SG-PALM: a Fast Physically Interpretable Tensor Graphical Model

Yu Wang [1] **Alfred Hero** [1]

Abstract

We propose a new graphical model inference procedure, called SG-PALM, for learning conditional dependency structure of high-dimensional tensor-variate data. Unlike most other tensor graphical models the proposed model is interpretable and computationally scalable to high dimension. Physical interpretability follows from the Sylvester generative (SG) model on which SG-PALM is based: the model is exact for any observation process that is a solution of a partial differential equation of Poisson type. Scalability follows from the fast proximal alternating linearized minimization (PALM) procedure that SG-PALM uses during training. We establish that SG-PALM converges linearly (i.e., geometric convergence rate) to a global optimum of its objective function. We demonstrate the scalability and accuracy of SG-PALM for an important but challenging climate prediction problem: spatio-temporal forecasting of solar flares from multimodal imaging data.

1. Introduction

High-dimensional tensor-variate data arise in computer vision (video data containing multiple frames of color images), neuroscience (EEG measurements taken from different sensors over time under various experimental conditions), and recommending system (user preferences over time). Due to the non-homogeneous nature of these data, second-order information that encodes (conditional) dependency structure within the data is of interest. Assuming the data are drawn from a tensor normal distribution, a straightforward way to estimate this structure is to vectorize the tensor and estimate the underlying Gaussian graphical model associated with the vector. However, such an approach ignores the tensor structure and requires estimating a rather high dimensional precision matrix, often with insufficient sample size. For instance, in the aforementioned EEG application the sample size is one if we aim to estimate the dependency structure across different sensors, time and experimental conditions for a single subject. To address such sample complexity challenges, sparsity is often imposed on the covariance $\mathbf{\Sigma}$ or the inverse covariance $\mathbf{\Omega}$, e.g., by using a sparse Kronecker product (KP) or Kronecker sum (KS) decomposition of $\mathbf{\Sigma}$ or $\mathbf{\Omega}$. The earliest and most popular form of sparse structured precision matrix estimation approaches represent $\mathbf{\Omega}$, equivalently $\mathbf{\Sigma}$, as the KP of smaller precision/covariance matrices (Allen & Tibshirani, 2010; Leng & Tang, 2012; Yin & Li, 2012; Tsiligkaridis et al., 2013; Zhou, 2014; Lyu et al., 2019). The KP structure induces a generative representation for the tensor-variate data via a separable covariance/inverse covariance model. Alternatively, Kalaitzis et al. (2013); Greenewald et al. (2019) proposed to model inverse covariance matrices using a KS representation. Rudelson & Zhou (2017); Park et al. (2017) proposed KS-structured covariance model which corresponds to an errors-in-variables model. The KS (inverse) covariance structure corresponds to the Cartesian product of graphs (Kalaitzis et al., 2013; Greenewald et al., 2019), which leads to more parsimonious representations of (conditional) dependency than the KP. However, unlike the KP model, KS lacks an interpretable generative representation for the data. Recently, Wang et al. (2020) proposed a new class of structured graphical models, called the Sylvester graphical models, for tensor-variate data. The resulting inverse covariance matrix has the KS structure in its square-root factors. This square-root KS structure is hinted in the paper to have a connection with certain physical processes, but no illustration is provided.

A common challenge for structured tensor graphical models is the efficient estimation of the underlying (conditional) dependency structures. KP-structured models are generally estimated via extension of GLasso (Friedman et al., 2008) that iteratively minimize the ℓ_1-penalized negative likelihood function for the matrix-normal data with KP covariance. This procedure was shown to converge to some local optimum of the penalized likelihood function (Yin & Li, 2012; Tsiligkaridis et al., 2013). Similarly, Kalaitzis et al. (2013) further extended GLasso to the KS-structured case for 2-way tensor data. Greenewald et al. (2019) extended this to multiway tensors, exploiting the linearity of the space of KS-structured matrices and developing a projected proximal gradient algorithm for KS-structured inverse covari-

[1] University of Michigan, Ann Arbor, Michigan, USA. Correspondence to: Yu Wang <wayneyw@umich.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

ance matrix estimation, which achieves linear convergence (i.e., geometric convergence rate) to the global optimum. In Wang et al. (2020), the Sylvester-structured graphical model is estimated via a nodewise regression approach inspired by algorithms for estimating a class of vector-variate graphical models (Meinshausen et al., 2006; Khare et al., 2015). However, no theoretical convergence result for the algorithm was established nor did they study the computational efficiency of the algorithm.

In the modern era of big data, both computational and statistical learning accuracy are required of algorithms. Furthermore, when the objective is to learn representations for physical processes, interpretablility is crucial. In this paper, we bridge this "Statistical-to-Computational-to-Interpretable gap" for Sylvester graphical models. We develop a simple yet powerful first-order optimization method, based on the Proximal Alternating Linearized Minimization (PALM) algorithm, for recovering the conditional dependency structure of such models. Moreover, we provide the link between the Sylvester graphical models and physical processes obeying differential equations and illustrate the link with a real-data example. The following are our principal contributions:

1. A fast algorithm that efficiently recovers the generating factors of a representation for high-dimensional multiway data, significantly improving on Wang et al. (2020).
2. A comprehensive convergence analysis showing linear convergence of the objective function to its global optimum and providing insights for choices of hyperparameters.
3. A novel application of the algorithm to an important multi-modal solar flare prediction problem from solar magnetic field sequences. For such problems, SG-PALM is physically interpretable in terms of the partial differential equations governing solar activities proposed by heliophysicists.

2. Background and Notation

2.1. Notations

In this paper, scalar, vector and matrix quantities are denoted by lowercase letters, boldface lowercase letters and boldface capital letters, respectively. For a matrix $\mathbf{A} = (\mathbf{A}_{i,j}) \in \mathbb{R}^{d \times d}$, we denote $\|\mathbf{A}\|_2, \|\mathbf{A}\|_F$ as its spectral and Frobenius norm, respectively. We define $\|\mathbf{A}\|_{1,\text{off}} := \sum_{i \neq j} |\mathbf{A}_{i,j}|$ as its off-diagonal ℓ_1 norm. For tensor algebra, we adopt the notations used by Kolda & Bader (2009). A K-th order tensor is denoted by boldface Euler script letters, e.g, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$. The $(i_1, \ldots, i_K)$-th element of $\boldsymbol{\mathcal{X}}$ is denoted by $\boldsymbol{\mathcal{X}}_{i_1,\ldots,i_K}$, and the vectorization of $\boldsymbol{\mathcal{X}}$ is the d-dimensional vector $\operatorname{vec}(\boldsymbol{\mathcal{X}}) := (\boldsymbol{\mathcal{X}}_{1,1,\ldots,1}, \boldsymbol{\mathcal{X}}_{2,1,\ldots,1}, \ldots, \boldsymbol{\mathcal{X}}_{d_1,1,\ldots,1}, \ldots, \boldsymbol{\mathcal{X}}_{d_1,d_2,\ldots,d_k})^T$ with $d = \prod_{k=1}^K d_k$. A fiber is the higher order analogue of the row and column of matrices. It is obtained by fixing all but one of the indices of the tensor. Matricization, also known as unfolding, is the process of transforming a tensor into a matrix. The mode-k matricization of a tensor $\boldsymbol{\mathcal{X}}$, denoted by $\boldsymbol{\mathcal{X}}_{(k)}$, arranges the mode-$k$ fibers to be the columns of the resulting matrix. The k-mode product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times d_k}$, denoted as $\boldsymbol{\mathcal{X}} \times_k \mathbf{A}$, is of size $d_1 \times \cdots \times d_{k-1} \times J \times d_{k+1} \times \ldots d_K$. Its entry is defined as $(\boldsymbol{\mathcal{X}} \times_k \mathbf{A})_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_K} := \sum_{i_k=1}^{d_k} \boldsymbol{\mathcal{X}}_{i_1,\ldots,i_K} A_{j,i_k}$. For a list of matrices $\{\mathbf{A}_k\}_{k=1}^K$ with $\mathbf{A}_k \in \mathbb{R}^{d_k \times d_k}$, we define $\boldsymbol{\mathcal{X}} \times \{\mathbf{A}_1, \ldots, \mathbf{A}_K\} := \boldsymbol{\mathcal{X}} \times_1 \mathbf{A}_1 \times_2 \cdots \times_K \mathbf{A}_K$. Lastly, we define the K-way Kronecker product as $\bigotimes_{k=1}^K \mathbf{A}_k = \mathbf{A}_1 \otimes \cdots \otimes \mathbf{A}_K$, and the equivalent notation for the Kronecker sum as $\bigoplus_{k=1}^K \mathbf{A}_k = \mathbf{A}_1 \oplus \cdots \oplus \mathbf{A}_K = \sum_{k=1}^K \mathbf{I}_{[d_{k+1:K}]} \otimes \mathbf{A}_k \otimes \mathbf{I}_{[d_{1:k-1}]}$, where $\mathbf{I}_{[d_{k:\ell}]} = \mathbf{I}_{d_k} \otimes \cdots \otimes \mathbf{I}_{d_\ell}$. For the case of $K = 2$, $\mathbf{A}_1 \oplus \mathbf{A}_2 = \mathbf{I}_{d_2} \otimes \mathbf{A}_1 + \mathbf{A}_2 \otimes \mathbf{I}_{d_1}$.

