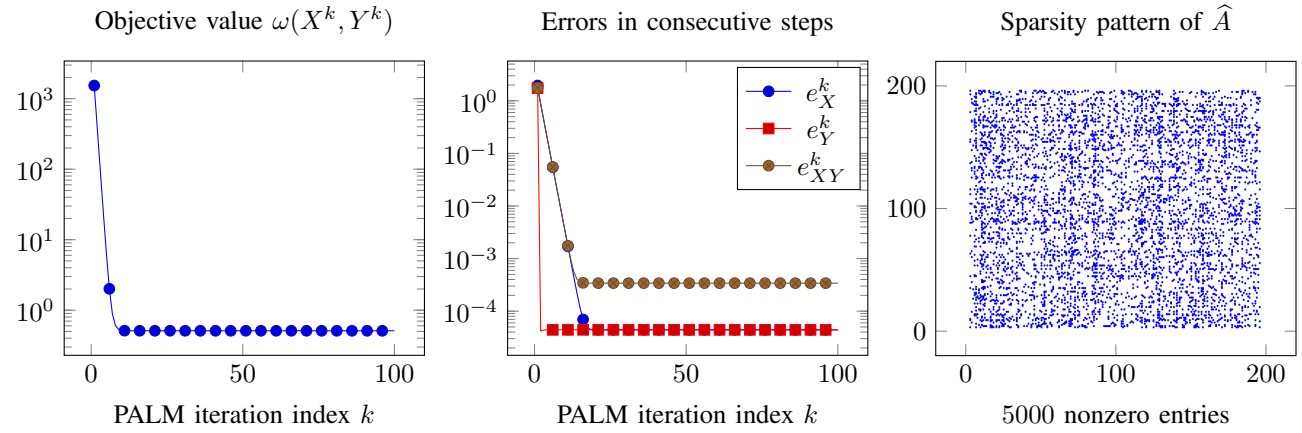

Fig. 1: Convergence results of PALM for the sparse example: the objective value (left), the errors in consecutive steps (middle), and the sparse solution with 5000 nonzero entries (right).

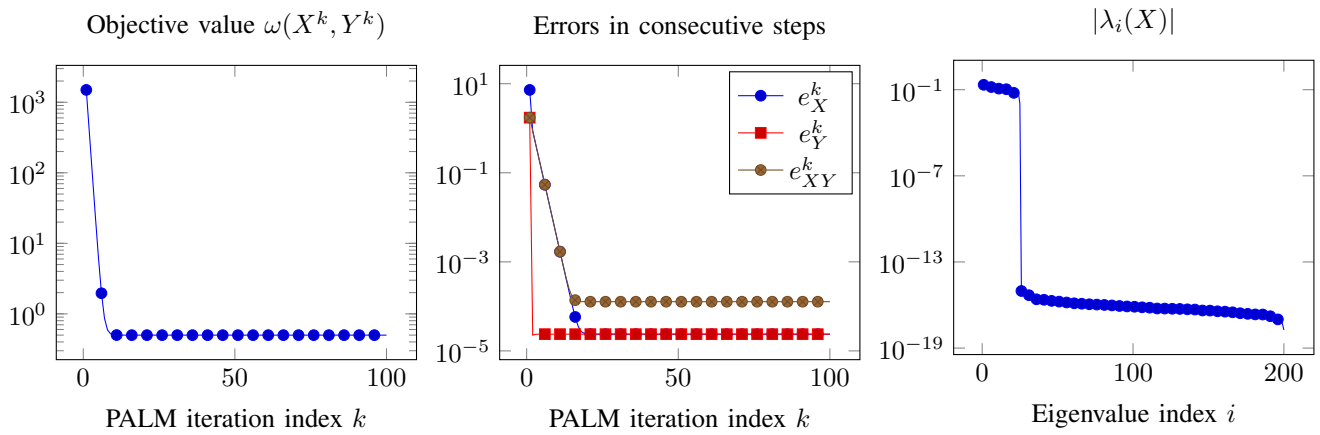

Fig. 2: Convergence results of PALM for the low-rank example: the objective value (left), the errors in consecutive steps (middle), and the low-rank solution with 25 nonzero eigenvalues (right).

approaches in achieving a smaller normalized error and a higher cosine score. The percentage of cases where PALM-card outperforms the others increases with the level of sparsity, from 62.5% for $\alpha = 1/4$ to 82.5% for $\alpha = 1/8$. Similarly PALM-card yields the highest cosine score in 60% of the systems when $\alpha = 1/4$ and in 77.5% of the systems when $\alpha = 1/8$. When the ℓ_1 constraint is used, PALM outperforms GP when $\alpha = 1/2$ and $\alpha = 1/4$.

