A LETTERS JOURNAL EXPLORING THE FRONTIERS OF PHYSICS

January 2016

EPL, **113** (2016) 18005
doi: 10.1209/0295-5075/113/18005

www.epljournal.org

Depicting network structures from variable data produced by unknown colored-noise driven dynamics

YANG CHEN[1], SHIHONG WANG[1], ZHIGANG ZHENG[2], ZHAOYANG ZHANG[2(a)] and GANG HU[2(b)]

[1] *School of Sciences, Beijing University of Posts and Telecommunications - Beijing, China*
[2] *Department of Physics, Beijing Normal University - Beijing, China*

received 30 July 2015; accepted in final form 7 January 2016
published online 28 January 2016

PACS 89.75.Hc – Networks and genealogical trees
PACS 05.10.Gg – Stochastic analysis methods (Fokker-Planck, Langevin, etc.)
PACS 05.45.Tp – Time series analysis

Abstract – In recent decades, the topic of depicting network structures from output variable data, *i.e.*, the inverse problem, turns to be a key issue in wide interdisciplinary areas, in particular, in biological and social fields. Noise inevitably exists in practical dynamic networks, and the output data are often generated via interplay between noise and network structures. The essential difficulty to solve the inverse problem is how to extract information of node links in networks under unknown and possibly strong noise. In this paper, based on the idea that the output variable data contain information not only for network topology but also for noise, we propose a method to deal with this problem, incorporating three crucial ingredients: Computing multiple matrices to extract as much as possible information on network topology and noise statistics; making a systematical matrix algebraic computation to obtain equations closed for network inference; using an effective iteration algorithm to solve the resulting nonlinear matrix equations. The above theory is established in an accurate and closed form, numerical computations convincingly verify the validity of theoretical analysis, and the possible applications in practical inverse problems are emphasized.

Copyright © EPLA, 2016

Today huge amounts of data have been accumulated in databases and more and more data are continually produced on purposes in various fields. How to extract as much as possible information from these data turns to be a central issue in almost all fields in natural and social sciences. Available data can be produced from sources of great diversity. In biological and social systems scientists are often faced with a crucial problem: data are produced by various dynamic networks, such as neuron networks [1,2], gene regulating networks [3,4], life cycle networks [5], financial and business networks [6] and so on, while the network structures generating these data are unknown. Depicting network structures from available variable data, *i.e.*, the inverse problem, has become a greatly attracting task in wide interdisciplinary fields, in particular, in the applications of many practically important social and biological systems [1,7–11].

Theoretically, different methods have been proposed to solve various inverse problems. According to different data requirements and network conditions all methods are roughly classified into two categories. The methods of the first class can perform rather accurate inference computation and depict both the natures and intensities of network interactions. For the price of this accuracy, they require much information, besides the output variable data, *e.g.*, those about internal network dynamics [12,13]. In [14] (also in [15]) authors proposed an interesting idea of depicting network structure through the bridge of noise, where the matrix of noise correlation should be known for the successful inference computation. The methods of the second class can make inference computation rather robustly with output variable data only [16,17]. However, as a fair trade-off, these methods have very low accuracy: most of them cannot make quantitative inferences of network interactions, and even qualitatively the probability of correct predictions are often rather low [18]. These methods fail definitely if noise is very strong and plays a key role in producing variable data [19].

Our task in solving the inverse problem is to enjoy the advantages of both classes of methods and avoid their own disadvantages. Precisely, on the one hand, we hope to

(a) E-mail: juycezhang@163.com
(b) E-mail: ganghu@bnu.edu.cn

solve the inverse problem with data of the observables only, without requiring all other information about noise correlations and other additional information of network dynamics, so that the method can be extensively applied to most of practical inference performances. On the other hand, we are aiming at accurate depictions and looking for inferring network interactions not only for qualitative natures, but also for quantitative values of link strengths. By accurate depiction, we mean controllable inference precision by the quality of variable data, such as measurement frequency, sample set size of variable data and so on. The only way to achieve this goal is to extract more information from the available variable data than all the above-mentioned previous methods, so that the additionally extracted information can reveal more about both unknown network structure and unknown noise property.