2.2. Tensor Graphical Models

A random tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ follows the tensor normal distribution with zero mean when $\operatorname{vec}(\boldsymbol{\mathcal{X}})$ follows a normal distribution with mean $\mathbf{0} \in \mathbb{R}^d$ and precision matrix $\mathbf{\Omega} := \mathbf{\Omega}(\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_K)$, where $d = \prod_{k=1}^K d_k$. Here, $\mathbf{\Omega}(\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_K)$ is parameterized by $\mathbf{\Psi}_k \in \mathbb{R}^{d_k \times d_k}$ via either Kronecker product, Kronecker sum, or the Sylvester structure, and the corresponding negative log-likelihood function (assuming N independent observations $\boldsymbol{\mathcal{X}}^i, i = 1, \ldots, N$)

$$-\frac{N}{2} \log |\mathbf{\Omega}| + \frac{N}{2} \operatorname{tr}(\mathbf{S}\mathbf{\Omega}), \tag{1}$$

where $\mathbf{\Omega} = \bigotimes_{k=1}^K \mathbf{\Psi}_k$, $\bigoplus_{k=1}^K \mathbf{\Psi}_k$, or $\left(\bigoplus_{k=1}^K \mathbf{\Psi}_k\right)^2$ for KP, KS, and Sylvester models, respectively; and $\mathbf{S} = \frac{1}{N} \sum_{i=1}^N \operatorname{vec}(\boldsymbol{\mathcal{X}}^i) \operatorname{vec}(\boldsymbol{\mathcal{X}}^i)^T$. To encourage sparsity, penalized negative log-likelihood function is proposed

$$-\frac{N}{2} \log |\mathbf{\Omega}| + \frac{N}{2} \operatorname{tr}(\mathbf{S}\mathbf{\Omega}) + \sum_{k=1}^K P_{\lambda_k}(\mathbf{\Psi}_k),$$

where $P_{\lambda_k}(\cdot)$ is a penalty function indexed by the tuning parameter λ_k and is applied elementwise to the off-diagonal elements of $\mathbf{\Psi}_k$. Popular choices for $P_{\lambda_k}(\cdot)$ include the lasso penalty (Tibshirani, 1996), the adaptive lasso penalty (Zou, 2006), the SCAD penalty (Fan & Li, 2001), and the MCP penalty (Zhang et al., 2010).

2.3. The Sylvester Generating Equation

Wang et al. (2020) proposed a Sylvester graphical model that uses the Sylvester tensor equation to define a generative process for the underlying multivariate tensor data. The Sylvester tensor equation has been studied in the context of finite-difference discretization of high-dimensional elliptical partial differential equations (Grasedyck, 2004; Kressner & Tobler, 2010). Any solution $\boldsymbol{\mathcal{X}}$ to such a PDE must have the (discretized) form:

$$\sum_{k=1}^{K} \boldsymbol{\mathcal{X}} \times_k \boldsymbol{\Psi}_k = \boldsymbol{\mathcal{T}} \Longleftrightarrow \Big(\bigoplus_{k=1}^{K} \boldsymbol{\Psi}_k\Big) \operatorname{vec}(\boldsymbol{\mathcal{X}}) = \operatorname{vec}(\boldsymbol{\mathcal{T}}). \tag{2}$$

where $\boldsymbol{\mathcal{T}}$ is the driving source on the domain, and $\bigoplus_{k=1}^{K} \boldsymbol{\Psi}_k$ is a Kronecker sum of $\boldsymbol{\Psi}_k$'s representing the discretized differential operators for the PDE, e.g., Laplacian, Euler-Lagrange operators, and associated coefficients. These operators are often sparse and structured.

For example, consider a physical process characterized as a function u that satisfies:

$$\mathcal{D}u = f \quad \text{in} \quad \Omega, \quad u(\Gamma) = 0, \quad \Gamma = \partial\Omega.$$

where f is a driving process, e.g., a Wiener process (white Gaussian noise); $\mathcal{D}$ is a differential operator, e.g, Laplacian, Euler-Lagrange; Ω is the domain; and Γ is the boundary of Ω. After discretization, this is equivalent to (ignoring discretization error) the matrix equation

$$\mathbf{D}\mathbf{u} = \mathbf{f}.$$

Here, $\mathbf{D}$ is a sparse matrix since $\mathcal{D}$ is an infinitesimal operator. Additionally, $\mathbf{D}$ admits Kronecker structure as a mixture of Kronecker sums and Kronecker products.

The matrix $\mathbf{D}$ reduces to a Kronecker sum when $\mathcal{D}$ involves no mixed derivatives. For instance, consider the Poisson equation in 2D, where $u(x, y)$ on $[0, 1]^2$ satisfies the elliptical PDE

$$\mathcal{D}u = (\partial_x^2 + \partial_y^2)u = f.$$

The Poisson equation governs many physical processes, e.g., electromagnetic induction, heat transfer, convection, etc. A simple Euler discretization yields $\mathbf{U} = (u(i, j))_{i,j}$, where $u(i, j)$ satisfies the local equation (up to a constant discretization scale factor)

$$\begin{aligned} 4u(i,j) = {} & u(i+1,j) + u(i-1,j) + u(i,j+1) \\ & + u(i,j-1) - f(i,j). \end{aligned}$$

Defining $\mathbf{u} = \operatorname{vec}(\mathbf{U})$ and $\mathbf{A}$ (a tridiagonal matrix)

$$\mathbf{A} = \begin{bmatrix} -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{bmatrix},$$

then $(\mathbf{A} \oplus \mathbf{A})\mathbf{u} = \mathbf{f}$, which is the Sylvester equation ($K = 2$).

