Learning Low-Complexity Autoregressive Models via Proximal Alternating Minimization

Fu Lin[a,*], Jie Chen[b]

[a] *Systems Department, United Technologies Research Center, 411 Silver Ln, East Hartford, CT 06118, USA.*
[b] *IBM Thomas J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA.*

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. The resulting nonconvex, nonsmooth problem is solved by using the proximal alternating linearization method (PALM). We prove that PALM is globally convergent to a critical point and that the estimation error monotonically decreases. Explicit formulas are obtained for the proximal operators to facilitate the implementation of PALM. We demonstrate the effectiveness of the developed method by numerical experiments.

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

1. 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 sufficient time sequence data

*Corresponding author
Email addresses: linf@utrc.utc.com (Fu Lin), chenjie@us.ibm.com (Jie Chen)

exist, the standard approach is to solve a least-squares problem of estimation errors. In modern applications, however, the dimension of the model can exceed the number of time sequence measurements, which results in multiple models that fit the data equally well through the standard least-squares approach. Such scenarios include tracking the progression of brain neurological diseases, for example, 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 limited number of measurements due to the intrusive nature of the measuring techniques [2, 3].

In such situations, one typically employs regularization techniques to induce additional modeling features. For example, ridge regularization is a common approach for ensuring a unique model. Other regularization methods introduce additional structures to the solution; in particular, sparsity and low-rank structures are extensively studied. These low-complexity regularization approaches are widely used partially because the resulting problem may be efficiently solved via convex optimization [4, 5, 6, 2, 1, 7]. In [4], a sparse VAR model for gene regulatory networks is obtained via LASSO. In [8], 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, 2, 1, 7].

The steady states of a dynamical system driven by white noise provide valuable data for improving model accuracy. Several authors show that when the VAR model is stable, the steady-state data can be utilized to reduce the estimation error [6, 9, 1, 2, 3]. In [1], the Lyapunov regularization is proposed to exploit the second-order statistics of the steady-state data. In [2], the perturbed steady-state data is used to infer sparse, stable gene expression networks. In [3], 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 [6, 9].

In this paper, we leverage both time sequence and steady-state nonsequence data for the model estimation. We propose a least-squares estimator for the time

sequence data, regularized by the Lyapunov penalty on the second-order errors of the steady-state data. In addition, the state transition matrix is subject to the sparsity and matrix rank constraints. The identification problem is nonconvex due to the Lyapunov penalty and nonsmooth due to the low-complexity constraints. We develop the proximal alternating linearization method (PALM) and prove that it converges to a critical point starting from any initial condition. Furthermore, we show that the estimation error is monotonically decreasing with the PALM iterations. Closed-form expressions are obtained for the proximal operators to facilitate implementation. We show that PALM can handle the stability constraints and also the convex low-complexity constraints (e.g., the ℓ_1 norm or the nuclear norm).

Our presentation is organized as follows. In Section 2, we formulate the estimation problem for the low-complexity VAR model. In Section 3, we present the PALM algorithm and derive explicit formulas for the proximal operators. In Section 4, we prove the global convergence of PALM. In Section 5, we demonstrate the effectiveness of PALM via numerical experiments. In Section 6, we summarize our contributions.

2. 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 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: $\phi(t+1) = A\phi(t)+\epsilon(t)$, 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 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 $\epsilon(t)$. Note that P is positive definite if and only if A is asymptotically stable.

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

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

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 (1).

We are interested in the scenario when the time sequence data is scarce but the steady-state nonsequence data is readily available [6, 9, 1]. Huang and Schneider [1] propose the Lyapunov penalty as a regularization term $\|XPX^T + Q - P\|_F^2$. 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 $S := \frac{1}{N}\sum_{i=1}^N (z^i - \bar{z})(z^i - \bar{z})^T$ where $\bar{z} := \frac{1}{N}\sum_{i=1}^N z^i$, and $\{z^i\}_{i=1}^N$ is the steady-state nonsequence data. The identification problem with the Lyapunov regularization can be expressed as

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

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

In [1], Huang and Schneider show that the Lyapunov penalty improves the quality of estimation. However, there is no guarantee that the solution of (2) is stable (i.e., spectral radius of X is less than 1). We next incorporate stability constraint into (2).