Figure 4 shows the normalized error and the cosine score for the three methods when $\alpha = 1/4$. Note that for 14 test problems, the errors resulted from GP-ℓ_1 is at least two times (and up to 43 times) of the errors from PALM-card and PALM-ℓ_1. Similar observations can be made for the cosine score. For 12 test problems, PALM-card and PALM-ℓ_1 result in cosine scores that are at least twice of those obtained from GP-ℓ_1. These results suggest that the cardinality constraint is more effective than the ℓ_1 constraint and that PALM outperforms GP by converging to better solutions.

As mentioned earlier, PALM requires no tuning for the stepsize in contrast to GP. This feature makes PALM computationally more efficient when the projection onto the constraint set becomes nontrivial. For the projection onto the ℓ_1-ball, it turns out that the most time-consuming computation in GP is to compute the stepsize by using the Armijo rule along the projection-arc [37]. This is because GP requires a number of ℓ_1 projections to compute the stepsize. As a consequence, for the ℓ_1 constraint PALM is computationally more efficient than GP.

VI. Conclusions

We estimate the state transition matrix of a vector autoregressive model, with limited time sequence data but abundant nonsequence steady-state data. To reduce the complexity of the model, we propose imposing a cardinality or a rank constraint on the transition matrix. We develop the PALM algorithm to solve the resulting nonconvex, nonsmooth problem and establish its global convergence to a critical point. Numerical experiments empirically verify the convergence and demonstrate the advantage of PALM over the gradient projection method.

Several directions may be pursued following this work. First, we observe a linear convergence of the algorithm (e.g., Fig. 1 and Fig. 2). We intend to investigate the convergence rate theoretically. Second, the identified model is only one

TABLE I: Performance of PALM-card, PALM-ℓ_1, and GP-ℓ_1 on COMPLeib test problems. The sparsity level is indicated by $\alpha = s/p^2$. The table shows the number of test problems where each method outperforms the other two. For example, when $\alpha = 1/4$, PALM-card achieves the smallest normalized error in 30 test problems and the highest cosine score in 27 test problems.

	Normalized error			Cosine score		
α	PALM-card	PALM-ℓ_1	GP-ℓ_1	PALM-card	PALM-ℓ_1	GP-ℓ_1
1/2	25	12	3	24	12	4
1/4	30	6	4	27	10	3
1/8	33	2	5	31	3	6