For the Poisson example, if the source $\mathbf{f}$ is a white noise random variable, i.e., its covariance matrix is proportional to the identity matrix, then the inverse covariance matrix of $\mathbf{u}$ has sparse square-root factors, since $\operatorname{Cov}^{-1}(\mathbf{u}) = (\mathbf{A} \oplus \mathbf{A})(\mathbf{A} \oplus \mathbf{A})^T$. Other physical processes that are generated from differential equations will also have sparse inverse covariance matrices, as a result of the sparsity of general discretized differential operators. Note that similar connections between continuous state physical processes and sparse "discretized" statistical models have been established by Lindgren et al. (2011), who elucidated a link between Gaussian fields and Gaussian Markov Random Fields via stochastic partial differential equations.

The Sylvester generative (SG) model (2) leads to a tensor-valued random variable $\boldsymbol{\mathcal{X}}$ with a precision matrix $\boldsymbol{\Omega} = \Big(\bigoplus_{k=1}^{K} \boldsymbol{\Psi}_k\Big)^2$, given that $\boldsymbol{\mathcal{T}}$ is white Gaussian. The Sylvester generating factors $\boldsymbol{\Psi}_k$'s can be obtained via minimization of the penalized negative log-pseudolikelihood

$$\begin{aligned} \mathcal{L}_\lambda(\boldsymbol{\Psi}) = & -\frac{N}{2} \log |(\bigoplus_{k=1}^{K} \operatorname{diag}(\boldsymbol{\Psi}_k))^2| \\ & + \frac{N}{2} \operatorname{tr}(\mathbf{S} \cdot (\bigoplus_{k=1}^{K} \boldsymbol{\Psi}_k)^2) + \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\Psi}_k\|_{1,\text{off}}. \end{aligned} \tag{3}$$

This differs from the true penalized Gaussian negative log-likelihood in the exclusion of off-diagonals of $\boldsymbol{\Psi}_k$'s in the log-determinant term. (3) is motivated and derived directly using the Sylvester equation defined in (2), from the perspective of solving a sparse linear system. This maximum pseudolikelihood estimation procedure has been applied to vector-variate Gaussian graphical models (see Khare et al. (2015) and references therein). Detailed derivations and further discussions are provided in Appendix A.

3. The SG-PALM Algorithm

Estimation of the generating parameters $\boldsymbol{\Psi}_k$'s of the SG model is challenging since the sparsity penalty applies to the square root factors of the precision matrix, which leads to a highly coupled likelihood function. Wang et al. (2020) proposed an estimation procedure called SyGlasso, that recovers only the off-diagonal elements of each Sylvester factor. This is a deficiency in many applications where the factor-wise variances are desired. Moreover, the convergence rate of the cyclic coordinate-wise algorithm used in SyGlasso is unknown and the computational complexity of the algorithm is higher than other sparse Glasso-type procedures. To overcome these deficiencies, we propose a proximal alternating linearized minimization method that is more flexible and versatile, called SG-PALM, for finding the minimizer of (3). SG-PALM is designed to exploit structures of the coupled objective function and yields simultaneous estimates for both off-diagonal and diagonal entries.

The PALM algorithm was originally proposed to solve nonconvex optimization problems with separable structures, such as those arising in nonnegative matrix factorization (Xu & Yin, 2013; Bolte et al., 2014). Its efficacy in solving convex problems has also been established, for example, in regularized linear regression problems (Shefi & Teboulle, 2016), it was proposed as an attractive alternative to iterative soft-thresholding algorithms (ISTA). The SG-PALM procedure is summarized in Algorithm 1.

For clarity of notation we write

$$\mathcal{L}_\lambda(\mathbf{\Psi}_1, \dots, \mathbf{\Psi}_K) = H(\mathbf{\Psi}_1, \dots, \mathbf{\Psi}_K) + \sum_{k=1}^{K} G_k(\mathbf{\Psi}_k), \tag{4}$$

where $H : \mathbb{R}^{d_1 \times d_1} \times \cdots \times \mathbb{R}^{d_K \times d_K} \to \mathbb{R}$ represents the log-determinant plus trace terms in (3) and $G_k : \mathbb{R}^{d_k \times d_k} \to (-\infty, +\infty]$ represents the penalty term in (3) for each axis $k = 1, \dots, K$. For notational simplicity we use $\mathbf{\Psi}$ (i.e., omitting the subscript) to denote the set $\{\mathbf{\Psi}_k\}_{k=1}^K$ or the K-tuple $(\mathbf{\Psi}_1, \dots, \mathbf{\Psi}_K)$ whenever there is no risk of confusion. The gradient of the smooth function H with respect to $\mathbf{\Psi}_k$, $\nabla_k H(\mathbf{\Psi})$, is given by

$$\begin{aligned} &\operatorname{diag}\left(\left\{ \operatorname{tr}[(\operatorname{diag}((\mathbf{\Psi}_k)_{ii}) + \bigoplus_{j \neq k} \operatorname{diag}(\mathbf{\Psi}_j))^{-1}] \right\}_{i=1}^{d_k}\right) \\ &\quad + \mathbf{S}_k \mathbf{\Psi}_k + \mathbf{\Psi}_k \mathbf{S}_k + 2 \sum_{j \neq k} \mathbf{S}_{j,k}. \end{aligned} \tag{5}$$

Here, the first "diag" maps a d_k-vector to a $d_k \times d_k$ diagonal matrix, the second one maps a scalar (i.e., $(\mathbf{\Psi}_k)_{ii}$) to a $(\prod_{j \neq k} d_j) \times (\prod_{j \neq k} d_j)$ diagonal matrix with the same elements, and the third operator maps a symmetric matrix to a matrix containing only its diagonal elements. In addition, we define:

$$\begin{aligned} \mathbf{S}_k &= \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\mathcal{X}}^i_{(k)} (\boldsymbol{\mathcal{X}}^i_{(k)})^T, \\ \mathbf{S}_{j,k} &= \frac{1}{N} \sum_{i=1}^{N} \mathbf{V}^i_{j,k} (\mathbf{V}^i_{j,k})^T, \\ \mathbf{V}^i_{j,k} &= \boldsymbol{\mathcal{X}}^i_{(k)} \Big(\mathbf{I}_{d_{1:j-1}} \otimes \mathbf{\Psi}_j \otimes \mathbf{I}_{d_{j:K}} \Big)^T, \quad j \neq k. \end{aligned} \tag{6}$$

A key ingredient of the PALM algorithm is a proximal operator associated with the non-smooth part of the objective, i.e., G_k's. In general, the proximal operator of a proper, lower semi-continuous convex function f from a Hilbert space $\mathcal{H}$ to the extended reals $(-\infty, +\infty]$ is defined by (Parikh & Boyd, 2014)

$$\operatorname{prox}_f(v) = \operatorname*{argmin}_{x \in \mathcal{H}} f(x) + \frac{1}{2} \|x - v\|_2^2$$

for any $v \in \mathcal{H}$. The proximal operator well-defined as the expression on the right-hand side above has a unique minimizer for any function in this class. For ℓ_1-regularized cases, the proximal operator for the function G_k is given by

$$\operatorname{prox}^{\lambda_k}_{G_k}(\mathbf{\Psi}_k) = \operatorname{diag}(\mathbf{\Psi}_k) + \operatorname{soft}(\mathbf{\Psi}_k - \operatorname{diag}(\mathbf{\Psi}_k), \lambda_k), \tag{7}$$

where the soft-thresholding operator $\operatorname{soft}_\lambda(x) = \operatorname{sign}(x) \max(|x| - \lambda, 0)$ has been applied elementwise. For popular choices of non-convex G_k, the proximal operators are derived in Appendix D.

