Linear Dynamic Network Reconstruction from Heterogeneous Datasets

Zuogong Yue * **Johan Thunberg** * **Wei Pan** ** **Lennart Ljung** ***
Jorge Gonçalves *

* *Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, L-4362, Luxembourg (e-mail: zuogong.yue@uni.lu, johan.thunberg@uni.lu, jorge.goncalves@uni.lu)*
** *Centre for Synthetic Biology and Innovation and Department of Bioengineering, Imperial College London, United Kingdom (e-mail: w.pan11@imperial.ac.uk)*
*** *Department of Electrical Engineering, Linköping University, Linköping, SE-58183, Sweden (e-mail: ljung@isy.liu.se)*

Abstract: This paper addresses reconstruction of linear dynamic networks from heterogeneous datasets. Those datasets consist of measurements from linear dynamical systems in multiple experiments subjected to different experimental conditions, e.g., changes/perturbations in parameters, disturbance or noise. A main assumption is that the Boolean structures of the underlying networks are the same in all experiments. The ARMAX model is adopted to parameterize the general linear dynamic network representation "Dynamical Structure Function" (DSF), which provides the Granger Causality graph as a special case. The network identification is performed by integrating all available datasets and promote group sparsity to assure both network sparsity and the consistency of Boolean structures over datasets. In terms of solving the problem, a treatment by the iterative reweighted l_1 method is used, together with its implementations via proximal methods and ADMM for large-dimensional networks.

Keywords: system identification, dynamic network reconstruction, heterogeneous datasets.

1. INTRODUCTION

Network inference has been widely applied in different fields to learn interaction structures or dynamic behaviors. Such fields include systems biology, computer vision, econometrics, social networks, etc. However, there is no agreement upon the definition of "network", and it usually refers to a larger class of graphical models than the standard definition in the graph theory. In particular, with increasingly easier access to time-series data, it is expected that networks can explain dynamics or causal interactions. For instance, biologists use causal network inference to determine critical genes that are responsible for diseases in pathology, see e.g. Bar-Joseph et al. (2012).

In the study of causal networks, a popular model used in biology is Bayesian network, e.g. Murphy and Mian (1999). Although it delivers certain causality information, the Bayesian network is defined for *directed acyclic graphs*. See (Pearl, 1995, p. 127) for more information on the domains of different probabilistic models. However, the feedback loops in networks could be particularly important in biological applications. Concerning general causal networks, as the primary contribution in Granger (1969), Granger causality (GC) provides causality graphs (see e.g. Eichler (2007)) based on predictability to identify causation between time-series variables. However, as it was realized in Granger (1969), this approach may be problematic in deterministic settings, especially in dynamic systems with weak to moderate coupling. Inspired by GC, which is equivalent to the *vector autoregression* form under fairly weak conditions (see e.g. Eichler (2007)), an idea of adopting system theory has been proposed to deal with causal network reconstructions. For instance, Chiuso and Pillonetto (2012) proposed a kernel-based system identification approach together with group LASSO to infer GC networks.

There has been several papers proposing methods for network inference by identifying simple dynamical models in biological applications, e.g. Beal et al. (2005). To study the identifiability issue in network inference, a general network representation for linear time-invariant (LTI) systems needs to be introduced. Similar or nearly equivalent such general representations are introduced in Goncalves and Warnick (2008) and Weerts et al. (2015) with different perspectives. The results on network identifiability were firstly manifested in Goncalves and Warnick (2008). In the sense of inference, these two representations are not different, and the model description used in later sections refers to either of them interchangeably.

Most biological experiments have replica. The ordinary treatment is to take averages with the purpose of removing effects of noise. However, mostly, only a few replica are available, e.g. less than 5, which implies it no longer makes sense to work with statistical averages. Pan et al. (2015) proposes a way to take advantage of replica to increase robustness or accuracy of estimation. Yet another aspect to highlight here is that the system parameters of biological processes could be fairly different due to the variance in individuals in experiment repetitions, as observed in biological experiments He et al. (2012). The essence is the interconnection structure, e.g. interactions

⋆ This work was supported by Fonds National de la Recherche Luxembourg (Ref. AFR-9247977).

between genes, or communication between neurons, which are consistent over experiment repetitions. The method proposed in the paper allows the system parameters to be significantly different, as long as the network topology keeps consistent over replica. The whole reconstruction method is illustrated in Figure 1.

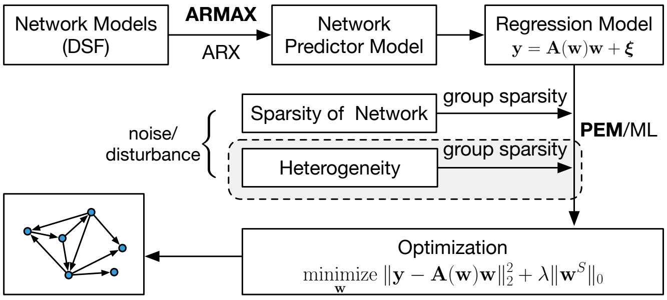

Network Models
(DSF)
ARMAX
ARX
Network
Predictor Model
Regression Model
$\mathbf{y} = \mathbf{A}(\mathbf{w})\mathbf{w} + \boldsymbol{\xi}$
noise/
disturbance
Sparsity of Network
group sparsity
Heterogeneity
group sparsity
PEM/ML
Optimization
$\underset{\mathbf{w}}{\text{minimize}}\, \|\mathbf{y} - \mathbf{A}(\mathbf{w})\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}^S\|_0$

Fig. 1. An overview of the network reconstruction method.

