Learning Multiple Granger Graphical Models via Group Fused Lasso

Jitkomut Songsiri
Department of Electrical Engineering
Faculty of Engineering, Chulalongkorn University
254 Phayathai Road, Pathumwan, Bangkok, Thailand 10330
Email: jitkomut.s@chula.ac.th

***Abstract*—Granger graphical models explain Granger causality between variables in time series through an estimation of zero pattern of coefficients in multivariate autoregressive (AR) models. In this paper, we consider a problem of estimating multiple Granger graphical models simultaneously that share similar topology structures from a set of time series data belonging to distinct classes. This is achieved by estimating a group of AR models and employing group fused lasso penalties to promote sparsity in AR coefficients of each model and sparsity in the difference between AR coefficients from two adjacent models. The resulting problem is in a class of group fused lasso formulation which fits nicely in a convex framework and then can be solved by a fast alternating directions method of multipliers (ADMM) algorithm. Advantages of the proposed method and the performance of the algorithm are illustrated through randomly generated data in a high-dimensional setting.**

I. Introduction

There has been a growing interest on exploring causal structures in multivariate time series considered in a number of fields, including biology, neuroscience, and finance; see examples in [1], [2], [3], [4]. Such relationship can be represented as a graphical model where the directional edges specify the *Granger causality* structure of variables [5], which states that time series y_i is Granger-caused by time series y_j if knowing the past values of y_j helps improve the prediction of y_i. The characterization of Granger causality for autoregressive processes which is widely used to model multivariate time series turns out to be very simple. Consider a q-dimensional autoregressive (AR) process of order p given by

$$y(t) = A_1 y(t-1) + A_2 y(t-2) + \cdots + A_p y(t-p) + u(t) \quad (1)$$

where $y(\cdot) \in \mathbf{R}^q$, $A_k \in \mathbf{R}^{q \times q}$, $k = 1, 2, \ldots, p$ and $u(\cdot)$ is input noise. The absence of a directed edge from node j to node i illustrates that y_i is not *Granger-caused* by y_j and this can be characterized in terms of AR coefficients as [5]

$$(A_k)_{ij} = 0, \quad k = 1, 2, \ldots, p \quad (2)$$

(where $(A_k)_{ij}$ denotes the (i, j) entry of A_k.) To explore Granger causality underlying in the data, we fit an AR model to a time series of interest and determine the zero entries in the estimated AR coefficients. To this end, an estimation problem based on the least-squares method with ℓ_1-regularization for learning Granger graphical models of time series was proposed in [6], [7]; see related work and applications in the references therein. The problem in [6] was regarded as a *group lasso* formulation [8] which is widely-known in the area of sparse estimation.

In this paper, we extend the work in [6] to the task of learning K Granger graphical models of from K sets of time series under the assumption that the K graphical models are similar (having some common edges) with some certain differences. As an application of this framework, we can consider a problem of learning brain networks from fMRI (function magnetic resonance imaging) time series of two types of patients: healthy patient and disordered patient (such as patients having cancer, Alzheimer, or Schizophrenia.) Brain connectivity networks of the two groups are expected to share some common connections due to a normal operation of brain functioning but they should not be identical because of some abnormality from the disease. Therefore, simply estimating a brain network of each group of patient separately fails to make use the fact that the two graphical models should be substantially similar. It is thus reasonable to *jointly* estimate the two graphical models such that they are promoted to have some common edges but at the same time we allow them to have some certain differences. To this end, we propose an ℓ_1 penalty on the consecutive difference between the AR coefficients of K models. This idea was initiated from a formulation called *fused lasso* [9], [10] where the goal is to promote sparsity in both the coefficients of the solution and their successive differences.

Related ideas include problems of learning multiple Gaussian graphical models [11], [12], [13], [14] which encodes the *conditional dependence* relationships among multivariate random variables. Learning a single Gaussian graphical model is based on estimating a sparse inverse of covariance matrix which can be formulated as a maximum likelihood estimation with ℓ_1 penalty on the inverse covariance. In those work, jointly estimating multiple models was obtained by adding an ℓ_1 penalty on the differences between the inverse covariance matrices of multiple models and has been typically known as *fused graphical lasso* formulation. While this approach is useful for many applications such as learning gene expression network, or micro-array data, etc., its main limitation is that it cannot be applicable to time series data.

We state the problem more clearly in Section II and show that it can be regarded as a *group fused lasso* problem in Section III, where we illustrate the effectiveness of our approach over the group lasso formulation. Our approach for learning multiple Granger graphical models can be formulated as a convex optimization, which can be readily solved by a generic solver such as SDPT3 or SeDumi called from a MATLAB package CVX [15]. To solve the problem in large scale, we apply an efficient alternating directions method of multipliers (ADMM) algorithm [16] that is greatly faster than generic convex optimization solvers and has been used for solving problems in related fields recently. Details of the algorithm is described in Section IV and numerical examples on synthetic data are presented in Section V.