Algorithm 1 SG-PALM

```
Input: Data tensor $\boldsymbol{\mathcal{X}}$, mode-$k$ Gram matrix $\mathbf{S}_k$, regularizing parameter $\lambda_k$, backtracking constant $c \in (0,1)$, initial step size $\eta_0$, initial iterate $\mathbf{\Psi}_k$ for each $k = 1, \dots, K$.
  while not converged do
    for $k = 1, \dots, K$ do
      Line search:
      Let $\eta^t_k$ be the largest element of $\{c^j \eta^t_{k,0}\}_{j=1,\dots}$ such that condition (8) is satisfied.
      Update:
      $\mathbf{\Psi}^{t+1}_k \leftarrow \operatorname{prox}^{G_k}_{\eta^t_k \lambda_k} \Big( \mathbf{\Psi}^t_k - \eta^t_k \nabla_k H(\mathbf{\Psi}^{t+1}_{i<k}, \mathbf{\Psi}^t_{i \geq k}) \Big)$.
    end for
    Update initial step size: Compute Barzilai-Borwein step size $\eta^{t+1}_0 = \min_k \eta^{t+1}_{k,0}$, where $\eta^{t+1}_{k,0}$ is computed via (9).
  end while
Output: Final iterates $\{\mathbf{\Psi}_k\}_{k=1}^K$.
```

3.1. Choice of Step Size

In the absence of a good estimate of the blockwise Lipchitz constant, the step size of each iteration of SG-PALM is chosen using backtracking line search, which, at iteration t, starts with an initial step size η^t_0 and reduces the size with a constant factor $c \in (0,1)$ until the new iterate satisfies the sufficient descent condition:

$$H(\mathbf{\Psi}^{t+1}_{i \leq k}, \mathbf{\Psi}^t_{i > k}) \leq Q_{\eta^t}(\mathbf{\Psi}^{t+1}_{i \leq k}, \mathbf{\Psi}^t_{i > k}; \mathbf{\Psi}^{t+1}_{i < k}, \mathbf{\Psi}^t_{i \geq k}). \tag{8}$$

Here,

$$\begin{aligned} &Q_\eta(\mathbf{\Psi}_{i<k}, \mathbf{\Psi}_k, \mathbf{\Psi}_{i>k}; \mathbf{\Psi}_{i<k}, \mathbf{\Psi}'_k, \mathbf{\Psi}_{i>k}) \\ &\quad = H(\mathbf{\Psi}_{i<k}, \mathbf{\Psi}_k, \mathbf{\Psi}_{i>k}) \\ &\quad + \operatorname{tr}\Big((\mathbf{\Psi}'_k - \mathbf{\Psi}_k)^T \nabla_k H(\mathbf{\Psi}_{i<k}, \mathbf{\Psi}_k, \mathbf{\Psi}_{i>k}) \Big) \\ &\quad + \frac{1}{2\eta} \|\mathbf{\Psi}'_k - \mathbf{\Psi}_k\|_F^2. \end{aligned}$$

The sufficient descent condition is satisfied with any $\frac{1}{\eta} = M_k$ and $M_k \geq L_k$, for any function that has a block-wise Lipschitz gradient with constant L_k for $k = 1, \dots, K$. In other words, so long as the function H has block-wise gradient that is Lipschitz continuous with some block Lipschitz constant $L_k > 0$ for each k, then at each iteration t, we can always find an η^t such that the inequality in (8) is satisfied. Indeed, we proved in Lemma C.1 in the Appendix that H has the desired properties. Additionally, in the proof of Theorem 4.2 we also showed that the step size found at each iteration t satisfies $\frac{1}{\eta_k^0} \leq L_k \leq \frac{1}{\eta_k^t} \leq cL_k$.

In terms of the initialization, a safe step size (i.e., very small η_0^t) often leads to slower convergence. Thus, we use the more aggressive Barzilai-Borwein (BB) step (Barzilai & Borwein, 1988) to set a starting η_0^t at each iteration (see Appendix B for justifications of the BB method). In our case, for each k, the step size is given by

$$\eta_{k,0}^t = \frac{\|\mathbf{\Psi}_k^{t+1} - \mathbf{\Psi}_k^t\|_F^2}{\mathrm{tr}(\mathbf{A})}, \tag{9}$$

where

$$\mathbf{A} = (\mathbf{\Psi}_k^{t+1} - \mathbf{\Psi}_k^t)^T \times \\ (\nabla_k H(\mathbf{\Psi}_{i\leq k}^{t+1}, \mathbf{\Psi}_{i>k}^t) - \nabla_k H(\mathbf{\Psi}_{i<k}^{t+1}, \mathbf{\Psi}_{i\geq k}^t)).$$

3.2. Computational Complexity

After pre-computing $\mathbf{S}_k$, the most significant computation for each iteration in the SG-PALM algorithm is the sparse matrix-matrix multiplications $\mathbf{S}_k \mathbf{\Psi}_k$ and $\mathbf{S}_{j,k}$ in the gradient calculation. In terms of computational complexity, if s_k is the number of non-zeros per column in $\mathbf{\Psi}_k$, then the former and latter can be computed using $O(s_k d_k^2)$ and $O(N\sum_{j\neq k} s_j d_j^2)$ operations, respectively. Thus, each iteration of SG-PALM can be computed using $O\Big(\sum_{k=1}^K (s_k d_k^2 + N\sum_{j\neq k} s_j d_j^2)\Big)$ floating point operations, which is significantly lower than competing methods.

For instance, other popular algorithms for tensor-variate graphical models, such as the TG-ISTA presented in Greenewald et al. (2019) and the Tlasso proposed in Lyu et al. (2019) both require inversion of $d_k \times d_k$ matrices, which is non-parallelizable and requires $O(d_k^3)$ operations for each k. In particular, TeraLasso's TG-ISTA algorithm requires $O\Big(Kd + \sum_{k=1}^K d_k^3\Big)$ operations. The TG-ISTA algorithm requires matrix inversions that cannot easily exploit the sparsity of $\mathbf{\Psi}_k$'s. In the sample-starved ultra-sparse setting ($N \ll d$ and $s_k \ll d_k$), the $O(N\sum_{j\neq k} s_j d_j^2)$ terms in SG-PALM are comparable to $O(Kd)$ in TG-ISTA, making it more appealing. The cyclic coordinate-wise method proposed in (Wang et al., 2020) does not allow for parallelization since it requires cycling through entries of each $\mathbf{\Psi}_k$ in specified order. In contrast, SG-PALM can be implemented in parallel to distribute the sparse matrix-matrix multiplications because at no step do the algorithms require storing all dense matrices on a single machine.

4. Convergence Analysis

In this section, we present the main convergence theorems. Detailed proofs are included in the supplement. Here, we study the statistical convergence behavior for the Sylvester graphical model with an ℓ_1 penalty function. The convergence behavior of the SG-PALM iterates is presented for convex cases but similar convergence rate can be established for non-convex penalties (see Appendix D).

We first establish statistical convergence of a global minimizer $\hat{\mathbf{\Psi}}$ of (3) to its true value, denoted as $\bar{\mathbf{\Psi}}$, under the correct statistical model.