2. PROBLEM FORMULATION

Let $Y \triangleq \{y(t), t \in \mathbb{Z}\}$, $U \triangleq \{u(t), t \in \mathbb{Z}\}$ be multivariate time series of dimension p and m, respectively, where the elements could be deterministic ($y(t) \in \mathbb{R}^p$, $u(t) \in \mathbb{R}^m$) or be real-valued random vectors defined on probability spaces $(\Omega, \mathscr{F}, \mathbb{P})$. We usually assume that $u(t)$ is deterministic in practice, which is interpreted as controlled input signals.

2.1 Linear dynamic networks

Consider the following model for LTI systems (the *Dynamical Structure Function* (DSF) in Goncalves and Warnick (2008); or a similar model in Weerts et al. (2015))

$$y(t) = Q(q)y(t) + P(q)u(t) + H(q)e(t), \tag{1}$$

where $y(t) = [y_1(t), \dots, y_p(t)]^T$, $u(t) = [u_1(t), \dots, u_m(t)]^T$, a p-variate i.i.d. $e(t) = [e_1(t), \dots, e_p(t)]^T$ with zero mean and covariance matrix I for all t,

$$\begin{aligned} Q(q) &= [Q_{ij}(q)]_{p\times p}, & Q_{ii}(q) &= 0,\ \forall i, \\ P(q) &= [P_{ij}(q)]_{p\times m}, & H(q) &= \operatorname{diag}(H_{ii}(q))\,[1], \end{aligned}$$

$Q_{ij}(q), P_{ij}(q), H_{ii}(q)$ are single-input-single-output (SISO) transfer functions, and q is the forward-shift operator, i.e. $qy(t) = y(t+1)$, $q^{-1}y(t) = y(t-1)$.

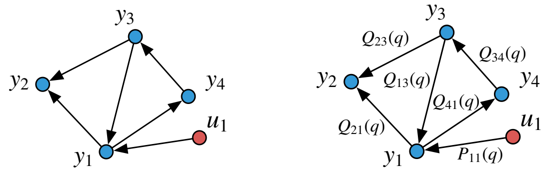
Definition 1. Let $\mathcal{G} = (V, E)$ be a digraph, where the vertex set $V = \{y_1, \dots, y_p, u_1, \dots, u_m\}$ [2], and the arc (directed edge) set E is defined by

- $(y_j, y_i) \in E \;\Leftrightarrow\; Q_{ij}(q) \neq 0$,
- $(u_k, y_i) \in E \;\Leftrightarrow\; P_{ik}(q) \neq 0$,
- $(y_i, u_k) \notin E$, $\forall i, k$.

Let f be a map defined as

$$\begin{aligned} f: \quad & E \to S_{\text{TF}} \\ & (y_j, y_i) \mapsto Q_{ij}(q) \quad \text{or} \quad (u_k, y_i) \mapsto P_{ik}(q), \end{aligned}$$

where S_{TF} is a subset of single-input-single-output (SISO) (strictly) proper real rational transfer functions. We call the tuple $\mathcal{N} := (\mathcal{G}, f)$ a (linear) *dynamic network* [3], f the (linear) *dynamic capacity function* of $\mathcal{N}$, and $\mathcal{G}$ the *underlying digraph* of $\mathcal{N}$, which is also called (linear) *Boolean dynamic network*.



(a) Boolean dynamic network

(b) dynamic network

Fig. 2. Graphical illustrations of a linear dynamic network.

2.2 Network reconstruction from multiple experiments

Now let us consider the case of multiple experiments. Let $\{Y^{[c]}, U^{[c]}\}_{c=1,\dots,C}$ denote the measurements from C experiments. We use $\mathcal{N}((Q, P, H))$ to denote the dynamic network $\mathcal{N}$ (i.e. $(\mathcal{G}, f)$) determined by (Q, P, H) (i.e. (1)); and $\mathcal{G}((Q, P, H))$ the corresponding Boolean dynamic network. The governing model (1) could be different in each experiment, denoted by $(Q, P, H)^{[c]}, c = 1, \dots, C$. In addition, $\mathcal{N}^0$ denotes a fixed dynamic network, and $\mathcal{G}^0$ a fixed Boolean dynamic network.

We say the datasets $\{Y^{[c]}, U^{[c]}\}$ are *homogeneous*, if $\mathcal{N}((Q, P, H)^{[c]}) \equiv \mathcal{N}^0, \forall c$, i.e. the measurements are from the same dynamic network. And the datasets $\{Y^{[c]}, U^{[c]}\}$ are called *heterogeneous*, if $\mathcal{G}((Q, P, H)^{[c]}) \equiv \mathcal{G}_0, \forall c$ but $\mathcal{N}((Q, P, H)^{[c]})$ are different between certain $c \in \{1, \dots, C\}$.

Assumption 2. The underlying systems in multiple experiments, which provide $\{Y^{[c]}, U^{[c]}\}_{c=1,\dots,C}$, satisfy that $\mathcal{G}((Q, P, H)^{[c]}) \equiv \mathcal{G}_0$ for any $c = 1, \dots, C$.

The problem is to find a method to infer the dynamic network using the datasets from multiple experiments satisfying Assumption 2. In particular, we focus on the heterogeneous case.