II. Problem Statement

In [6], the author has proposed an estimation formulation for learning a Granger graphical model of time series. Given the measurements $y(1), y(2), \ldots, y(N)$, the problem is essentially to fit an AR process to measurement data in a least-squares sense, while to promote sparsity in the AR coefficients. The cost objective in the resulting optimization problem consists of two terms: the quadratic penalty and the ℓ_1-regularization term, which can be described as

$$\min_{A} \quad \tfrac{1}{2}\|Y - AH\|_2^2 + \lambda \sum_{i \neq j} \|[(A_1)_{ij} \; (A_2)_{ij} \; \cdots \; (A_p)_{ij}]\|_2 \tag{3}$$

with variables $A = (A_1, \ldots, A_p)$, and $A_k \in \mathbf{R}^{q \times q}$, $k = 1, \ldots, p$. The matrices Y and H contain the past measurements of $y(t)$. The first term in the objective represents a quadratic goodness of fit indicating the mismatch error between the model and the data. The second term which is an ℓ_1 penalty promotes a group sparsity in all time-lag AR coefficients (applied only the off-diagonal entries) and the sparseness can be controlled via the positive-valued regularization parameter λ. If multiple sets of time series data are given without any assumption on the similarity among those data sets, estimation of multiple AR models with Granger causality can then be obtained by solving (3) independently.

We propose a problem of *jointly* estimating K autoregressive models where a common Granger-causality structure in K models is referred as

$$\begin{aligned} \underset{A^{(1)},\ldots,A^{(K)}}{\text{minimize}} \; & \sum_{k=1}^{K} \frac{1}{2}\|Y^{(k)} - A^{(k)}H^{(k)}\|_2^2 + \lambda_1 \sum_{i \neq j} \sum_{k=1}^{K} \left\| B_{ij}^{(k)} \right\|_2 \\ & + \lambda_2 \sum_{i \neq j} \sum_{k=1}^{K-1} \left\| B_{ij}^{(k+1)} - B_{ij}^{(k)} \right\|_2 \end{aligned} \tag{4}$$

where $B_{ij}^{(k)} = \left[(A_1^{(k)})_{ij} \quad (A_2^{(k)})_{ij} \quad \cdots \quad (A_p^{(k)})_{ij} \right]^T \in \mathbf{R}^p$. The notation $A^{(k)} = (A_1^{(k)}, \ldots, A_p^{(k)})$ refers to the AR coefficients of the k^{th} model; $Y^{(k)}$ and $H^{(k)}$ contain measurement data of the k^{th} time series. The two regularization parameters, λ_1 and λ_2, are positive real numbers. The third term in the objective is a sum of 2-norm which is an ℓ_1 penalty applied on the differences between corresponding off-diagonal elements of AR coefficients from two consecutive models. When the tuning parameter λ_2 is large enough, many elements of $(A^{(1)})_{ij}, (A^{(2)})_{ij}, \ldots, (A^{(K)})_{ij}$ will be identical, resulting in a set of common edges in the K graphical models. When λ_1 is large enough, many elements of $(A^{(k)})_{ij}$ (for some k) will be zero which indicates that sparse AR models are obtained. Therefore, this formulation encourages not only the sparseness of the models, but also similar network structure and similar AR coefficients across the K models.

As a toy example to illustrate an advantage of this formulation, we solve (4) by using $q = 3, p = 2$ and $K = 3$, *i.e.*, we estimate three 3-dimensional AR models of order 2. In the first case, we set λ_2 to be relatively large (compared to λ_1). This means we aim to have common values of AR coefficients in the three models. Figure 1 shows the q^2 values of $(A^{(k)})_{ij}$ and it illustrates that for each time-lag coefficient (each column in the figure), all the three models share the same off-diagonal coefficient values (blue circles) if we set λ_2 to be large enough. However, the diagonal entries of AR coefficients (red squares) from the three models can be different. In the second case, if λ_1 is relatively large (compared to λ_2), this means sparse models are preferred but all the K network structures do not need to lie in common. Therefore, Figure 2 shows that for each time-lag coefficient (each column in the figure), all the three models could have different coefficients. However, each model (each row in the figure) contains many zero coefficients and the zero entries of each time-lag coefficient must occur at the same location because we promote a group sparsity across $(A_1^{(k)})_{ij}, \ldots, (A_p^{(k)})_{ij}$.