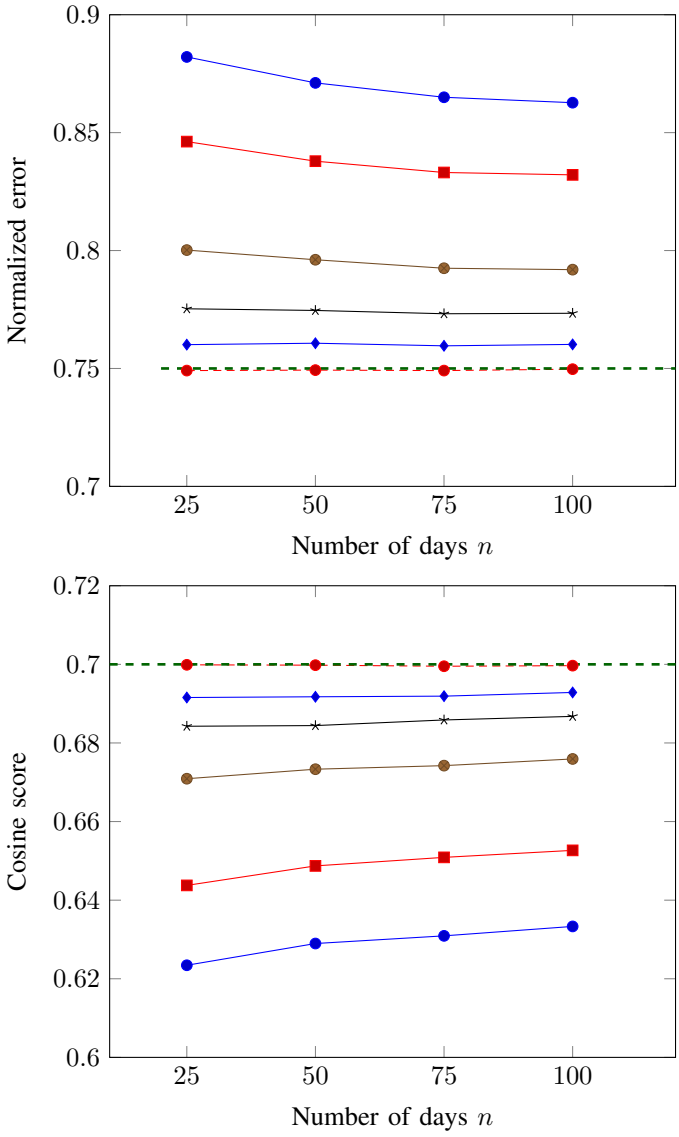

Fig. 3: Electricity load data: Performance comparison between the least squares estimator (2) when $p < n$ (–) and the Lyapunov-penalized model (11) when $p > n$, with different levels of cardinality: $s = 100p$ (●), $s = 125p$ (■), $s = 150p$ (●), $s = 175p$ (∗), $s = 200p$ (◇), $s = p^2$ (●).

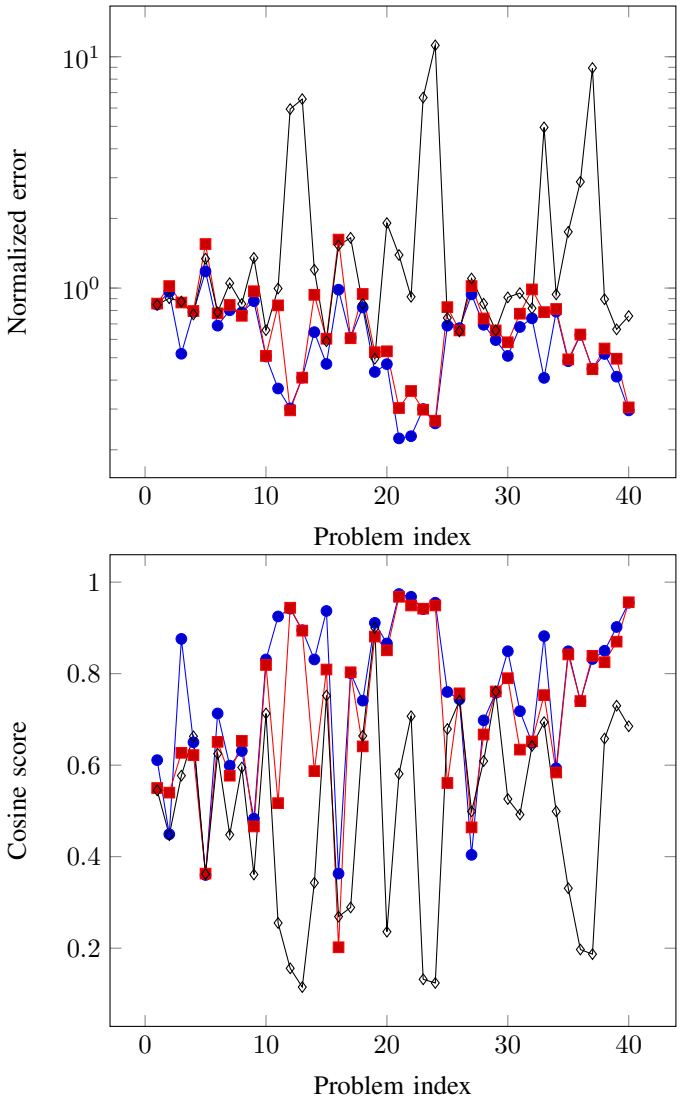

Fig. 4: Performance of PALM-card (●), PALM-ℓ_1 (■), and GP-ℓ_1 (◇) on COMPLeib test problems, with sparsity level $\alpha = 1/4$.

of many legitimate models that explain the given data. It is thus of interest to understand under what conditions the low-complexity model is asymptotically consistent with the ground truth, if it is sparse or low-rank in the first place. Third, while the VAR model itself has low complexity, the optimization algorithm still requires storage and computation with $p \times p$ matrices. When p is too large, it would be interesting to investigate methods that reduce the cost through approximately updating the unknowns (for example, in the rank-constraint case, randomized SVD is more efficient than standard SVD).