3. NETWORK MODEL STRUCTURES

3.1 ARMAX model structure

Consider the network *model description* of (1) for system identification

$$y(t) = Q(q, \theta)y(t) + P(q, \theta)u(t) + H(q, \theta)e(t), \tag{2}$$

where θ is the model parameter. Its element-wise form is

$$y_i(t) = \sum_{j=1}^{p} Q_{ij}(q, \theta)y_j(t) + \sum_{k=1}^{m} P_{ik}(q, \theta)u_k(t) + H_{ii}(q, \theta)e_i(t). \tag{3}$$

We introduce ARMAX model for (3), which is given by

$$A_i(q)y_i(t) = \sum_{j=1}^{p} B^y_{ij}(q)y_j(t) + \sum_{k=1}^{m} B^u_{ik}(q)u_k(t) + C_{ii}(q)e_i(t),$$

where

$$\begin{aligned} A_i(q) &= 1 + a_{i1}\, q^{-1} + \dots + a_{in^a_i}\, q^{-n^a_i}, \\ B^y_{ij}(q) &= b^y_{ij1}\, q^{-1} + \dots + b^y_{ijn^{by}_{ij}}\, q^{-n^{by}_{ij}}, \\ B^u_{ij}(q) &= b^u_{ij1}\, q^{-1} + \dots + b^u_{ijn^{bu}_{ij}}\, q^{-n^{bu}_{ij}}, \\ C_i(q) &= 1 + c_{i1}\, q^{-1} + \dots + c_{in^c_i}\, q^{-n^c_i}. \end{aligned}$$

[1] Choosing $H(q)$ to be diagonal is due to network identifiability studied in Goncalves and Warnick (2008). See also Hayden et al. (2016).

[2] Here y_i, u_k are label names of the vertices, instead of signals $y_i(t), u_k(t)$.

[3] The definition is modified from the standard definition of *network* in the graph theory (e.g. (Diestel, 2006, p. 141)). The term “dynamic” is referred to that E, f are defined based on dynamical systems, and “linear” indicates (1) is for linear dynamical systems. The concepts of *source* and *sink* are not included due to little contribution to our study.

Hence the ARMAX model for (2) is

$$A(q)y(t) = B^y(q)y(t) + B^u(q)u(t) + C(q)e(t), \tag{4}$$

where

$$A \triangleq \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix}, \quad B^y \triangleq \begin{bmatrix} 0 & B^y_{12} & \cdots & B^y_{1p} \\ B^y_{21} & 0 & \dots & B^y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B^y_{p1} & B^y_{p2} & \cdots & 0 \end{bmatrix},$$
$$C \triangleq \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_p \end{bmatrix}, \quad B^u \triangleq \begin{bmatrix} B^u_{11} & B^u_{12} & \cdots & B^u_{1m} \\ B^u_{21} & B^u_{22} & \dots & B^u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B^u_{m1} & B^u_{m2} & \cdots & B^u_{mm} \end{bmatrix}. \tag{5}$$

It is easy to see

$$Q(q,\theta) = A^{-1}B^y, \quad P(q,\theta) = A^{-1}B^u, \quad H(q,\theta) = A^{-1}C. \tag{6}$$

Remark 3. The form (4) simplifies to be an ARX model if $C(q) \equiv I$; and it becomes an FIR model when $A(q) \equiv I$ in addition. Moreover, the polynomial orders $n^a_i, n^{by}_{ij}, n^{bu}_{ij}, n^c_i$ can be set to different values. However, in practice, we could start with setting $n^a_i \equiv n^a$, $n^{by}_{ij} \equiv n^{by}$, $n^{bu}_{ij} \equiv n^{bu}$, $n^c_i \equiv n^c$; and tune these orders differently over i, j if necessary.

3.2 Network predictor model

Consider the network model (1) and notice that $\big[I - Q(q)\big]$ is invertible. We have

$$\begin{aligned} y(t) &= \big(I-Q\big)^{-1}Pu(t) + \big(I-Q\big)^{-1}He(t) \\ &\triangleq G_u u(t) + G_e e(t). \end{aligned} \tag{7}$$

We refer to (Ljung, 1999, pp. 70) for the one-step-ahead prediction of y is

$$\hat{y}(t|t-1) = G_e^{-1}G_u u(t) + (I - G_e^{-1})y(t),$$

and thus the network *predictor model* of (2) is given by

$$\hat{y}(t|t-1) = H^{-1}Pu(t) + H^{-1}\big(Q + H - I\big)y(t). \tag{8}$$

The one-step-ahead predictor of the ARMAX model follows by substituting the expressions in (6)

$$\hat{y}(t|\theta) = C^{-1}B^u u(t) + (C^{-1}B^y + I - C^{-1}A)y(t), \tag{9}$$

where $\hat{y}(t|t-1) \triangleq \hat{y}(t|\theta)$ to emphasize the dependency on model parameters θ.

3.3 Regression forms

Rewriting (9) and adding $[I - C(q)]\hat{y}(t|\theta)$ to both sides, it yields

$$\begin{aligned} \hat{y}(t|\theta) = {} & B^u(q)u(t) + \big[B^y(q) - \big(A(q) - I\big)\big]y(t) + \\ & \big(C(q) - I\big)\big[y(t) - \hat{y}(t|\theta)\big]. \end{aligned} \tag{10}$$

To formulate a regression form, let us introduce the prediction error $\varepsilon(t|\theta) := y(t) - \hat{y}(t|\theta)$, and consider the prediction of the i-th output $y_i(t)$

$$\begin{aligned} \hat{y}_i(t|\theta) = {} & \bar{B}^u_i(q)u(t) + \big[\bar{B}^y_i(q) - \big(\bar{A}_i(q) - \bar{I}_i\big)\big]y(t) + \\ & \big(C_i(q) - \bar{I}_i\big)\big[y_i(t) - \hat{y}_i(t|\theta)\big], \end{aligned} \tag{11}$$

where $\bar{A}_i, \bar{B}^y_i, \bar{B}^u_i, \bar{I}_i$ are the corresponding i-th rows of A, B^y, B^u and I. Provided with the notations