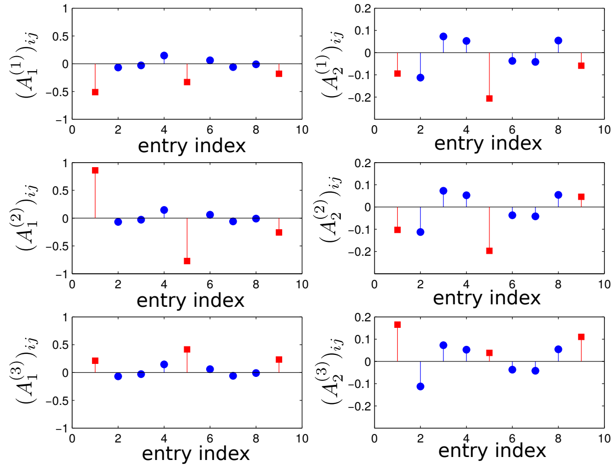


Fig. 1. Profiles of AR coefficients estimated by solving (4) when λ_2 is sufficiently large. Red squares denote the diagonal entries and blue circles denote the off-diagonal entries of AR coefficients.

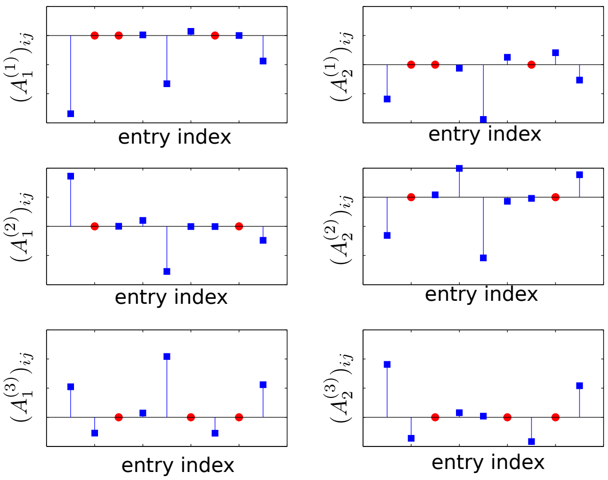


Fig. 2. Profiles of AR coefficients estimated by solving (4) when λ_1 is sufficiently large. Red circles denote the zero entries and blue circles denote the nonzero entries of AR coefficients.

III. Group Fused Lasso VS Group Lasso

In this section, we rewrite the problem (4) into a vector form, and show that the problem can be regarded as a *group fused Lasso* formulation [17]. Define

$$x = \left(B_{11}^{(1)}, \ldots, B_{11}^{(K)}, B_{12}^{(1)}, \ldots, B_{12}^{(K)}, \ldots, B_{qq}^{(1)}, \ldots, B_{qq}^{(K)}\right). \tag{5}$$

In another word, x is obtained by vectorizing the AR coefficients of the K models. x has n entries where $n = q^2 pK$ and can be partitioned into q^2 main blocks. Each main block has K subblocks; each of which has size p. Suppose $z = (z_1, z_2, \ldots, z_L)$ and $z_k \in \mathbf{R}^p$ for $k = 1, 2, \ldots, L$. Define *the sum of* 2*-norm:* $\|z\|_{2,1} = \sum_{k=1}^{L} \|z_k\|_2$ and a projection matrix $P \in \mathbf{R}^{(q^2-q)\times q^2}$ such that for any matrix $X \in \mathbf{R}^{q\times q}$, the off-diagonal entries of X is obtained by $P \cdot \mathrm{vec}(X)$. For example, for $X = [x_{ij}] \in \mathbf{R}^{2\times 2}$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P \cdot \mathrm{vec}(X) = \begin{bmatrix} x_{21} \\ x_{12} \end{bmatrix}.$$

As a result, the second term of the objective in (4) can be written as $\lambda_2 \|\mathcal{P}x\|_{2,1}$ where $\mathcal{P} = P \otimes I_{pK}$. Moreover, the last term of the objective in (4) can be expressed as $\lambda_2 \|\mathcal{D}x\|_{2,1}$ where $\mathcal{D} = P \otimes D$ and D is the forward difference matrix:

$$D = \begin{bmatrix} -I_p & I_p & & & \\ & -I_p & I_p & & \\ & & \ddots & \ddots & \\ & & & -I_p & I_p \end{bmatrix}.$$

By the definition of x in (5), it is straightforward (but requires some derivation we opt to omit the details) to rewrite the problem (4) in a vector form as

$$\underset{x}{\text{minimize}} \quad (1/2)\|Gx - b\|_2^2 + \lambda_1 \|\mathcal{P}x\|_{2,1} + \lambda_2 \|\mathcal{D}x\|_{2,1} \tag{6}$$

with variable $x \in \mathbf{R}^n$. The matrices $G \in \mathbf{R}^{m\times n}, b \in \mathbf{R}^m, \mathcal{P} \in \mathbf{R}^{s\times n}$ and $\mathcal{D} \in \mathbf{R}^{r\times n}$ are parameters of the problem. It can be shown that G is a sparse matrix. Moreover, G and b contain the measurement values of K time series $y^{(1)}(t), \ldots, y^{(K)}(t)$ for $t = 1, 2, \ldots, N$. The dimensions of $G, b, \mathcal{P}$ and $\mathcal{D}$ are related to those of (4) by $n = q^2 pK, m = nNK, s = (n^2 - n)pK$ and $r = (n^2 - n)p(K-1)$.