2.1. Stability Constraint

Since stability is a necessary condition for the use of Lyapunov penalty, a stability constraint is included in the identification problem (2). 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:

$$\begin{array}{ll} \underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{\rho}{2}\|XSX^T + Q - S\|_F^2 \\ \text{subject to} & \tau(X) < 1. \end{array} \tag{3}$$

Dealing with τ directly in optimization is difficult because spectral radius is neither convex nor locally Lipschitz [10]. One approach is to employ a convex proxy as a conservative upper bound [11]. Since $\tau(X) \leq \sigma_{\max}(X) \leq \|X\|_F$, we can incorporate the stability constraint in the cost function

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

where μ is a positive constant. While the spectral norm, $\sigma_{\max}(X)$, is less conservative than the Frobenius norm, we adopt the latter because both the Lyapunov penalty and the stability penalty $\|XX^T\|_F^2$ are quadratic functions of X in Frobenius norm squared. Hence, the stability term 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 [12, 11, 10].

2.2. Low-Complexity Models

In several applications, it is desired to impose sparsity or low-rank structures on the state transition matrix [4, 1, 2, 8]. In gene expression networks, for example, the nonzero elements of the state transition matrix determine the interaction graph of the expression network [4, 2]. A sparse state transition ma-

trix 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:

$$\|X\|_{\ell_1} := \sum_{i,j=1}^{p} |X_{ij}| \leq l, \tag{5}$$

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:

$$\mathbf{card}(X) := \text{ number of nonzero entries of } X \leq s, \tag{6}$$

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 lower dimension model. An implicit way to promote low-rank solutions is to use the nuclear norm

$$\|X\|_* := \sum_{i=1}^{p} \sigma_i(X) \leq \nu, \tag{7}$$

where ν is a prescribed positive number and the σ_i 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:

$$\mathbf{rank}(X) := \text{ number of nonzero singular values of } X \leq r, \tag{8}$$

where r is a given positive integer.

Hence we consider the following estimation problem:

$$\begin{aligned} \widehat{A} = \underset{X\in\mathbb{R}^{p\times p}}{\operatorname{argmin}} \ & \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{\rho}{2}\|XSX^T + Q - S\|_F^2 \\ & \text{subject to constraint (5) or (6) or (7) or (8).} \end{aligned} \tag{9}$$

For the convex constraints (5) and (7), 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 (9) with the ℓ_1 constraint (5). For the nonconvex constraints (6) and (8), on the other hand, we next develop the PALM algorithm.

3. 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 globally converge to a critical point for both convex and nonconvex constraints in (9).

We begin with a reformulation of (9)

$$\begin{aligned} \underset{X,Y\in\mathbb{R}^{p\times p}}{\text{minimize}} \quad & \frac{1}{2}\|X\Phi - \Psi\|_F^2 + \frac{\rho}{2}\|YSX^T + Q - S\|_F^2 \\ \text{subject to} \quad & Y - X = 0, \\ & \text{(5) or (6) or (7) or (8).} \end{aligned}$$

Let f denote the least-squares term $f(X) = \frac{1}{2}\|X\Phi - \Psi\|_F^2$, and let g denote the indicator function of the individual constraints in (5)-(8). For the cardinality constraint (6), for example, $g(Y) = 0$ if $\mathbf{card}(Y) \le s$ and $g(Y) = \infty$ otherwise. Then we have

$$\underset{X,Y\in\mathbb{R}^{p\times p}}{\text{minimize}} \ \omega(X,Y) := f(X) + g(Y) + h(X,Y), \tag{10}$$

where h denotes the coupling term

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

Here, the penalty parameter $\rho_1 > 0$ resumes the role of ρ in (9) 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.

3.1. Generic PALM

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 [13, 14, 15]. 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):

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

Convergence of the iteration requires a unique solution in each minimization step; otherwise, Gauss-Seidel may cycle indefinitely [16]. When ω is convex but *not strictly* convex, uniqueness can be achieved by including a quadratic proximal term

$$X^{k+1} \;\in\; \operatorname*{argmin}_X \left\{\omega(X,Y^k) \;+\; \frac{c_k}{2}\|X \;-\; X^k\|_F^2\right\} \tag{11a}$$

$$Y^{k+1} \;\in\; \operatorname*{argmin}_Y \left\{\omega(X^{k+1},Y) \;+\; \frac{d_k}{2}\|Y \;-\; Y^k\|_F^2\right\}, \tag{11b}$$

where c_k and d_k are positive coefficients. This class of proximal methods is well studied; see [14] for a recent survey.