$$\begin{aligned} \varphi(t,\theta_i) \triangleq \big[\; & y_1(t-1) \; \dots \; y_1(t - n^{by}_{i1}) \; \dots \\ & -y_i(t-1) \; \dots \; -y_i(t - n^a_i) \; \dots \\ & y_p(t-1) \; \dots \; y_p(t - n^{by}_{ip}) \\ & u_1(t-1) \; \dots \; u_1(t - n^{bu}_{i1}) \; \dots \\ & u_i(t-1) \; \dots \; u_i(t - n^{bu}_{ii}) \; \dots \\ & u_m(t-1) \; \dots \; u_m(t - n^{bu}_{im}) \\ & \varepsilon_i(t-1) \; \dots \; \varepsilon_i(t - n^c_i) \; \big]^T \end{aligned} \tag{12}$$

and

$$\begin{aligned} \theta_i \triangleq \Big[& \boxed{b^y_{i11} \; \cdots \; b^y_{i1n^{by}_{i1}}} \cdots \boxed{a_{i1} \; \cdots \; a_{in^a_i}} \cdots \\ & \boxed{b^y_{ip1} \; \cdots \; b^y_{ipn^{by}_{ip}}} \\ & \boxed{b^u_{i11} \; \cdots \; b^u_{i1n^{bu}_{i1}}} \cdots \boxed{b^u_{ii1} \cdots b^u_{iin^{bu}_{ii}}} \cdots \\ & \boxed{b^u_{im1} \cdots b^u_{imn^{bu}_{im}}} \\ & \boxed{c_{i1} \; \cdots \; c_{in^c_i}} \Big]^T, \quad (N \text{ blocks}) \end{aligned} \tag{13}$$

where $N = p + m + 1$, we obtain a pseudo-linear regression form

$$\hat{y}_i(t|\theta_i) = \varphi^T(t,\theta_i)\theta_i, \quad i = 1, \dots, p. \tag{14}$$

Remark 4. Note that there is an important link between the framed parameter blocks in (13) and the network. Each directed link in the digraph corresponds to a linear dynamic system from an input u_j or an output y_j to an output y_i. The parameters of this linear system are given in the block with parameters $b^u_{ij\cdot}$ or $b^y_{ij\cdot}$ together with $a_{i\cdot}$. We will later (in (17) and (21)) denote these parameter blocks by $\mathbf{w}^{[c]}_k$ or $\mathbf{w}_k$ with a numbering k (see Figure 3 for an example).

4. HETEROGENEOUS DATASETS

4.1 Regression forms of multiple datasets

Considering the regression form for network inference, since the p regression problems are independent, the whole network can be inferred by formulating and solving (14) for p output variables. Therefore, without loss of generality, it is assumed in the later sections that we are dealing with the i-th output variable y_i. Thus, for simplicity, we introduce the following notations

$$\mathbf{y}^{[c]} \triangleq \begin{bmatrix} y_i(t_1|\theta_i) \\ \vdots \\ y_i(t_M|\theta_i) \end{bmatrix}, \quad \mathbf{A}^{[c]}(\mathbf{w}^{[c]}) \triangleq \begin{bmatrix} \varphi^T(t_1,\theta_i) \\ \vdots \\ \varphi^T(t_M,\theta_i) \end{bmatrix}, \tag{15}$$

where $\mathbf{w}^{[c]} \triangleq \theta_i$[4], and (14) is evaluated at $\{t_1, \dots, t_M\}$. Furthermore, we rewrite into block matrices

$$\mathbf{y}^{[c]} = \mathbf{A}^{[c]}(\mathbf{w}^{[c]})\,\mathbf{w}^{[c]} + \boldsymbol{\xi}^{[c]}, \quad c = 1, \dots, C, \tag{16}$$

where

$$\begin{aligned} \mathbf{A}^{[c]} &\triangleq \Big[\mathbf{A}^{[c]}_{:,1} \quad \mathbf{A}^{[c]}_{:,2} \quad \cdots \quad \mathbf{A}^{[c]}_{:,N}\Big], \\ \mathbf{w}^{[c]} &\triangleq \Big[\big(\mathbf{w}^{[c]}_1\big)^T \quad \cdots \; \big(\mathbf{w}^{[c]}_i\big)^T \; \cdots \; \big(\mathbf{w}^{[c]}_p\big)^T, \\ & \qquad \big(\mathbf{w}^{[c]}_{p+1}\big)^T \; \cdots \; \big(\mathbf{w}^{[c]}_{p+i}\big)^T \; \cdots \; \big(\mathbf{w}^{[c]}_{p+m}\big)^T, \\ & \qquad \big(\mathbf{w}^{[c]}_N\big)^T\Big]^T \\ \boldsymbol{\xi}^{[c]} &\triangleq \big[\xi^{[c]}(t_1) \quad \xi^{[c]}(t_2) \quad \cdots \quad \xi^{[c]}(t_M)\big], \end{aligned} \tag{17}$$

and $\mathbf{w}^{[c]}$ is partitioned into N blocks as illustrated in (13), and $\boldsymbol{\xi}^{[c]}$ denotes the prediction error, which represents the part of the output $\mathbf{y}^{[c]}$ that cannot be predicted from past data. Note that the blocks may have different dimensions due to the general setup of the ARMAX model (see Remark 3).

[4] Here θ_i represents the system parameters of the underlying model in the c-th experiment.

Letting

$$\mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_k^{[1]} \\ \hline \vdots \\ \hline \mathbf{w}_k^{[C]} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \hline \vdots \\ \hline \mathbf{w}_N \end{bmatrix}, \tag{18}$$

we integrate all datasets by stacking (16) for each dataset and rearranging blocks of matrices, yielding (19), and, for simplicity, use $\mathbf{y} = \mathbf{A}(\mathbf{w})\mathbf{w} + \boldsymbol{\xi}$ to denote (19b).