When $\lambda_2 = 0$ and $\lambda_1 > 0$, the problem (6) reduces to the *group lasso* formulation [8], [18] and it essentially becomes the problem (3) considered in [6]. The resulting problem corresponds to estimating multiple AR models independently and as we increase λ_1, we obtain sparser models. If $\lambda_1 = 0$ and $\lambda_2 > 0$, then the problem (6) is in a class of *total variation regularized problem* [19] which finds many applications including image reconstruction [20], [21] or estimation of piecewise constant parameters in time-varying models [22]. The key feature that these work have in common is the additional penalty on the difference of two successive variables (the term $\|\mathcal{D}x\|_{2,1}$) in the cost objective of their formulations. When both λ_1 and λ_2 are positive and $p = 1$ in (4) (estimating a first-order AR process), then it is equivalent to replacing the sum of 2-norm in (6) by the 1-norm where we neglect the group structure of the variable. In this case, the problem is simplified to *fused lasso* proposed by [9]. For $p > 1$, the problem (6) is termed as *group fused lasso* problem and has been discussed in [17] with the application of image denoising. We note that the formulation in [17] is identical to (6) when $\mathcal{P} = I$ and $\mathcal{D} = D$, or equivalently when $q = 1$, *i.e.*, we estimate a scalar AR process.

IV. Alternating Direction Method of Multipliers (ADMM)

In this section we describe an efficient algorithm for solving the problem (6) in large scale through convex optimization techniques. Due to non-differentiability of $\|\cdot\|_{2,1}$ terms in the cost objective, we explore gradient-based methods that are applicable to non-smooth problems. The alternating direction method of multiplier (ADMM), which is a Douglas-Rachford splitting technique applied to the dual problem [16], [23] is one of the techniques that has been recently applied to many large-scale machine learning problems due to i) its inexpensive computational cost per iteration and ii) its favorably fast convergence in practice. To apply this method, we reformulate the problem by splitting the cost objective into several terms and introducing some auxiliary variables and equality constraints. In [23], it can be shown that ADMM can also be represented as a proximal algorithm since its update rule relies on the use of proximal operators. A proximal method is a technique to minimize convex problems (but possibly non-smooth) by splitting the cost objective into several terms, one of which is differentiable and the iteration rule involves performing the proximal operators of the splitted functions. A proximal method can be beneficial only when the cost of computing proximal operators is cheap depending on how we split the cost function. Therefore, different splitting results in different implementations of the proximal method for the same problem, and not all of them is efficient.

In what follows, we shall explain in details how to split the cost objective in (6) that leads to efficient ADMM framework and describe the algorithm with implementation. If we define

$$f(x) = (1/2)\|Gx - b\|_2^2,$$
$$g(x) = \lambda_1 \|x\|_{2,1}, \quad h(x) = \lambda_2 \|x\|_{2,1}$$

then we can rearrange (6) into ADMM format as

$$\begin{array}{ll} \text{minimize} & f(x_1) + g(x_2) + h(x_3) \\ \text{subject to} & \begin{bmatrix} \mathcal{P} \\ \mathcal{D} \end{bmatrix} x_1 = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \end{array} \tag{7}$$

with variables $x_1 \in \mathbf{R}^n, x_2 \in \mathbf{R}^s$ and $x_3 \in \mathbf{R}^r$. The augmented Lagrangian which is the Lagrangian function plus the quadratic penalty for the constraint is given by

$$\begin{aligned} L_\rho = f(x_1) + g(x_2) + h(x_3) + (\rho/2)\|\mathcal{P}x_1 - x_2 + z_1/\rho\|_2^2 \\ + (\rho/2)\|\mathcal{D}x_1 - x_3 + z_2/\rho\|_2^2, \end{aligned} \tag{8}$$

where $\rho > 0$ is called the *penalty parameter* and its value affects the convergence speed. The ADMM algorithm is to perform minimization on the augmented Lagrangian over x_1 and (x_2, x_3) alternatingly. Since the terms involving x_2 and x_3 in L_ρ are separable, we can minimize L_ρ over x_2 and x_3 separately. Denote x_i^+ the updated variable in the next step.