When ω is nonconvex, as in our case (10), we need to modify the proximal terms to ensure convergence. Instead of taking the proximal term around X^k

as in (11a), we take the term around X^k modified with a scaled partial gradient of h:

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{ f(X) + \frac{c_k}{2}\|X - U^k\|_F^2 \right\}, \tag{12}$$

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:

$$Y^{k+1} \in \underset{Y}{\operatorname{argmin}} \left\{ g(Y) + \frac{d_k}{2}\|Y - V^k\|_F^2 \right\}, \tag{13}$$

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 (12)-(13).

3.2. Formulas for Lipschitz Constants and Solutions to (12)-(13)

To implement (12)-(13), 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 $\nabla_X h = \rho_1(XS^TY^TYS + (Q-S)^TYS) + \rho_2(X-Y)$ and $\nabla_Y h = \rho_1(YSX^TXS^T + (Q-S)XS^T) + \rho_2(Y-X)$. Since $\nabla_X h$ is linear in X and $\nabla_Y h$ is linear in Y, we obtain formulas for Lipschitz constants

$$L_1(Y) = \|\rho_1 S^T Y^T Y S + \rho_2 I\|_2, \quad L_2(X) = \|\rho_1 S X^T X S^T + \rho_2 I\|_2 \tag{14}$$

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

We next show that the proximal operators (12)-(13) can be computed efficiently. The proximal operator (12) 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 (13) can be expressed as

$$\begin{array}{ll} \underset{Y}{\text{minimize}} & \dfrac{d_k}{2}\|Y - V^k\|_F^2 \\ \text{subject to} & (5) \text{ or } (6) \text{ or } (7) \text{ or } (8). \end{array}$$

For the cardinality constraint (6), 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 (8), 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 (5), the projection onto the ℓ_1-ball can be computed by an algorithm developed in [17]. For the nuclear-norm constraint (7), 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 the constraints (6) and (8). We conclude this section with a remark on stability.

Remark 1 (Stability)**.** *As discussed in Section 2.1, we can incorporate the stability constraint by penalizing $\|XX^T\|_F^2$ in the cost function. In this case, the coupling term becomes $h(X,Y) = \dfrac{\rho_1}{2}\|YSX^T + Q - S\|_F^2 + \dfrac{\rho_2}{2}\|X - Y\|_F^2 + \dfrac{\mu}{2}\|YX^T\|_F^2$. Its partial gradients are $\nabla_X h = \rho_1(XS^TY^TYS + (Q-S)^TYS) + \rho_2(X - Y) + \mu XY^TY$ and $\nabla_Y h = \rho_1(YSX^TXS^T + (Q-S)XS^T) + \rho_2(Y - X) + \mu YX^TX$, whose Lipschitz constants are given by $L_1(Y) = \|\rho_1 S^TY^TYS + \mu Y^TY + \rho_2 I\|_2$ and $L_2(X) = \|\rho_1 SX^TXS^T + \mu X^TX + \rho_2 I\|_2$. 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*

Algorithm 1 Proximal Alternating Linearization Method for (10)

Initialization: Start with any (X^0, Y^0).
for $k = 0, 1, 2, \ldots$ until convergence **do**
  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
  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 (6) **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 (8) **then**
    Y^{k+1} is the rank-r truncated SVD of V^k.
  end if
end for

optimization [18, 19, 20]. Since ADMM is a class of proximal algorithms [14], 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 [18, 19, 20]. For the cardinality (6) *and the rank function* (8)*, ADMM may not converge for* (9)*. The solution to which ADMM converges may also depend on the value of ρ; see [18]. Furthermore, efficient methods for subproblems in ADMM that deal with the Lyapunov penalty are yet to be developed.*

4. Convergence Analysis

In this section, we show that Algorithm 1 globally converges to a critical point of the nonconvex, nonsmooth problem (10). 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 [15]. 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 (10) 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.*
3. *There exist bounded constants q_1^-, q_1^+, q_2^-, $q_2^+ > 0$ such that*

$$\inf_k\{L_1(Y^k)\} \geq q_1^-, \inf_k\{L_2(X^k)\} \geq q_2^-, \sup_k\{L_1(Y^k)\} \leq q_1^+, \sup_k\{L_2(X^k)\} \leq q_2^+. \tag{15}$$