Remark 5. When experiments are perfectly repeated, i.e. the homogeneous case (refer to Section 2.2), we have the ideal case $\mathbf{w}^{[1]} = \cdots = \mathbf{w}^{[C]} \equiv \mathbf{w}$. A single linear regression form is formulated for identification by concatenation:

$$\begin{bmatrix} \mathbf{y}^{[1]} \\ \vdots \\ \mathbf{y}^{[C]} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{[1]}(\mathbf{w}) \\ \vdots \\ \mathbf{A}^{[C]}(\mathbf{w}) \end{bmatrix} \mathbf{w} + \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \vdots \\ \boldsymbol{\xi}^{[C]} \end{bmatrix}. \tag{20}$$

This treatment can also be used when $\mathbf{w}^{[c]}$'s are not significantly different, e.g. being perturbed by white noise.

4.2 Simultaneous sparsity regularization

Now we consider the two essential requirements for network inference from heterogeneous datasets: 1) sparse networks is acquired in the presence of noise; 2) $\mathbf{w}^{[c]}$ is required to give the same network topology for all c, i.e. the inference results $(\hat{Q}, \hat{P}, \hat{H})^{[c]}$ satisfy $\mathcal{G}((\hat{Q}, \hat{P}, \hat{H})^{[c]}) \equiv \mathcal{G}^0, \forall c$ (see Section 2.2). The example in Figure 3 helps to understand.

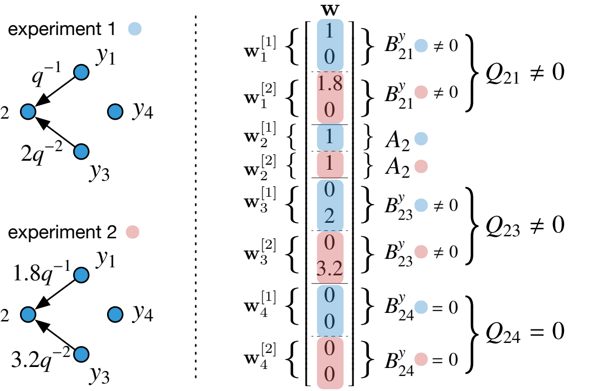


Fig. 3. An example of $\mathbf{w}$ in the setup of multiple experiments.

Let us introduce a term of group sparsity based on $\mathbf{w}$,

$$\mathbf{w}^S := [\|\mathbf{w}_1\|_2, \ \cdots, \ \|\mathbf{w}_N\|_2]^T, \tag{21}$$

in which $\mathbf{w}^S \in \mathbb{R}^N$, $\|\cdot\|_2$ the l_2-norm of vectors, N the number of large groups (see (13)). The dimensions of each block $\mathbf{w}_k^{[c]}$ and $\mathbf{w}_k$ in (18) are saved in two vectors ρ^E, ρ^S, respectively

$$\begin{aligned} &\rho \triangleq \left[n_{i1}^{by}, \dots, n_i^a, \dots, n_{ip}^{by}, n_{i1}^{bu}, \dots, n_{ip}^{bu}, n_i^c\right]^T, \\ &\rho^E := \rho \otimes \mathbf{1}_C, \qquad \rho^S := C\rho, \end{aligned} \tag{22}$$

where the elements of ρ are defined in (13), $\mathbf{1}_C$ is a C-dimensional column vector of 1's, and the index i in ρ indicates that we are dealing with i-th output $y_i(t)$.

Group sparsity is needed such that the sparsity of networks is guaranteed and the network topology is consistent over replica (i.e. the interconnection structure determined by the $\mathbf{w}^{[c]}$'s are identical). The sparsity is imposed on each large group $\mathbf{w}_k$, $k = 1, \dots, N$, and the penalty term is $\lambda\|\mathbf{w}^S\|_0$, where $\lambda \in \mathbb{R}^+$. The mechanism on how the group sparsity functions is described as follows.

Recall that the setup (19) allows the system parameters to be different in values for different c's. Note that each small block $\mathbf{w}_k^{[c]}$ corresponds to an arc in the underlying digraph of the dynamical system in the c-th experiment (see an example Figure 3). The $\|\mathbf{w}_k\|_2$ chosen to be zero yields that all $\mathbf{w}_k^{[c]}, c = 1, \dots, C$ equals zero. It implies that the arc corresponding to $\mathbf{w}_k^{[c]}$ does not exist in dynamic networks for any c. In addition, thanks to the effect of noise, it is nearly guaranteed that $\mathbf{w}_k^{[c]}$ is not identical to zero for almost all c if $\|\mathbf{w}_k\|_2 \neq 0$. This is how the group sparsity (defined via $\mathbf{w}^S$) guarantees that the resultant networks of different datasets share the same topology. Moreover, when a classical least squares objective is augmented with a penalty term of $\lambda\|\mathbf{w}^S\|_0$, the optimal solution favors zeros of $\mathbf{w}_k, k = 1, \dots, N$, which guarantees the sparsity of network structures.

In summary, to perform dynamic network reconstruction from noisy heterogeneous datasets, we can solve the following optimization problem

$$\underset{\mathbf{w}}{\text{minimize}} \ \|\mathbf{y} - \mathbf{A}(\mathbf{w})\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}^S\|_0. \tag{23}$$

5. A TREATMENT BY CLASSICAL L_1/L_2-METHODS

The section considers how to solve the problem (23) under different parametrization models. Due to the page limitation, we present treatments to the ARX models. See Yue et al. (2016) for a complementary discussion on ARMAX models and another treatment by *Sparse Bayesian Learning* in the Bayesian perspective on statistical estimation.