By following the details in [16], the update rule is given by

$$\begin{aligned}
x_1^+ &= \operatorname*{argmin}_{x_1} \ \frac{1}{2}\|Gx_1 - b\|_2^2 + \frac{\rho}{2}\|\mathcal{P}x_1 - x_2 + z_1/\rho\|_2^2 \\
&\quad + \frac{\rho}{2}\|\mathcal{D}x_1 - x_2 + z_2/t\|_2^2, && (9)\\
x_2^+ &= \operatorname*{argmin}_{x_2} \ \lambda_1\|x_2\|_{2,1} + \frac{\rho}{2}\|\mathcal{P}x_1^+ - x_2 + z_1/\rho\|_2^2, && (10)\\
x_3^+ &= \operatorname*{argmin}_{x_3} \ \lambda_2\|x_3\|_{2,1} + \frac{\rho}{2}\|\mathcal{D}x_1^+ - x_3 + z_2/\rho\|_2^2, && (11)\\
z_1^+ &= z_1 + \rho(\mathcal{P}x_1^+ - x_2^+), && (12)\\
z_2^+ &= z_2 + \rho(\mathcal{D}x_1^+ - x_3^+). && (13)
\end{aligned}$$

The x_1-update can be obtained in closed form by setting the gradient of the objective to zero. This gives the linear equation:

$$\begin{aligned}
&\left[G^TG + \rho(\mathcal{P}^T\mathcal{P} + \mathcal{D}^T\mathcal{D})\right] x_1^+ = \\
&\qquad G^Tb + \mathcal{P}^T(\rho x_2 - z_1) + \mathcal{D}^T(\rho x_3 - z_2). \quad (14)
\end{aligned}$$

Since $G^TG + \rho(\mathcal{P}^T\mathcal{P} + \mathcal{D}^T\mathcal{D})$ is positive definite, we can perform a Cholesky factorization and cache the Cholesky factor, denoted by L. We also note that $\mathcal{P}^T\mathcal{P}$ is diagonal; $\mathcal{D}^T\mathcal{D}$ is a banded matrix and G^TG can be shown to be sparse. Hence, performing Cholesky factorization can be very efficient (and further improved if we exploit the structure of $G^TG + \rho(\mathcal{P}^T\mathcal{P} + \mathcal{D}^T\mathcal{D})$) and L is also sparse. The update on x_1 is then obtained by solving the linear equations: $L^Tv = c$, $Lx_1 = v$, where $c = G^Tb + \mathcal{P}^T(\rho x_2 - z_1) + \mathcal{D}^T(\rho x_3 - z_2)$.

The updates on x_2 and x_3 share the same structure which can be written in a general form as

$$\operatorname*{minimize}_{x} \ \gamma\|x\|_{2,1} + (1/2)\|x - u\|_2^2,$$

for some $\gamma > 0$ and $u \in \mathbf{R}^n$. Suppose u and x can be partitioned into L blocks. The above problem is known as finding the *proximal operator* of $f_1(x) = \|x\|_{2,1} = \sum_{k=1}^{L}\|x_k\|_2$ which can be obtained in closed-form as

$$\begin{aligned}
\mathbf{prox}_{\gamma f_1}(u) &= \operatorname*{argmin}_{x} \ \gamma\|x\|_{2,1} + (1/2)\|x - u\|_2^2, \\
(\mathbf{prox}_{\gamma f_1}(u))_k &= \max\left\{1 - \frac{\gamma}{\|u_k\|_2}, 0\right\} u_k,
\end{aligned}$$

for $k = 1, 2, \ldots, L$. The proximal operator of the sum of 2-norm function is typically known as the *block soft thresholding* operator [23]. As for completeness of the ADMM algorithm for solving (6), we describe the update rule explicitly as follows.

ADMM for Group Fused Lasso problem. Initialize x_1, x_2, x_3, z_1, z_2. Set an ADMM parameter $\rho > 0$. Denote (x, x^+) the variables in the current and next iteration, respectively. Repeat the following steps

$$\begin{aligned}
c &= G^Tb + \mathcal{P}^T(\rho x_2 - z_1) + \mathcal{D}^T(\rho x_3 - z_2), \\
x_1^+ &= \left[G^TG + \rho(\mathcal{P}^T\mathcal{P} + \mathcal{D}^T\mathcal{D})\right]^{-1} c, \\
x_2^+ &= \mathbf{prox}_{(\lambda_1/\rho)f_1}(\mathcal{P}x_1^+ + z_1/\rho), \\
x_3^+ &= \mathbf{prox}_{(\lambda_2/\rho)f_1}(\mathcal{D}x_1^+ + z_2/\rho), \\
z_1^+ &= z_1 + \rho(\mathcal{P}x_1^+ - x_2^+), \\
z_2^+ &= z_2 + \rho(\mathcal{D}x_1^+ - x_3^+),
\end{aligned}$$

until the primal residual, r and the dual residual, s, are less than some tolerance values:

$$\begin{aligned}
\|r\|_2 &= \left\| \begin{bmatrix} \mathcal{P}x_1 - x_2 \\ \mathcal{D}x_1 - x_3 \end{bmatrix} \right\|_2 \leq \epsilon^{\text{pri}}, \\
\|s\|_2 &= \rho \left\| \begin{bmatrix} \mathcal{P}^T(x_2^+ - x_2) \\ \mathcal{D}^T(x_3^+ - x_3) \end{bmatrix} \right\|_2 \leq \epsilon^{\text{dual}}.
\end{aligned}$$

The tolerance values ϵ^{pri} and ϵ^{dual} can be computed according to [16]. We see that the iteration will return x_2 as sparse vector due to the block soft-thresholding operator, while x_1 is close to x_2 but not sparse.