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 and h, and the indicator function g. Property 2 follows from the Lipschitz constants derived in (14). To show property 3, note that $L_1(Y)$ in (14) is clearly bounded below for all Y. In particular, $L_1^2(Y) = \rho_1^2\|S^TY^TYS\|_F^2 + 2\rho_1\rho_2\|YS\|_F^2 + \rho_2^2 \geq \rho_2^2 > 0$. 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 (15). An analogous argument shows that the Lipschitz constant $L_2(X)$ satisfies (15).

Property 4 is a direct consequence of the twice continuous differentiability of h and the mean value theorem. $\square$

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, $\frac{\delta}{2}\|Z^{k+1} - Z^k\|_F^2 < \omega(Z^k) - \omega(Z^{k+1}), \forall k \geq 0$ 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 $\mathbf{u}^{k+1} \in \operatorname{argmin}\left\{\eta(\mathbf{u}) + \frac{\tau}{2}\|\mathbf{u} - (\mathbf{u}^k - \frac{1}{\tau}\nabla\mathbf{h}(\mathbf{u}^k))\|^2\right\}$ 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 [15, Lemma 3.2]

$$\mathbf{h}(\mathbf{u}^{k+1}) + \eta(\mathbf{u}^{k+1}) \leq \mathbf{h}(\mathbf{u}^k) + \eta(\mathbf{u}^k) - \frac{\tau - L_{\mathbf{h}}}{2}\|\mathbf{u}^{k+1} - \mathbf{u}^k\|^2. \tag{16}$$

Applying (16) to (12) and (13) yields

$$\begin{aligned} h(X^{k+1}, Y^k) + f(X^{k+1}) &\leq h(X^k, Y^k) + f(X^k) - \tfrac{c_k - L_1}{2}\|X^{k+1} - X^k\|_F^2 \\ h(X^{k+1}, Y^{k+1}) + g(Y^{k+1}) &\leq h(X^{k+1}, Y^k) + g(X^k) - \tfrac{d_k - L_2}{2}\|Y^{k+1} - Y^k\|_F^2. \end{aligned}$$

Adding these two inequalities leads to

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

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

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

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 (15). 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 5 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 [15])**.** *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: $\psi'(\mathbf{f}(\mathbf{u}) - \mathbf{f}(\bar{\mathbf{u}})) \cdot dist(0, \partial\mathbf{f}(\mathbf{u})) \geq 1$, 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 [15] that a large class of nonsmooth problems that arise in modern applications satisfy the KL property. For the low-complexity autoregressive model (10), the concept of

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

semi-algebraic function is instrumental in establishing the KL property.

Definition 2 (Semi-algebraic function [15]). *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 $\mathcal{S} = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{\mathbf{u} \in \mathbb{R}^d : \mathbf{g}_{ij}(\mathbf{u}) = 0 \text{ and } \mathbf{h}_{ij}(\mathbf{u}) < 0\}$. 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 [15, Theorem 5.1]. We now show the KL property of ω.

Lemma 2. *The objective function ω in (10) 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 $\{Y \,|\, \mathbf{card}(Y) \leq s\}$ is semi-algebraic, and the indicator function of the semi-algebraic set $\{Y \,|\, \mathbf{rank}(Y) \leq r\}$ is also semi-algebraic; see [15]. A finite sum of semi-algebraic functions is semi-algebraic. This completes the proof. □

We conclude this section by invoking the convergence result [15, 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.*

5. Numerical Experiments

In this section, we evaluate the performance of Algorithm 1 via numerical experiements. We conduct extensive experiments on both synthetic and real-world data and compare PALM with gradient projection method; see [21]. Due to space limitation, we report the performance of PALM on a sparse example and

a low-rank example. 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 in the convergence analysis.

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.

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 performance of the identified autoregressive model is evaluated by using the normalized error [1] $\frac{1}{m-1}\sum_{t=1}^{m-1}\frac{\|\phi(t+1)-\widehat{A}\phi(t)\|}{\|\phi(t+1)-\phi(t)\|}$ and 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.

5.1. 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 (6). 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.