Theorem 4.1. *Let $\mathcal{A}_k := \{(i,j) : (\bar{\mathbf{\Psi}}_k)_{i,j} \neq 0, i \neq j\}$ and $q_k := |\mathcal{A}_k|$ for $k = 1, \ldots, K$. If $N > O(\max_k q_k d_k \log d)$ and $d := d_N = O(N^\kappa)$ for some $\kappa \geq 0$, and further, if the penalty parameter satisfies $\lambda_k := \lambda_{N,k} = O(\sqrt{\frac{d_k \log d}{N}})$ for all $k = 1, \ldots, K$, then under conditions (A1-A3) in Appendix C.1, there exists a constant $C > 0$ such that for any $\eta > 0$ the following events hold with probability at least $1 - O(\exp(-\eta \log d))$:*

$$\sum_{k=1}^K \|\textit{offdiag}(\hat{\mathbf{\Psi}}_k) - \textit{offdiag}(\bar{\mathbf{\Psi}}_k)\|_F \\ \leq C\sqrt{K} \max_k \sqrt{q_k}\lambda_k.$$

Here offdiag($\mathbf{\Psi}_k$) contains only the off-diagonal elements of $\mathbf{\Psi}_k$. If further $\min_{(i,j)\in\mathcal{A}_k} |(\bar{\mathbf{\Psi}}_k)_{i,j}| \geq 2C \max_k \sqrt{q_k}\lambda_k$ for each k, then sign($\hat{\mathbf{\Psi}}_k$)=sign($\bar{\mathbf{\Psi}}_k$).

Theorem 4.1 means that under regularity conditions on the true generative model, and with appropriately chosen penalty parameters λ_k's guided by the theorem, one is guaranteed to recover the true structures of the underlying Sylvester generating parameters $\mathbf{\Psi}_k$ for $k = 1, \ldots, K$ with probability one, as the sample size and dimension grow.

We next turn to convergence of the iterates $\{\mathbf{\Psi}^t\}$ from SG-PALM to a global optimum of (3).

Theorem 4.2. *The Let $\{\mathbf{\Psi}^{(t)}\}_{t\geq 0}$ be generated by SG-PALM. Then, SG-PALM converges in the sense that*

$$\frac{\mathcal{L}_\lambda(\mathbf{\Psi}^{(t+1)}) - \min \mathcal{L}_\lambda}{\mathcal{L}_\lambda(\mathbf{\Psi}^{(t)}) - \min \mathcal{L}_\lambda} \\ \leq \left(\frac{\alpha^2 L_{\min}}{4Kc^2(\sum_{j=1}^K L_j)^2 + 4c^2 L_{\max}} + 1\right)^{-1},$$

where α, $L_k, k = 1, \ldots, K$ are positive constants, $L_{\min} = \min_j L_j$, $L_{\max} = \max_j L_j$, and $c \in (0,1)$ is the backtracking constant defined in Algorithm 1.

Note that the term on the right hand side of the inequality above is strictly less than 1. This means that the SG-PALM algorithm converges linearly, which is a strong results for a non-strongly convex objective (i.e., $\mathcal{L}_\lambda$). Although similar convergence behaviors of the PALM-type algorithms have been studied for other problems (Xu & Yin, 2013; Bolte et al., 2014), such as nonnegative matrix/tensor factorization, the analysis of this paper works for non-strongly block multi-convex objectives, leveraging more recent analyses of multi-block PALM and a class of functions satisfying the the Kurdyka - Łojasiewicz (KL) property (defined in Section C of the Appendix). To the best of our knowledge, for first-order optimization methods, our rate is faster than any other Gaussian graphical models having non-strongly convex objectives (see Khare et al. (2015); Oh et al. (2014) and references therein) and comparable with those having strongly-convex objectives (see, for example, Guillot et al. (2012); Dalal & Rajaratnam (2017); Greenewald et al. (2019)).

5. Experiments

Experiments in this section were performed in a system with `8-core Intel Xeon CPU E5-2687W v2 3.40GHz` equipped with `64GB RAM`. Both SG-PALM and SyGlasso were implemented in `Julia v1.5` (`https://github.com/ywa136/sg-palm`). For real data analyses, we used the `Tlasso` package implementation in `R` (Sun et al., 2016) and the TeraLasso implementation in `MATLAB` (`https://github.com/kgreenewald/teralasso`).

5.1. Synthetic Data

We first validate the convergence theorems discussed in the previous section via simulation studies. Synthetic datasets were generated from true sparse Sylvester factors $\{\mathbf{\Psi}_k\}_{k=1}^K$ where $K = \{2, 3\}$ and $d_k = \{16, 32, 64, 128\}$ for all k. Instances of the random matrices used here have uniformly random sparsity patterns with edge densities (i.e., the proportion of non-zero entries) ranging from $0.1\% - 30\%$ on average over all $\mathbf{\Psi}_k$'s. For each d and edge density combination, random samples of size $N = \{10, 100, 1000\}$ were tested. For comparison, the initial iterates, convergence criteria were matched between SyGlasso and SG-PALM. Highlights of the results in run times are summarized in Table 1.

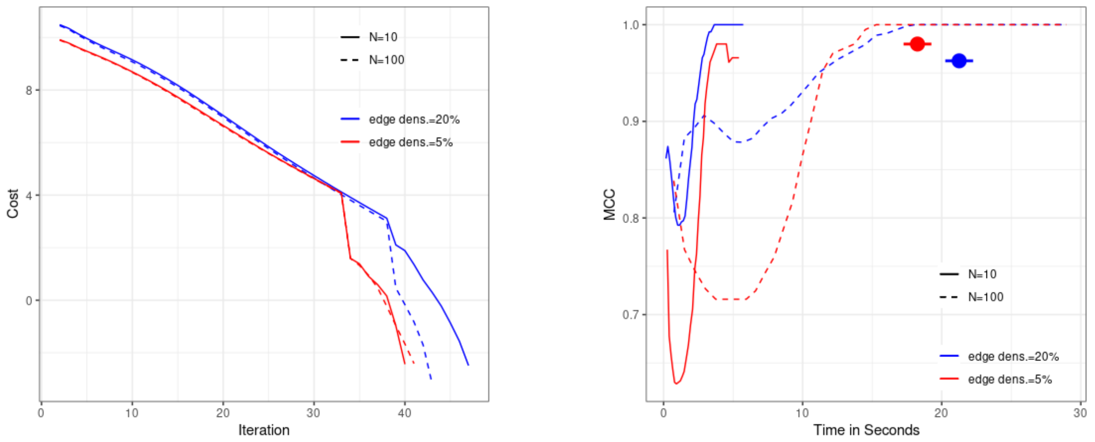
Convergence behavior of SG-PALM is shown in Figure 1 (a) for the datasets with $d_k = 32$, $N = \{10, 100\}$, and edge densities roughly around 5% and 20%, respectively. Geometric convergence rate of the function value gaps under Theorem 4.2 can be verified from the plot. Note an acceleration in the convergence rate (i.e., a steeper slope) near the optimum, which is suggested by the "localness" of the KL property of the objective function close to its global optimum. Further for the same datasets, in Figure 1 (b), SG-PALM graph recovery performances is illustrated, where the Matthew's Correlation Coefficients (MCC) is plotted against run time. Here, MCC is defined by

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}},$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives of the estimated edges (i.e., non-zero elements of $\mathbf{\Psi}_k$'s). An MCC of 1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation. The results validate the statistical accuracy under Theorem 4.1. It also shows that SG-PALM outperforms SyGlasso (indicated by blue/red solid dots) within the same time budget.