Lastly we have some following comments on other algorithms that are related to our problem. i) Another accelerated proximal gradient algorithm such as FISTA (which has been applied extensively in many image processing problems [20], [24]) cannot apply to (6) directly since the resulting update rule would involve finding the proximal operator of the sum of two functions $(\lambda_1\|\mathcal{P}x\|_{2,1} + \lambda_2\|\mathcal{D}x\|_{2,1})$ which cannot be readily built by the proximal operator of each function. ii) While the general setting of our problem and the formulation considered in [17] are not identical, it could be possible that the algorithm used in [17] could be applicable to ours. In [17], they applied FISTA algorithm and solve the proximal operator of the sum of two functions by the proximal Dykstra algorithm [25]. To this end, it requires computing the proximal operator of $\|\mathcal{P}x\|_{2,1}$ and $\|\mathcal{D}x\|_{2,1}$ separately. The first proximal operator is simple, but the latter has to be solved by the projected gradient algorithm on the dual of proximal problem. As a result, it requires solving a subproblem in each iteration of the FISTA update. We believe that the ADMM method should be more efficient and can be implemented more directly in this problem. iii) Our problem (6) can be fit into the framework considered in [21] where their primary focus is on the primal-dual splitting technique that compares favorably with the Douglas-Rachford algorithm applied to the primal or to the dual problem (the latter is ADMM).

V. Numerical Examples

In this section, first we illustrate an advantage of using the group fused lasso formulation in estimating multiple AR models with the assumption that these AR models are sparse and have a *similar* topology of Granger causality network. To this end, we consider three 10-dimensional stable AR models of order 2, *i.e.*, $q = 10, p = 2, K = 3$. Then we set $A^{(2)} = A^{(1)}$ and add a few nonzero off-diagonal entries on $A^{(2)}$ and do the same for $A^{(3)}$. We generate 100 data sets; each of which contains 50 points of time series corrupted by noise of unit variance. We note that the number of samples ($N = 50$) is fairly low compared to the number of estimation variables. Grid values of (λ_1, λ_2) in log-scale are chosen. For each fixed λ_2 there exists a closed-form expression of the maximum value of λ_1 such that the estimated model is the sparsest. This value of $\lambda_{1,\max}$ is derived from the optimality condition, but due to space limitation, we do not include the formula here. For each pair of (λ_1, λ_2) we solve (6) and obtain the estimated Granger structure through the estimated zero pattern of AR coefficients and compare it with the true structure. Figure 3 displays the plot of True Positive Rate (TPR) versus the False Positive Rate

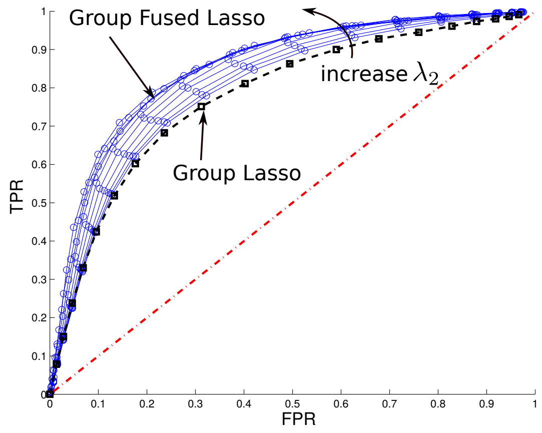


Fig. 3. Performance of Group Lasso and Group Fused Lasso formulations in estimating Granger graphical model topology. The True Positive Rate (TPR) is plotted versus the False Positive Rate (FPR) where TPR is the number of correctly identified entries as nonzero in AR coefficients (True Positive-TP) normalized by the total number of true nonzero entries and FPR is the number of incorrectly identified entries as nonzero in AR coefficients (False Positive-FP) normalized by the total number of true zero entries. Black squares represent Group Lasso model and the blue circles are from Group Fused Lasso models with different values of λ_2.