5.2. 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

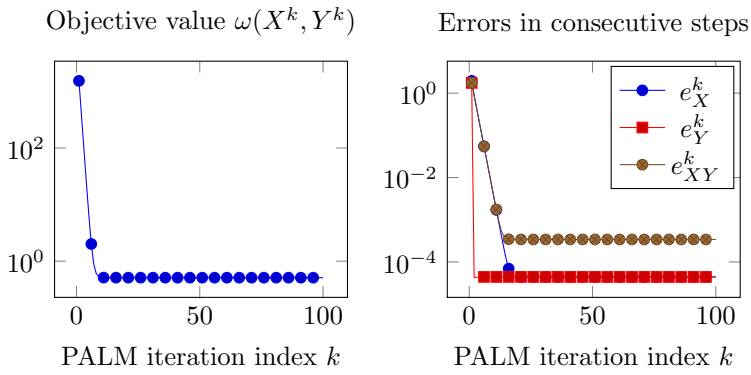

Figure 1: Convergence results of PALM for the sparse example: the objective value (left) and the errors in consecutive steps (right).

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 (8).

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.

Additional numerical experiments on real-world data and comparison of PALM with gradient projection method can be found in [21].

6. 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 a cardinality and 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 behavior of PALM.

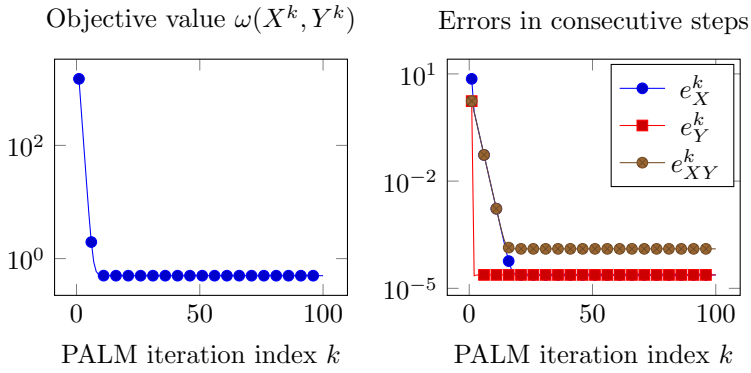

Figure 2: Convergence results of PALM for the low-rank example: the objective value (left) and the errors in consecutive steps (right).

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, 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] M. M. Zavlanos, A. A. Julius, S. P. Boyd, G. J. Pappas, Inferring stable genetic networks from steady-state data, Automatica 47 (6) (2011) 1113–1122.

[3] 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 8 (8) (2013) e72103.

[4] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, C. E. Ferreira, Modeling gene expression regulatory networks with the sparse vector autoregressive model, BMC Systems Biology 1 (1) (2007) 39.

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

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

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

[8] M. T. Bahadori, Y. Liu, 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.

[9] T.-K. Huang, L. Song, 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.

[10] Y. Kim, D.-W. Gu, I. Postlethwaite, Spectral radius minimization for optimal average consensus and output feedback stabilization, Automatica 45 (6) (2009) 1379–1386.

[11] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, Systems & Control Letters 53 (1) (2004) 65–78.

[12] J. V. Burke, M. L. Overton, Variational analysis of non-lipschitz spectral functions, Mathematical Programming 90 (2) (2001) 317–351.

[13] H. Attouch, J. Bolte, P. Redont, A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality, Mathematics of Operations Research 35 (2) (2010) 438–457.

[14] N. Parikh, S. Boyd, Proximal algorithms, Foundations and Trends in Optimization 1 (3) (2013) 123–231.

[15] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1-2) (2014) 459–494.

[16] M. J. D. Powell, On search directions for minimization algorithms, Mathematical Programming 4 (1) (1973) 193–201.

[17] J. Duchi, S. Shalev-Shwartz, Y. Singer, 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.

[18] M. Hong, Z.-Q. Luo, M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, SIAM Journal on Optimization 26 (1) (2016) 337–364.

[19] D. Hajinezhad, T.-H. Chang, X. Wang, Q. Shi, 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.

[20] Y. Wang, W. Yin, J. Zeng, Global convergence of admm in nonconvex nonsmooth optimization, arXiv preprint arXiv:1511.06324.

[21] F. Lin, J. Chen, Learning low-complexity autoregressive models via proximal alternating minimization, arXiv preprint arXiv:1609.05341.