5.1 A fundamental case: ARX models

As addressed in Section 3.3, choosing ARX forms for network parametric models results in a linear regression form, in which $\mathbf{A}$ does not depend on $\mathbf{w}$ in (19). The treatment of classical group LASSO yields

$$\underset{\mathbf{w}}{\text{minimize}} \ \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}^S\|_1, \tag{24}$$

where

$$\lambda\|\mathbf{w}^S\|_1 = \lambda \sum_{i=1}^{N} \sqrt{\rho_i^S}\,\|\mathbf{w}_i\|_2. \tag{25}$$

This is a convex optimization and has been soundly studied in Yuan and Lin (2006).

To achieve a better approximation of the l_0-norm, alternatively one may use *Iterative Reweighted* l_1/l_2 *Methods* (e.g. see Candes et al. (2008); Chartrand and Yin (2008)). When being applied to group sparsity, both methods turn to be a similar scheme (differing in the usage of $\|\cdot\|_2$ or $\|\cdot\|_2^2$ for blocks of $\mathbf{w}$ in (26) and (27)). Here we present the solution using the l_1 method.

$$\mathbf{w}^{(k+1)} \leftarrow \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^{N} \nu_i^{(k)} \sqrt{\rho_i^S}\,\|\mathbf{w}_i\|_2, \tag{26}$$

where

$$\nu_i^{(k)} \leftarrow \left[\|\mathbf{w}_i^{(k)}\|_2 + \epsilon^{(k)}\right]^{-1}, \tag{27}$$

where k is the index of iterations. In regard to the selection of ϵ, $\{\epsilon^{(k)}\}$ should be a sequence converging to zero, as addressed in

$$\begin{bmatrix}\mathbf{y}^{[1]}\\ \vdots\\ \mathbf{y}^{[C]}\end{bmatrix} = \underbrace{\left[\begin{array}{c|c|c} \mathbf{A}^{[1]}_{:,1}(\mathbf{w}^{[1]})\ \dots\ \mathbf{A}^{[1]}_{:,N}(\mathbf{w}^{[1]}) & & \\ & \ddots & \\ & & \mathbf{A}^{[C]}_{:,1}(\mathbf{w}^{[C]})\ \dots\ \mathbf{A}^{[C]}_{:,N}(\mathbf{w}^{[C]})\end{array}\right]}_{C\ \textbf{Blocks}} \left[\begin{array}{c}\mathbf{w}^{[1]}\\ \hline \vdots \\ \hline \mathbf{w}^{[C]}\end{array}\right] + \left[\begin{array}{c}\boldsymbol{\xi}^{[1]}\\ \hline \vdots\\ \hline \boldsymbol{\xi}^{[C]}\end{array}\right] \tag{19a}$$

$$= \underbrace{\left[\begin{array}{c|c|c} \begin{matrix}\mathbf{A}^{[1]}_{:,1}(\mathbf{w}^{[1]}) & & \\ & \ddots & \\ & & \mathbf{A}^{[C]}_{:,1}(\mathbf{w}^{[C]})\end{matrix} & \dots & \begin{matrix}\mathbf{A}^{[1]}_{:,N}(\mathbf{w}^{[1]}) & & \\ & \ddots & \\ & & \mathbf{A}^{[C]}_{:,N}(\mathbf{w}^{[C]})\end{matrix}\end{array}\right]}_{N\ \textbf{Blocks}} \left[\begin{array}{c}\mathbf{w}_1\\ \hline \vdots \\ \hline \mathbf{w}_N\end{array}\right] + \left[\begin{array}{c}\boldsymbol{\xi}^{[1]}\\ \hline \vdots\\ \hline \boldsymbol{\xi}^{[C]}\end{array}\right] \tag{19b}$$

Chartrand and Yin (2008) based on the *Unique Representation Property*. It suggests in Chartrand and Yin (2008) a fairly simple update rule of ϵ, i.e. $\epsilon^{(k)} \in (0,1)$ is reduced by a factor of 10 until reaching a minimum of 10^{-8} (the factor and lower bound could be tuned to fit specific problems). One may also adopt the adaptive rule of ϵ given in Candes et al. (2008).

5.2 *Fast implementations via Proximal Methods and ADMM*

To solve the convex optimization in Section 5.1, for example, *CVX* for MATLAB could be an easy solution. However, the performance is not promising for large-dimension problems. This section presents algorithms using *Proximal Methods* and *ADMM* (Parikh and Boyd (2013)) to handle large-dimension network reconstruction problems.

Let us first consider (24), which is rewritten as

$$\underset{\mathbf{w}}{\text{minimize}}\ f(\mathbf{w}) + g(\mathbf{w}), \tag{28}$$

where $f(\mathbf{w}) \triangleq (1/2)\|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2$, $g(\mathbf{w}) \triangleq \lambda\|\mathbf{w}^S\|_1$, λ is twice larger than the value in (25). Given $\nabla f(\mathbf{w}) = \mathbf{A}^T(\mathbf{A}\mathbf{w} - \mathbf{y})$, the *Proximal Gradient Method* is to update $\mathbf{w}$ by $\mathbf{w}^{k+1} = \mathbf{prox}_{\gamma g}(\mathbf{w}^k - \gamma\nabla f(\mathbf{w}^k))$, $\gamma \in \mathbb{R}_+$, where k denotes the iteration index. It is easy to see that $g(\mathbf{w}) = \sum_{i=1}^N g_i(\mathbf{w}_i)$, where $g_i(\mathbf{w}_i) := \lambda\sqrt{\rho_i^S}\|\mathbf{w}_i\|_2$. Firstly we partition the variable $\mathbf{v}$ of $\mathbf{prox}_{\gamma g}(\mathbf{v})$ in the same way as $\mathbf{w}$ in terms of $\mathbf{w}_i, i = 1,\dots,N$, i.e. $\mathbf{v} = [\mathbf{v}_1^T,\dots,\mathbf{v}_N^T]^T$. Then we calculate the proximal operator $\mathbf{prox}_{\gamma g_i}(\mathbf{v}_i)$, which equals