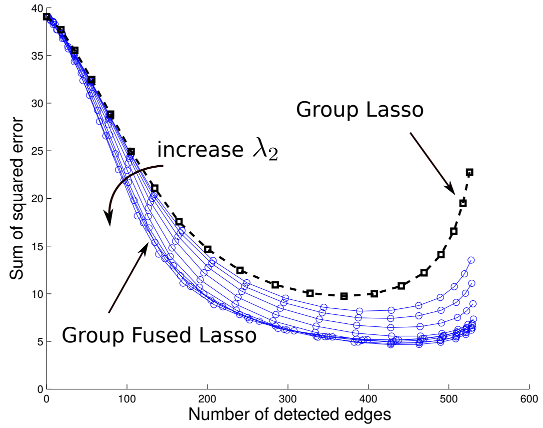


Fig. 4. The norm squared error $\|Gx - b\|_2^2$ is plotted versus the number of total edges detected in the graphical model (the number of nonzero x solved from (6).) Black squares represent Group Lasso model and the blue circles are from Group Fused Lasso model corresponding to different values of λ_2.

(FPR) that are averaged over the 100 data sets. At a fixed value of False Positives Rate, we see the curves from Group Fused Lasso model (blue circles) lie above the Group Lasso model (black squares) indicating that our approach yields a more accurate Granger structure as we increase λ_2 since we put more penalty on forcing the K models to be similar.

Figure 4 explains a trade-off between the model errors and the model sparseness but the model errors tend to be increasing if the estimated Granger network is too dense (overfitting problem.) We see that the Group Fused Lasso yields a lower model error than that of the Group Lasso model as λ_2 increases.

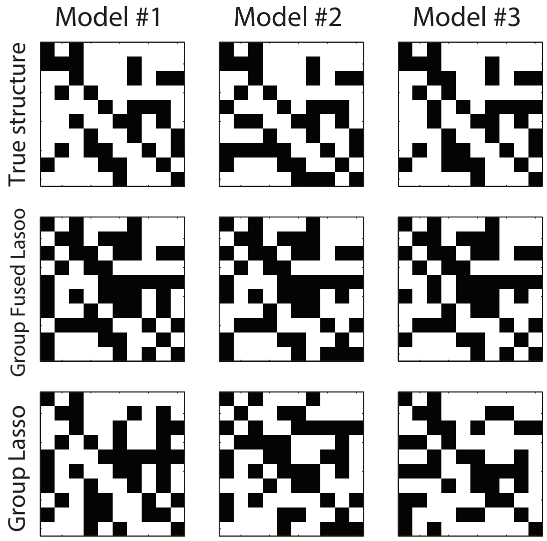
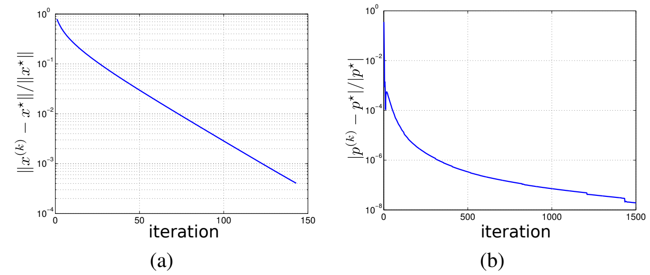


Fig. 5. Zero pattern of the estimated AR coefficients from Group Fused Lasso (middle row) and Group Lasso (bottom row) formulations compared to the true sparsity pattern.

The grey scale binary matrices in Figure 5 represent one realization of the estimated zero patterns of AR coefficients from the three models where black color represents nonzero entries of $B_{ij}^{(k)}$ for $k = 1, 2, 3$. We select the zero patterns estimated from Group Fused Lasso and Group Lasso that yield about the same FPR which is approximately around $0.21 - 0.23$. We see that the Granger causality networks of Group Fused Lasso models share a high similarity among the three models (by selecting λ_2 to be large enough). However, if we estimate the three AR models separately, then the Granger structures shown in the bottom row of Figure 5 are quite different.

In the last experiment, in order to solve large-scale problems in many applications, we illustrate the efficiency of the ADMM algorithm when the problem dimension becomes moderate to large. We solve the problem (6) with $n = 4800$. Figure 6 (a) shows the relative error $\|x^{(k)} - x^\star\|/\|x^\star\|$ where $x^{(k)}$ denotes the solution from ADMM algorithm at the k^{th} iteration and $x^\star$ is computed using CVX [15], which is a MATLAB package for solving generic convex optimization problems and returns a solution with high accuracy. The ADMM parameter (ρ) was tuned by trial and error to give a fast convergence. As can be seen, the ADMM algorithm can return a solution with relative error of 10^{-3} just by a few hundred iterations and within $2 - 3$ seconds. In the next experiment, we increase the problem dimension to $n = 30,000$. In this setting, a generic solver called by CVX faces a memory storage problem. We run ADMM for $10,000$ iterations and assume that its objective after $10,000$ iteration, denoted by $p^\star$ is a nearly optimal value. Figure 6 (b) shows $(p^{(k)} - p^\star)/p^\star$ versus the iteration number, k, where $p^{(k)}$ is the objective of (7) at iteration k. It clearly shows that we can achieve the relative error of 10^{-6} just by 300 iterations and this takes only around $300 - 400$ seconds, given that we have up to $30,000$ variables. All the experiments are programmed in MATLAB, and executed in PC with Intel Core i3 2.9 Hz and RAM 2GB. We also note that the convergence speed depends on the choice of ρ and our observation found that setting $\rho \approx 10 \max\{\lambda_1, \lambda_2\}$ gives a good convergence result but we do not have a theoretical explanation to support this.