A nonlinear dynamic network can be generally linearized around a certain space point as [19]

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \hat{\boldsymbol{A}}\boldsymbol{x}(t) + \boldsymbol{\eta}(t), \\ \boldsymbol{x}(t) &= (x_1(t), x_2(t), \cdots, x_N(t))^T, \end{aligned} \tag{1}$$

where $\hat{\boldsymbol{A}}$ is a $N \times N$ constant matrix, the target of the inverse problem, $\boldsymbol{\eta}(t) = (\eta_1(t), \eta_2(t), \cdots, \eta_N(t))^T$ is a vector of inevitable random driving forces (noises) involved in the network dynamics. Suppose all the output variable time sequences $x_i(t_k)$ for $i = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, L$ are available with sufficiently high measurement frequency (*i.e.*, $0 < \Delta t_k = t_{k+1} - t_k = \Delta t \ll 1$) and sufficiently many data accumulation ($L\Delta t \gg 1$). Two points should be emphasized about eq. (1). First, the matrix elements A_{ij} represent the network structure, including interactions between nodes ($A_{ij}, i \neq j$) and local dynamics of nodes ($A_{ii},\ i = 1, \cdots, N$), determing the full dynamics of the whole network (in the linearized version). Second, though eq. (1) is linear, it can be used to discuss the inverse problem of nonlinear networks around different phase space points, including oscillatory and chaotic networks. The detail is discussed in [19] for white-noise driven systems.

Our task is to depict matrix $\hat{\boldsymbol{A}}$ from the available data $\boldsymbol{x}(t)$ only. A standard algorithm is to multiply both sides of eq. (1) by the variable $\boldsymbol{x}^T(t)$ and compute the corresponding correlation matrices. The matrix algebra thus reads

$$\hat{\boldsymbol{B}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{C}} + \hat{\boldsymbol{H}}_1 \tag{2}$$

with

$$\hat{\boldsymbol{B}} = \langle \dot{\boldsymbol{x}}\boldsymbol{x}^T \rangle, \quad \hat{\boldsymbol{C}} = \langle \boldsymbol{x}\boldsymbol{x}^T \rangle, \quad \hat{\boldsymbol{H}}_1 = \langle \boldsymbol{\eta}\boldsymbol{x}^T \rangle,$$

where any above matrix elements is defined as

$$\begin{aligned} \dot{\boldsymbol{x}}(t_k) &= \frac{\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k)}{\Delta t}, \\ \hat{\boldsymbol{P}} &= \langle \boldsymbol{y}\boldsymbol{z}^T \rangle, \quad P_{ij} = \frac{1}{L}\sum_{k=1}^{L} y_i(t_k) z_j(t_k). \end{aligned} \tag{3}$$

From eq. (2) the inverse problem can be solved as

$$\hat{\boldsymbol{A}} = (\hat{\boldsymbol{B}} - \hat{\boldsymbol{H}}_1)\hat{\boldsymbol{C}}^{-1}. \tag{4}$$

In [20] authors depicted $\hat{\boldsymbol{A}}$ by considering $\boldsymbol{\eta} = \hat{\boldsymbol{H}}_1 = 0$. In [19] authors identified $\hat{\boldsymbol{H}}_1 = 0$ under white-noise approximation by making a noise-variable decorrelation technique. In all the above cases, one should know all correlation matrices in the right side of eq. (2), *i.e.*, one should know additional information to the observable variable data. Now a practical problem is: If noise is crucial for producing data and thus not negligible ($\hat{\boldsymbol{H}}_1 = O(1) \neq 0$), and if the noise is unknown with all its intensity and time correlations, how can one solve the matrix algebraic equation to depict topology $\hat{\boldsymbol{A}}$ without any additional information to $\boldsymbol{x}(t)$ data?