Table 1. Run time comparisons (in seconds with N/As indicating those exceeding 24 hour) between SyGlasso and SG-PALM on synthetic datasets with different dimensions, sample sizes, and densities of the generating Sylvester factors. Note that the proposed SG-PALM has average speed-up ratios ranging from 1.5 to 10 over SyGlasso.

d	N	NZ%	**SyGlasso**		**SG-PALM**	
			iter	**sec**	**iter**	**sec**
128^2	10^1	1.20	17	138.5	46	5.8
		24.0	20	169.3	48	6.2
	10^2	1.30	21	211.3	50	12.6
		27.0	30	303.6	47	21.9
	10^3	1.30	21	2045.8	50	80.1
		25.0	47	4782.7	51	373.1
16^3	10^1	0.11	9	4.6	11	4.5
		4.10	9	5.1	32	5.1
	10^2	0.21	8	8.8	11	5.4
		2.60	8	10.8	35	7.2
	10^3	0.26	8	82.4	12	14.3
		3.40	10	99.2	37	33.5
32^3	10^1	0.13	10	191.2	19	7.3
		7.50	17	304.8	42	10.2
	10^2	0.46	9	222.4	24	28.9
		7.00	17	395.2	41	48.5
	10^3	0.10	9	1764.8	22	226.4
		6.90	19	3789.4	41	473.9
64^3	10^1	0.65	10	583.7	42	91.3
		14.5	22	952.2	47	119.0
	10^2	0.62	9	6683.7	41	713.9
		14.4	21	15607.2	48	1450.9
	10^3	0.85		N/A	39	6984.4
		14.0		N/A	48	12968.7



(a) Cost gap vs. Iteration

(b) MCC vs. Run time

Figure 1. Convergence of SG-PALM algorithm under datasets with varying sample sizes (solid and dashed) generated via matrices with different sparsity (red and blue). The function value gaps on log-scale (left) verifies the geometric convergence rate in all cases and the MCC over time (right) demonstrates the algorithm's accuracy and efficiency. Note that the SG-PALM reached almost perfect recoveries (i.e., MCC of 1) within 20 seconds in all cases. In comparison, SyGlasso (big solid dots with line-range) was only able to achieve at lower MCCs for lower sample-size cases within 30 seconds.

5.2. Solar Flare Imaging Data

A solar flare occurs when magnetic energy that has built up in the solar atmosphere is suddenly released. Such events strongly influence space weather near the Earth. Therefore, reliable predictions of these flaring events are of great interest. Recent work (Chen et al., 2019; Jiao et al., 2019; Sun et al., 2019) has shown the promise of machine learning methods for early forecasting of these events using imaging data from the solar atmosphere. In this work, we illustrate the viability of the SG-PALM algorithm for solar flare prediction using data acquired by multiple instruments: the Solar Dynamics Observatory (SDO)/Helioseismic and Magnetic Imager (HMI) and SDO/Atmospheric Imaging Assembly (AIA). It is evident that these data contain information about the physical processes that govern solar activities (see Appendix E for detailed data descriptions).

The data samples are summarized in $d_1 \times d_2 \times d_3 \times d_4$ tensors with $q = d_1 \cdot d_2 \cdot d_3 = 50 \cdot 100 \cdot 7 = 35000$ and $p = d_4 = 13$. The first two modes represent the images' heights and widths, the third mode represents the HMI/AIA components/channels, and the last mode represents the length of the temporal window. Previous studies (Chen et al., 2019; Jiao et al., 2019) found that the time series of solar images from the SDO/HMI data provide useful information for distinguishing strong solar flares of M/X class from weak flares of A/B class roughly 24 to 12 hours prior to the flare event. Thus, in this study we use a 13-hour temporal window recorded with 1-hour cadence, prior to the occurrence of a solar flare. The task is to predict the pth frame using the frames in each of the $p-1$ previous hours (i.e., one hour ahead prediction). Each observation is a video with full dimension $d = pq$, and each p-dimensional observation vector is formed by concatenating the p time-consecutive q-dimensional vectors (vectorization of the matrices representing pixels of the multichannel images) without overlapping the time segments. The training set contains two types (B- and MX-class flares) of active regions producing flares. Each is distinguished by the flaring intensities, and there are a total of 186 B flares and 48 MX flares. Forward linear predictors were constructed using estimated precision matrices in a multi-output least squares regression setting. Specifically, we constructed the linear predictor of a frame from the $p-1$ previous frames in the same video:

$$\hat{\mathbf{y}}_t = -\mathbf{\Omega}_{2,2}^{-1}\mathbf{\Omega}_{2,1}\mathbf{y}_{t-1:t-(p-1)}, \qquad (10)$$

where $\mathbf{y}_{t-1:t-(p-1)} \in \mathbb{R}^{(p-1)q}$ is the stacked set of pixel values from the previous $p-1$ time instances and $\mathbf{\Omega}_{2,1} \in \mathbb{R}^{q\times(p-1)q}$ and $\mathbf{\Omega}_{2,2} \in \mathbb{R}^{q\times q}$ are submatrices of the $pq \times pq$ estimated precision matrix:

$$\hat{\mathbf{\Omega}} = \begin{pmatrix} \mathbf{\Omega}_{1,1} & \mathbf{\Omega}_{1,2} \\ \mathbf{\Omega}_{2,1} & \mathbf{\Omega}_{2,2} \end{pmatrix}.$$

The predictors were tested on the data containing flares observed from different active regions than those in training set, so that the predictor has never "seen" the frames that it attempts to predict, corresponding to 117 observations of which 93 are B-class flares and 24 are MX-class flares. Figure 2 shows the root mean squared error normalized by the difference between maximum and minimum pixels (NRMSE) over the testing samples, for the forecasts based on the SG-PALM estimator, TeraLasso estimator (Greenewald et al., 2019), Tlasso estimator (Lyu et al., 2019), and IndLasso estimator. Here, the TeraLasso and the Tlasso are estimation algorithms for a KS and a KP tensor precision matrix model, respectively; the IndLasso denotes an estimator obtained by applying independent and separate ℓ_1-penalized regressions to each pixel in $\mathbf{y}_t$. The SG-PALM estimator was implemented using a regularization parameter $\lambda_N = C_1\sqrt{\frac{\min(d_k)\log(d)}{N}}$ for all k with the constant C_1 chosen by optimizing the prediction NRMSE on the training set over a range of λ values parameterized by C_1. The TeraLasso estimator and the Tlasso estimator were implemented using $\lambda_{N,k} = C_2\sqrt{\frac{\log(d)}{N\prod_{i\neq k} d_i}}$ and $\lambda_{N,k} = C_3\sqrt{\frac{\log(d_k)}{Nd}}$ for $k = 1, 2, 3$, respectively, with C_2, C_3 optimized in a similar manner. Each sparse regression in the IndLasso estimator was implemented and tuned independently with regularization parameters chosen from a grid via cross-validation.

We observe that SG-PALM outperforms all three other methods, indicated by NRMSEs across pixels. Figure 3 depicts examples of predicted images, comparing with the ground truth. The SG-PALM estimates produced most realistic image predictions that capture the spatially varying structures and closely approximate the pixel values (i.e., maintaining contrast ratios). The latter is important as the flares are being classified into weak (B-class) and strong (MX-class) categories based on the brightness of the images, and stronger flares are more likely to lead to catastrophic events, such as those damaging spacecrafts. Lastly, we compare run times of the SG-PALM algorithm for estimating the precision matrix from the solar flare data with SyGlasso. Table 2 in Appendix E illustrates that the SG-PALM algorithm converges faster in wallclock time. Note that in this real dataset, which is potentially non-Gaussian, the convergence behavior of the algorithms is different compare to synthetic examples. Nonetheless, SG-PALM enjoys an order of magnitude speed-up over SyGlasso.

5.3. Physical Interpretability

To explain the advantages of the proposed model over other similar models (e.g., Tlasso, TeraLasso), we provide further discussions here on the connection between the Sylvester generating model and PDEs. Consider the 2D spatio-temporal process $u(\mathbf{x}, t)$:

$$\partial u/\partial t = \theta \sum_{i=1}^{2} \partial^2 u/\partial x_i^2 + \epsilon \sum_{i=1}^{2} \partial u/\partial x_i, \qquad (11)$$

where θ, ϵ are positive real (unknown) coefficients. This is the basic form of a class of parabolic and hyperbolic PDEs, the Convection-Diffusion equation that generalizes the Poisson equation presented in Section 2 by incorporating temporal evolution. These equations are closely related to the Navier-Stokes equation commonly used in stochastic modelling for weather and climate prediction. Coupled with Maxwell's equations, they can be used to model and study magneto-hydrodynamics (Roberts, 2006), which characterize solar activities including flares.