$$\mathbf{prox}_{\gamma g_i}(\mathbf{v}_i) = \left(1 - \gamma\lambda\sqrt{\rho_i^S}/\|\mathbf{v}_i\|_2\right)_+ \mathbf{v}_i, \tag{29}$$

where $(\cdot)_+$ replaces each negative elements with 0. It follows that

$$\mathbf{prox}_{\gamma g}(\mathbf{v}) = \left[\left(\mathbf{prox}_{\gamma g_1}(\mathbf{v}_1)\right)^T \cdots \left(\mathbf{prox}_{\gamma g_N}(\mathbf{v}_N)\right)^T\right]^T. \tag{30}$$

The value of γ needs to be selected appropriately so as to guarantee the convergence. One simple solution is using line search methods, e.g. see Section 4.2 in Parikh and Boyd (2013).

Provided with the above calculations, it is straightforward to implement the *(Accelerated) Proximal Gradient Method* (see Yue et al. (2016) for details). To implement ADMM, the proximal operator of $f(\mathbf{w})$ needs to be calculated,

$$\mathbf{prox}_{\gamma f}(\mathbf{v}) = (I + \gamma\mathbf{A}^T\mathbf{A})^{-1}(\gamma\mathbf{A}^T\mathbf{y} + \mathbf{v}). \tag{31}$$

Given $\mathbf{prox}_{\gamma g}(\mathbf{v})$ as (29) and (30), the ADMM method is presented in Algorithm 1.

To use this algorithm for the iterative reweighted l_1 method (26), we only need to modify (29), which now should be

Algorithm 1 ADMM method

```
Precompute $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{y}$
**given** an initial value $\mathbf{w}^0, \mathbf{z}^0, \mathbf{u}^0$, $\gamma^0 = 1$, and $\beta = 1/2$
**repeat**
    $\gamma \leftarrow \gamma^k$
    **repeat**
        $\hat{\mathbf{w}} \leftarrow \mathbf{prox}_{\gamma f}(\mathbf{z}^k - \mathbf{u}^k)$ using (31)
        **break** if $f(\hat{\mathbf{w}}) \le f(\mathbf{w}^k) + \nabla f(\mathbf{w}^k)^T(\hat{\mathbf{w}} - \mathbf{w}^k) + (1/2\gamma)\|\hat{\mathbf{w}} - \mathbf{w}^k\|_2^2$
        $\gamma \leftarrow \beta\gamma$
    **until** ;
    $\mathbf{w}^{k+1} \leftarrow \hat{\mathbf{w}}$, $\gamma^{k+1} \leftarrow \gamma$
    Compute $\mathbf{prox}_{\gamma g_i}(\mathbf{w}_i^{k+1} + \mathbf{u}_i^k)$ by (29) for $i = 1, ..., N$
    $\mathbf{z}^{k+1} \leftarrow \mathbf{prox}_{\gamma g}(\mathbf{w}^{k+1} + \mathbf{u}^k)$ using (30)
    $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k + \mathbf{w}^{k+1} - \mathbf{z}^{k+1}$
**until** any standard stopping criteria
```

$$\mathbf{prox}_{\gamma g_i}(\mathbf{v}_i) = \left(1 - \gamma\lambda\nu_i\sqrt{\rho_i^S}/\|\mathbf{v}_i\|_2\right)_+ \mathbf{v}_i. \tag{32}$$

In each "outer" loop indicated by (26), we update ν_i by (27) and implement ADMM as Algorithm 1 to solve (26).

6. NUMERICAL EXAMPLES

We consider a Monte Carlo study of 50 runs where *random stable sparse* networks of 40 nodes are simulated and inferred using the proposed methods. In regard to the adjective words for networks, here are further explanations:

- *random*: the DSF model in each run are randomly chosen (both network topology and model parameters);
- *stable*: each DSF model is stable, i.e. all transfer-function elements in (Q, P, H) are stable;
- *sparse*: the number of edges of the network is much less than that of its complete digraph.

See Yue et al. (2016) for details on the setup of random network models with random network topology. Our setup of systems (networks) makes network inference particularly challenging. In these networks, there exist many feedback loops, whose sizes are quite random. Moreover, the networks cannot be decoupled into smaller unconnected components.

The following performance indices are used to benchmark our algorithms. As analogous to concepts in statistics, *Type-I error* is asserting arcs that is absent (a false hit) and *Type-II error* is failing to assert arcs that are present (a miss), which are defined as follows

$$\text{Type-I error} = \frac{\text{FP}}{\text{TP + FP}}, \quad \text{Type-II error} = \frac{\text{FN}}{\text{TP + FN}}.$$

Here TP (true-positive), FP (false-positive) and FN (false-negative) are the standard concepts in the *Precision-Recall* curve (e.g. see Sahiner et al. (2017)).

As known in biological data analysis, the time series are usually of low sampling frequencies and limited numbers of samples. To address the importance of these factors, we run network inference methods (the ARX case) on a range of their values, shown in Figure 4. The sampling frequency is critical for applying discrete-time approaches for network inference, since the network topology from discrete-time systems will more and more different from the ground truth that is defined by the underlying continuous-time systems, with the decrease of sampling frequencies. The rule of thumb is choosing the sample frequency that is at least 10 time faster than the critical sampling frequency of system aliasing [5] (e.g. $f_s/4$ in Figure 4 is this suggested value). The sparsity is to handle the effect of noise. The simulation tells us that the value that is at least four times larger than the number of unknown parameters in estimation is a fair choice for the number of samples in network inference.