In all practical systems noise has finite correlation time. Then we generally consider in eq. (1) Gaussian colored noise $\boldsymbol{\eta}(t)$ as

$$\begin{aligned} \boldsymbol{\eta} &= \hat{\boldsymbol{M}}\boldsymbol{\epsilon}, \quad \hat{\boldsymbol{M}} \in \mathbb{R}^{N\times N}, \quad \boldsymbol{\epsilon} \in \mathbb{R}^{N\times 1}, \\ \langle \epsilon_i(t)\epsilon_j(t-\Delta t) \rangle &= d_i \delta_{ij} \exp(-r_i \Delta t). \end{aligned} \tag{5}$$

It is also well known that the Gausian colored noise $\boldsymbol{\eta}$ can be represented by a white-noise formulism as [21]

$$\dot{\boldsymbol{\eta}}(t) = -\hat{\boldsymbol{R}}\boldsymbol{\eta}(t) + \boldsymbol{\Gamma}(t), \qquad \hat{\boldsymbol{R}} = \hat{\boldsymbol{M}}\hat{\boldsymbol{r}}\hat{\boldsymbol{M}}^{-1} \tag{6}$$

with $r_{ij} = r_i\delta_{ij}$, $\langle \Gamma_i(t)\Gamma_j(t') \rangle = Q_{ij}\delta(t-t') = (\hat{\boldsymbol{M}}\hat{\boldsymbol{d}}\hat{\boldsymbol{M}}^T)_{ij}$ and $\hat{\boldsymbol{r}}, \hat{\boldsymbol{d}}, \hat{\boldsymbol{Q}}, \hat{\boldsymbol{R}} \in \mathbb{R}^{N\times N}$. Since noise is an uncontrollable fact in network dynamics, all matrix elements R_{ij} and Q_{ij} above are unknown. And thus the matrix $\hat{\boldsymbol{H}}_1$ does not vanish and it is unknown. At the first glimpse on eq. (2), the above-mentioned task is impossible to be fulfilled for the number of unknown matrices is more than the number of matrix equation. The only way to solve the difficulty in eq. (2) is to extract more information, besides $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$, from data sequences $x_i(t), i = 1, 2, \cdots, N$, since the sequences $x_i(t)$'s contain surely not only the information of matrix $\hat{\boldsymbol{A}}$, but also of the noise correlation $\hat{\boldsymbol{H}}_1$. For doing so we make a systematical matrix computation.

For extracting more from the $\boldsymbol{x}(t)$ data, a natural further step is to compute the derivatives of eq. (1) as

$$\ddot{\boldsymbol{x}}(t) = \hat{\boldsymbol{A}}\dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{\eta}}(t) \tag{7}$$

with

$$\begin{aligned} \ddot{\boldsymbol{x}}(t_k) &= \frac{\dot{\boldsymbol{x}}(t_{k+1}) - \dot{\boldsymbol{x}}(t_k)}{\Delta t} = \frac{\boldsymbol{x}(t_{k+2}) + \boldsymbol{x}(t_k) - 2\boldsymbol{x}(t_{k+1})}{\Delta t^2}, \\ \dot{\boldsymbol{\eta}}(t_k) &= \frac{\boldsymbol{\eta}(t_{k+1}) - \boldsymbol{\eta}(t_k)}{\Delta t}. \end{aligned} \tag{8}$$

Moreover we can further obtain another matrix equation by multiplying eq. (7) by $\boldsymbol{x}^T(t)$ and computing correlation matrices as

$$\hat{\boldsymbol{D}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{B}} + \hat{\boldsymbol{H}}_2 \tag{9}$$

with $\hat{\boldsymbol{D}} = \langle \ddot{\boldsymbol{x}}\boldsymbol{x}^T \rangle, \hat{\boldsymbol{H}}_2 = \langle \dot{\boldsymbol{\eta}}\boldsymbol{x}^T \rangle$, where $\hat{\boldsymbol{D}}$ can be identified solely by $\boldsymbol{x}(t)$ while matrix $\hat{\boldsymbol{H}}_2$ becomes another unknown matrix, which has to be determined, too. It is emphasized that matrix $\hat{\boldsymbol{D}}$ contains new information on the available data $\boldsymbol{x}(t)$'s which is not contained by matrices $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$. Here $\hat{\boldsymbol{C}}$ represents the variable-variable correlation $\hat{\boldsymbol{C}} = \langle \boldsymbol{x}\boldsymbol{x}^T \rangle$, which does not include any information on the variable variation rates which is included by velocity-variable correlations $\hat{\boldsymbol{B}} = \langle \dot{\boldsymbol{x}}\boldsymbol{x}^T \rangle$. Similarly, $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{B}}$ matrices do not contain any information on the velocity variation rates, which is included by the matrix $\hat{\boldsymbol{D}} = \langle \ddot{\boldsymbol{x}}\boldsymbol{x}^T \rangle$.