After finite-difference discretization, Equation (11) is equivalent to the Sylvester matrix equation $\mathbf{A}_{\theta,\epsilon}\mathbf{U}_t + \mathbf{U}_t\mathbf{A}_{\theta,\epsilon} = \mathbf{U}_{t-1}$, where $\mathbf{U}_t = (u((i,j),t))_{ij}$ and $\mathbf{A}_{\theta,\epsilon}$ is a tridiagonal matrix with values that depend on the coefficients θ, ϵ and discretization step sizes. Assuming a linear Gaussian state-space model for some observed process $\mathbf{X}_t$ governed by the Convection-Diffusion dynamics:

$$\begin{aligned} &\mathbf{A}_{\theta,\epsilon}\mathbf{U}_t + \mathbf{U}_t\mathbf{A}_{\theta,\epsilon} = \mathbf{U}_{t-1}, \\ &\mathbf{X}_t = \mathbf{U}_t + \mathbf{V}_t, \end{aligned}$$

where $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is some time-invariant white noise. Then the precision matrix of the true process $\mathbf{U}_t$ evolves as $\mathbf{\Omega}_t = (\mathbf{A}_{\theta,\epsilon} \oplus \mathbf{A}_{\theta,\epsilon})\mathbf{\Omega}_{t-1}(\mathbf{A}_{\theta,\epsilon} \oplus \mathbf{A}_{\theta,\epsilon})^T + \sigma^{-2}\mathbf{I}$. Note that this is not necessarily sparse as assumed by the Sylvester graphical model, but the steady-state precision matrix satisfies $\mathbf{\Omega}_\infty = (\mathbf{A}_{\theta,\epsilon} \oplus \mathbf{A}_{\theta,\epsilon})\mathbf{\Omega}_\infty(\mathbf{A}_{\theta,\epsilon} \oplus \mathbf{A}_{\theta,\epsilon})^T + \sigma^{-2}\mathbf{I}$, which is indeed sparse because $\mathbf{A}_{\theta,\epsilon}$ is tridiagonal. This strong connection between the Sylvester graphical model and the underlying physical processes governing solar activities make the proposed approach particularly suitable for the case study presented in the previous section.

Additionally, the learned generating factors $\mathbf{A}_{\theta,\epsilon}$ could be further used to interpret physical processes that involve both *unknown structure and unknown parameters*. Particularly, in Equation (11), the coefficients θ (diffusion constant) and ϵ (convective constant) affect the dynamics. Similarly, with the estimated Sylvester generating factors ($\mathbf{\Psi}_k$'s), we are not only able to extract the sparsity patterns of the discretized differential operators but also estimate the coefficients of the underlying magneto-hydrodynamics equation for solar flares. Therefore, the SG-PALM can be used as a data-driven method for PDE parameter estimation from physical observations.

6. Conclusion

We proposed SG-PALM, a proximal alternating linearized minimization method for solving a pseudo-likelihood based sparse tensor-variate Gaussian precision matrix estimation problem. Geometric rate of convergence of the proposed algorithm is established building upon recent advances in the theory of PALM-type algorithms. We demonstrated that SG-PALM outperforms the coordinate-wise minimization method in general, and in ultra-high dimensional settings SG-PALM can be faster by at least an order of magnitude. A link between the Sylvester generating equation underlying the graphical model and the Convection-Diffusion type of PDEs governing certain physical processes was established. This connection was illustrated on a novel astrophysics ap-

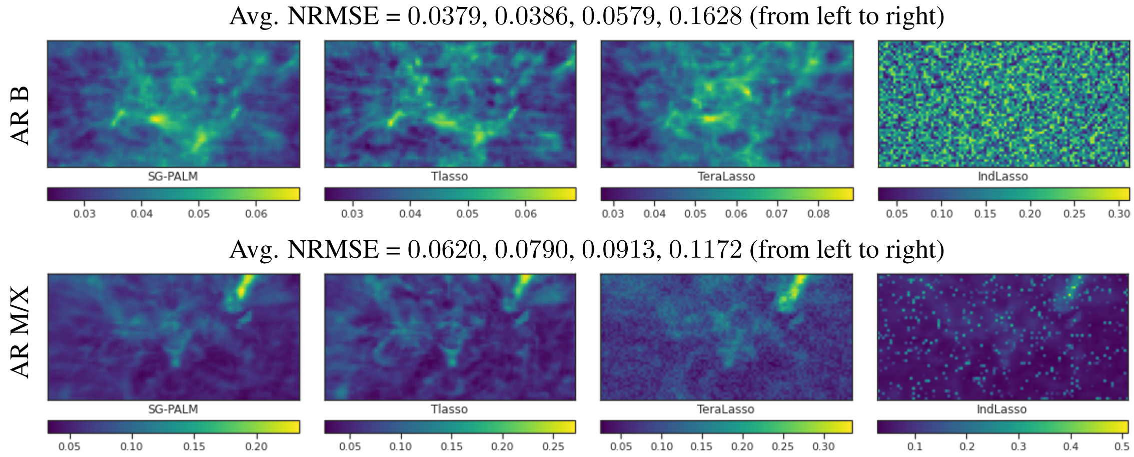


Figure 2. Comparison of the SG-PALM, Tlasso, TeraLasso, IndLasso performances measured by NRMSE in predicting the last frame of 13-frame video sequences leading to B- and MX-class solar flares. The NRMSEs are computed by averaging across testing samples and AIA channels for each pixel. 2D images of NRMSEs are shown to indicate that certain areas on the images (usually associated with the most abrupt changes of the magnetic field/solar atmosphere) are harder to predict than the rest. SG-PALM achieves the best overall NRMSEs across pixels. B flares are generally easier to predict due to both a larger number of samples in the training set and smoother transitions from frame to frame within a video (see the supplemental material for details).

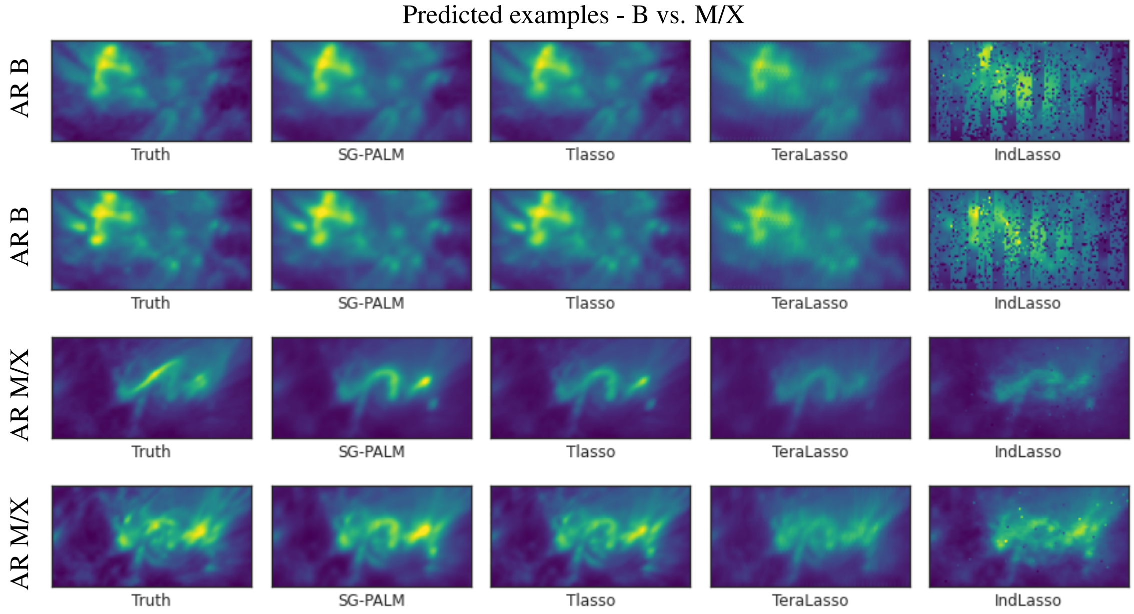


Figure 3. Examples of one-hour ahead prediction of the first two AIA channels of last frames of 13-frame videos, leading to B- (first two rows) and MX-class (last two rows) flares, produced by the SG-PALM, Tlasso, TeraLasso, IndLasso algorithms, comparing to the real image (far left column). Note that in general linear forward predictors tend to underestimate the contrast ratio of the images. The proposed SG-PALM produced the best-quality images in terms of both the spatial structures and contrast ratios.See the supplemental material for examples of predicted images from the HMI instrument.