Another comment is for the "trade-off" between type-I and type-II errors when selecting regularization parameters λ. In theory, there could be an optimal value of λ that gives small values of both type-I and type-II errors. However, in practice, type-I error is more critical in the sense that it has to be small enough to keep results useful. Otherwise, even if type-II error is small, the result will predicate too many wrong arcs to be useful in applications. As a rule implied from Figure 4, in biological practice, we may choose an aggressive value of λ to make sure that we could have most predictions of arcs correctly; then, if more links need to be explored, we could decrease λ to get more connections covered.

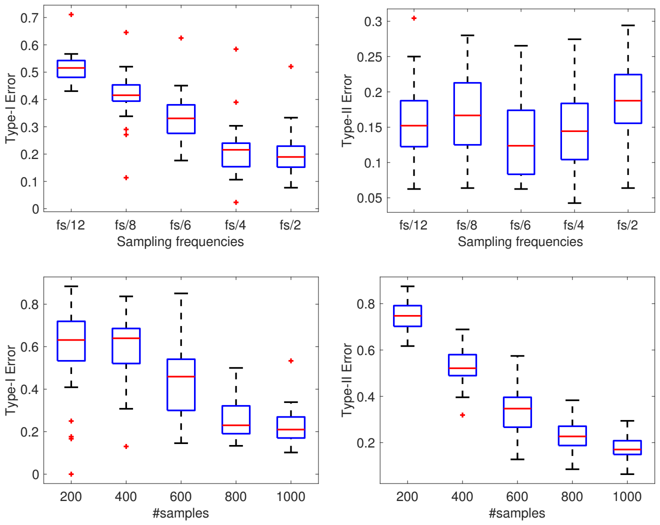


Fig. 4. Performance of the proposed method on 50 random networks. See Yue et al. (2016) for further details.

7. CONCLUSIONS

Large-dimensional linear dynamic network reconstruction from heterogeneous datasets in the framework of Dynamical Structure Function (or P. Van den Hof's network representation) has been discussed. It has been addressed that the linear dynamic network reconstruction can be formulated as identification of DSF with sparse structures. To take advantage of heterogeneous datasets from multiple experiments, the proposed method integrates all datasets in one regression form and resorts to group sparsity to guarantee network topology to be consistent over replica. To solve the cardinality optimization problem, the treatment of classical l_1/l_2 heuristic methods has been introduced. In the numerical examples, we have shown the performance of methods and pointed out several factors that should be considered in network reconstruction applications.

REFERENCES

Bar-Joseph, Z., Gitter, A., and Simon, I. (2012). Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8), 552–564.

Beal, M.J., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D.L. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3), 349–356.

Candes, E.J., Wakin, M.B., and Boyd, S.P. (2008). Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6), 877–905.

Chartrand, R. and Yin, W. (2008). Iteratively reweighted algorithms for compressive sensing. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, 3869–3872. IEEE.

Chiuso, A. and Pillonetto, G. (2012). A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8), 1553–1565.

Diestel, R. (2006). *Graph theory*. Graduate Texts in Mathematics. Springer, Berlin, 3rd edition.

Eichler, M. (2007). Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 137, 334–353. doi: 10.1016/j.jeconom.2005.06.032.

Goncalves, J. and Warnick, S. (2008). Necessary and Sufficient Conditions for Dynamical Structure Reconstruction of LTI Networks. *Automatic Control, IEEE Transactions on*, 53(7), 1670–1674.

Granger, C.W.J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 424–438.

Hayden, D., Yuan, Y., and Goncalves, J. (2016). Network Identifiability from Intrinsic Noise. *IEEE Transactions on Automatic Control*, PP(99), 1. doi: 10.1109/TAC.2016.2640219.

He, F., Chen, H., Probst-Kepper, M., Geffers, R., Eifes, S., del Sol, A., Schughart, K., Zeng, A., and Balling, R. (2012). PLAU inferred from a correlation network is critical for suppressor function of regulatory T cells. *Molecular Systems Biology*, 8(1).

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice-Hall information and system sciences series. Prentice Hall PTR.

Murphy, K. and Mian, S. (1999). Modelling gene expression data using dynamic Bayesian networks. Technical report.

Pan, W., Yuan, Y., Ljung, L., Gon, J., and Stan, G.B. (2015). Identifying biochemical reaction networks from heterogeneous datasets. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 2525–2530. IEEE.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in optimization*, 1(3), 123–231.

Pearl, J. (1995). Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Synthese-Dordrecht*, 104(1), 161.

Sahiner, B., Chen, W., Pezeshk, A., and Petrick, N. (2017). Comparison of two classifiers when the data sets are imbalanced: the power of the area under the precision-recall curve as the figure of merit versus the area under the ROC curve. In *SPIE Medical Imaging*, 101360G–101360G. International Society for Optics and Photonics.

Weerts, H.H.M., Dankers, A.G., and Van den Hof, P.M.J. (2015). Identifiability in dynamic network identification. *IFAC-PapersOnLine*, 48(28), 1409–1414.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49–67.

Yue, Z., Pan, W., Thunberg, J., Ljung, L., and Goncalves, J. (2016). Linear Dynamic Network Reconstruction from Heterogeneous Datasets. *arXiv preprint*.

[5] This is only theoretically useful since we have not yet known how to access the critical frequency of system aliasing without the ground truth.