(a) (b)

Fig. 6. (a) The relative error of the solution to (7) versus the iteration numbers. (b) The relative error between the primal objective of (7) and a nearly optimal value versus the iteration numbers.

VI. Conclusions

We have presented an optimization formulation for estimating jointly multiple autoregressive models with sparse coefficients and similarity Granger causality across the models. Applications of this work includes learning multiple Granger graphical models of time series such as fMRI collected from different patient's conditions where the goal is to learn brain connectivities with some common edges but allow them to have some structured differences due to variation in patient's condition. Our approach is based on the use of the sum of 2-norm of the difference between the AR coefficients of successive models. The problem can be cast as a group fused lasso formulation which also finds many other applications such as total variation regularized problems in image reconstruction. We have solved the problem via the ADMM method which involves solving linear equations, matrix addition/multiplication, and performing block soft thresholding in each iteration. These operations can be implemented cheaply and even more efficiently if sparse structures of the problem parameters are further exploited. Numerical examples on synthetic data sets showed that using the group fused lasso formulation yields a better estimation result given that the true models have some common structures.

Acknowledgment

The author received the New Researcher Scholarship from the Ministry of Science, Thailand to complete all the work done in this paper.

References

[1] A. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical granger modeling for gene expression regulatory networks discovery," *Bioinformatics*, vol. 25, pp. 110–118, 2009.

[2] W. Tang, S. Bressler, C. M. Sylvester, S. L. Gordon, and M. Corbetta, "Measuring Granger causality between cortical regions from voxelwise fmri bold signals with lasso," *PLoS Computational Biology*, vol. 8, no. 5, pp. 1–14, 2012.

[3] A. Shojaie and G. Michailidis, "Discovering graphical Granger causality using the truncating lasso penalty," *Bioinformatics*, vol. 26, no. 18, pp. 517–523, 2010.

[4] J. C. Rajapakse and P. A. Mundra, "Stability of building gene regulatory networks with sparse autoregressive models," *BMC Bioinformatics*, vol. 12, no. 13, pp. 1–10, 2011.

[5] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

[6] J. Songsiri, "Sparse autoregressive model estimation for learning Granger causality in time series," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3198–3202.

[7] A. Pongrattarakul, P. Lerdkultanon, and J. Songsiri, "Sparse system identification for discovering brain connectivity from fMRI time series," in *Proceedings of SICE Annual Conference*, 2013, pp. 949–954.

[8] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B Statistical Methodology*, vol. 68, no. 1, pp. 49–67, 2006.

[9] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[10] J. Liu, L. Yuan, and J. Ye, "An efficient algorithm for a class of fused lasso problems," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 323–332.

[11] S. Yang, Z. Pan, X. Shen, P. Wonka, and J. Ye, "Fused multiple graphical lasso," *arXiv preprint arXiv:1209.2139*, 2012.

[12] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *arXiv preprint arXiv:1111.0324*, 2011.

[13] K. Mohan, M. J. Chung, S. Han, D. Witten, S. Lee, and M. Fazel, "Structured learning of gaussian graphical models," in *Advances in Neural Information Processing Systems*, 2012, pp. 629–637.

[14] J. Guo, E. Levina, G. Michailidis, and J. Zhu, "Joint estimation of multiple graphical models," *Biometrika*, vol. 98, no. 1, pp. 1–15, 2011.

[15] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming (web page and software)*, `http://stanford.edu/~boyd/cvx`, 2007.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[17] C. M. Alaíz, A. Barbero, and J. R. Dorronsoro, "Group fused lasso," *Artificial Neural Networks and Machine Learning–ICANN 2013*, pp. 66–73, 2013.

[18] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[19] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An admm algorithm for a class of total variation regularized estimation problems," in *the 16th IFAC Symposium on System Identification*, 2012, pp. 83–88.

[20] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[21] D. O'Connor and L. Vandenberghe, "Primal-dual decomposition by operator splitting and applications to image deblurring," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1724–1754, 2014.

[22] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of ARX-models using sum-of-norms regularization," *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[23] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: http://dx.doi.org/10.1561/2400000003

[24] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/ jye02/Software/SLEP

[25] P. L. Combettes and J. C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.