plication, where multi-instrument imaging datasets characterizing solar flare events were used. The proposed methodology was able to robustly forward predict both the patterns and intensities of the solar atmosphere, yielding potential insights to the underlying physical processes that govern the flaring events.

Acknowledgements

The authors thank Zeyu Sun and Xiantong Wang for their help in pre-processing the solar flare datasets. The research was partially supported by US Army grant W911NF-15-1-0479 and NASA grant 80NSSC20K0600.

References

Allen, G. I. and Tibshirani, R. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764, 2010.

Attouch, H. and Bolte, J. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009.

Barzilai, J. and Borwein, J. M. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

Besag, J. Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, pp. 616–618, 1977.

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

Chen, Y., Manchester, W. B., Hero, A. O., Toth, G., DuFumier, B., Zhou, T., Wang, X., Zhu, H., Sun, Z., and Gombosi, T. I. Identifying solar flare precursors using time series of sdo/hmi images and sharp parameters. *Space Weather*, 17(10):1404–1426, 2019.

Dai, Y.-H. and Liao, L.-Z. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal of Numerical Analysis*, 22(1):1–10, 2002.

Dalal, O. and Rajaratnam, B. Sparse gaussian graphical model estimation via alternating minimization. *Biometrika*, 104(2):379–395, 2017.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Fletcher, R. On the Barzilai-Borwein method. In *Optimization and control with applications*, pp. 235–256. Springer, 2005.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C., Wright, P. J., Bobra, M. G., Liu, Y., Mason, J., et al. A machine-learning data set prepared from the nasa solar dynamics observatory mission. *The Astrophysical Journal Supplement Series*, 242(1):7, 2019.

Grasedyck, L. Existence and computation of low kronecker-rank approximations for large linear systems of tensor product structure. *Computing*, 72(3-4):247–265, 2004.

Greenewald, K., Zhou, S., and Hero III, A. Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(5):901–931, 2019.

Guillot, D., Rajaratnam, B., Rolfs, B. T., Maleki, A., and Wong, I. Iterative thresholding algorithm for sparse inverse covariance estimation. *arXiv preprint arXiv:1211.2532*, 2012.

Jiao, Z., Sun, H., Wang, X., Manchester, W., Hero, A., Chen, Y., et al. Solar flare intensity prediction with machine learning models. *arXiv preprint arXiv:1912.06120*, 2019.

Kalaitzis, A., Lafferty, J., Lawrence, N. D., and Zhou, S. The bigraphical lasso. In *International Conference on Machine Learning*, pp. 1229–1237, 2013.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Khare, K., Oh, S.-Y., and Rajaratnam, B. A convex pseudo-likelihood framework for high dimensional partial correlation estimation with convergence guarantees. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pp. 803–825, 2015.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Kressner, D. and Tobler, C. Krylov subspace methods for linear systems with tensor product structure. *SIAM journal on matrix analysis and applications*, 31(4):1688–1714, 2010.

Leng, C. and Tang, C. Y. Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499): 1187–1200, 2012.

Li, G. and Pong, T. K. Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.

Lindgren, F., Rue, H., and Lindström, J. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Lourenço, B. F. and Takeda, A. Generalized subdifferentials of spectral functions over euclidean jordan algebras. *arXiv preprint arXiv:1902.05270*, 2019.

Lyu, X., Sun, W. W., Wang, Z., Liu, H., Yang, J., and Cheng, G. Tensor graphical model: Non-convex optimization and statistical inference. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Meinshausen, N., Bühlmann, P., et al. High-dimensional graphs and variable selection with the lasso. *Annals of statistics*, 34(3):1436–1462, 2006.

Oh, S., Dalal, O., Khare, K., and Rajaratnam, B. Optimization methods for sparse pseudo-likelihood graphical model selection. In *Advances in Neural Information Processing Systems*, pp. 667–675, 2014.

Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

Park, S., Shedden, K., and Zhou, S. Non-separable covariance models for spatio-temporal data, with applications to neural encoding analysis. *arXiv preprint arXiv:1705.05265*, 2017.

Raydan, M. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13(3):321–326, 1993.

Raydan, M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997.

Roberts, B. Slow magnetohydrodynamic waves in the solar atmosphere. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1839):447–460, 2006.

Rudelson, M. and Zhou, S. Errors-in-variables models with dependent measurements. *Electronic Journal of Statistics*, 11(1):1699–1797, 2017.

Shefi, R. and Teboulle, M. On the rate of convergence of the proximal alternating linearized minimization algorithm for convex problems. *EURO Journal on Computational Optimization*, 4(1):27–46, 2016.

Sun, H., Manchester, W., Jiao, Z., Wang, X., and Chen, Y. Interpreting lstm prediction on solar flare eruption with time-series clustering. *arXiv preprint arXiv:1912.12360*, 2019.

Sun, W. W., Wang, Z., Lyu, X., Liu, H., and Cheng, G. *Tlasso: Non-Convex Optimization and Statistical Inference for Sparse Tensor Graphical Models*, 2016. URL https://CRAN.R-project.org/package=Tlasso. R package version 1.0.1.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tsiligkaridis, T., Hero III, A. O., and Zhou, S. On convergence of kronecker graphical lasso algorithms. *IEEE transactions on signal processing*, 61(7):1743–1755, 2013.

Varin, C., Reid, N., and Firth, D. An overview of composite likelihood methods. *Statistica Sinica*, pp. 5–42, 2011.

Wang, Y. and Ma, S. Projected Barzilai-Borwein method for large-scale nonnegative image restoration. *Inverse Problems in Science and Engineering*, 15(6):559–583, 2007.

Wang, Y., Jang, B., and Hero, A. The sylvester graphical lasso (syglasso). In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1943–1953, 2020. URL http://proceedings.mlr.press/v108/wang20d.html.

Wen, Z., Yin, W., Goldfarb, D., and Zhang, Y. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.

Wright, S. J., Nowak, R. D., and Figueiredo, M. A. Sparse reconstruction by separable approximation. *IEEE Transactions on signal processing*, 57(7):2479–2493, 2009.

Xu, Y. and Yin, W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3):1758–1789, 2013.

Yin, J. and Li, H. Model selection and estimation in the matrix normal graphical model. *Journal of multivariate analysis*, 107:119–140, 2012.

Zhang, C.-H. et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2): 894–942, 2010.

Zhang, H. New analysis of linear convergence of gradient-type methods via unifying error bound conditions. *Mathematical Programming*, 180(1):371–416, 2020.

Zhou, S. GEMINI: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014.

Zou, H. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.