ACKNOWLEDGMENTS

We thank the reviewers for constructive comments that improve this work. F. Lin is supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02-06CH11357. J. Chen is supported in part by XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

REFERENCES

[1] T.-K. Huang and J. G. Schneider, "Learning auto-regressive models from sequence and non-sequence data," in *Advances in Neural Information Processing Systems*, 2011, pp. 1548–1556.

[2] R. Yoshida, S. Imoto, and T. Higuchi, "Estimating time-dependent gene networks from time series microarray data by dynamic linear models with Markov switching," in *Proceedings of the 2005 Computational Systems Bioinformatics Conference*, 2005, pp. 289–298.

[3] M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas, "Inferring stable genetic networks from steady-state data," *Automatica*, vol. 47, no. 6, pp. 1113–1122, 2011.

[4] Y. K. Wang, D. G. Hurley, S. Schnell, E. J. Crampin *et al.*, "Integration of steady-state and temporal gene expression data for the inference of gene regulatory networks," *PloS one*, vol. 8, no. 8, p. e72103, 2013.

[5] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira, "Modeling gene expression regulatory networks with the sparse vector autoregressive model," *BMC Systems Biology*, vol. 1, no. 1, p. 39, 2007.

[6] H. Wang, G. Li, and C.-L. Tsai, "Regression coefficient and autoregressive order shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 1, pp. 63–78, 2007.

[7] A. Gupta and Z. Bar-Joseph, "Extracting dynamics from static cancer expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, pp. 172–182, 2008.

[8] T.-K. Huang and J. Schneider, "Learning linear dynamical systems without sequence information," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 425–432.

[9] F. Han and H. Liu, "Transition matrix estimation in high dimensional time series," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 172–180.

[10] M. T. Bahadori, Y. Liu, and E. P. Xing, "Fast structure learning in generalized stochastic processes with latent factors," in *Proceedings of the 19th SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 284–292.

[11] P. Geiger, K. Zhang, B. Schoelkopf, M. Gong, and D. Janzing, "Causal inference by identification of vector autoregressive processes with hidden components," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1917–1925.

[12] T.-K. Huang, L. Song, and J. Schneider, "Learning nonlinear dynamic models from nonsequenced data," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 350–357.

[13] J. E. Larvie, M. S. Gorji, and A. Homaifar, "Inferring stable gene regulatory networks from steady-state data," in *41st Annual Northeast Biomedical Engineering Conference*, 2015, pp. 1–2.

[14] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*. Springer Science & Business Media, 2013, vol. 36.

[15] A. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical granger modeling for gene expression regulatory networks discovery," *Bioinformatics*, vol. 25, no. 12, pp. i110–i118, 2009.

[16] M. Overton and R. Womersley, "On minimizing the spectral radius of a nonsymmetric matrix function: Optimality conditions and duality theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, pp. 474–498, 1988.

[17] Y. Kim, D.-W. Gu, and I. Postlethwaite, "Spectral radius minimization for optimal average consensus and output feedback stabilization," *Automatica*, vol. 45, no. 6, pp. 1379–1386, 2009.

[18] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[19] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[20] J. V. Burke and M. L. Overton, "Variational analysis of non-lipschitz spectral functions," *Mathematical Programming*, vol. 90, no. 2, pp. 317–351, 2001.

[21] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the 2001 American Control Conference*, 2001, pp. 4734–4739.

[22] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.

[23] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[24] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.

[25] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[26] D. Hajinezhad, T.-H. Chang, X. Wang, Q. Shi, and M. Hong, "Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 4742–4746.

[27] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 438–457, 2010.

[28] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[29] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[30] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[31] M. J. D. Powell, "On search directions for minimization algorithms," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, 1973.

[32] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 272–279.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[34] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *arXiv preprint arXiv:1511.06324*, 2018.

[35] F. Leibfritz, "Compleib: Constraint matrix optimization problem library - A collection of test examples for nonlinear semidefinite programs, control system design and related problems," University of Trier, Tech. Rep., 2004.

[36] F. Leibfritz and W. Lipinski, "COMPleib 1.0 - user manual and quick reference," University of Trier, Tech. Rep., 2004.

[37] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific Belmont, 1999.