Multiplying eq. (6) by $\boldsymbol{x}^T(t)$ and computing the corresponding correlations, we can derive an additional matrix algebraic equation

$$\hat{\boldsymbol{H}}_2 = -\hat{\boldsymbol{R}}\hat{\boldsymbol{H}}_1. \tag{10}$$

Here we use the white-noise property $\langle \boldsymbol{\Gamma}(t)\boldsymbol{x}^T(t) \rangle = L^{-1}\sum_{k=1}^{L} \boldsymbol{\Gamma}(t'_k)\boldsymbol{x}^T(t_k) = 0$, where $\Gamma(t'_k)$ is taken in the time interval $t_k < t'_k < t_{k+1}$ in the computation $\dot{\boldsymbol{\eta}}(t_k)$ in eqs. (6) and (8). Since $\boldsymbol{\Gamma}(t)$ is a white-noise vector, $\boldsymbol{x}^T(t_k)$ is uncorrelated with $\boldsymbol{\Gamma}(t'_k)$.

Now there are three equations, (2), (9), (10), for four unknown matrices: $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{H}}_1, \hat{\boldsymbol{H}}_2$ and $\hat{\boldsymbol{R}}$. One has then to extract from the variable data $\boldsymbol{x}(t)$ one more relationship between these unknown matrices. We come to a systematical matrix computation to solve this problem. We further multiply eqs. (1), (7), (6) by $\boldsymbol{\eta}^T(t)$, compute the corresponding correlations, and arrive at the following three novel matrix equations

$$\hat{\boldsymbol{G}}_2 = \hat{\boldsymbol{A}}\hat{\boldsymbol{G}}_1 + \hat{\boldsymbol{W}}_1, \tag{11}$$

$$\hat{\boldsymbol{G}}_3 = \hat{\boldsymbol{A}}\hat{\boldsymbol{G}}_2 + \hat{\boldsymbol{W}}_2, \tag{12}$$

$$\hat{\boldsymbol{W}}_2 = -\hat{\boldsymbol{R}}\hat{\boldsymbol{W}}_1 \tag{13}$$

with $\hat{\boldsymbol{G}}_1 = \langle \boldsymbol{x}\boldsymbol{\eta}^T \rangle, \hat{\boldsymbol{G}}_2 = \langle \dot{\boldsymbol{x}}\boldsymbol{\eta}^T \rangle, \hat{\boldsymbol{G}}_3 = \langle \ddot{\boldsymbol{x}}\boldsymbol{\eta}^T \rangle, \hat{\boldsymbol{W}}_1 = \langle \boldsymbol{\eta}\boldsymbol{\eta}^T \rangle, \hat{\boldsymbol{W}}_2 = \langle \dot{\boldsymbol{\eta}}\boldsymbol{\eta}^T \rangle$. In these three new equations there appear five additional unknown matrices, $\hat{\boldsymbol{G}}_1, \hat{\boldsymbol{G}}_2, \hat{\boldsymbol{G}}_3, \hat{\boldsymbol{W}}_1$ and $\hat{\boldsymbol{W}}_2$, and the inverse problem seems to become even farther from being solvable. It is very interesting and elegant that through correlation computations the three matrices $\hat{\boldsymbol{G}}_1$, $\hat{\boldsymbol{G}}_2$, $\hat{\boldsymbol{G}}_3$ can be solved in terms of known variable data and other known and unknown matrices as

$$\begin{aligned} \hat{\boldsymbol{G}}_1 &= \hat{\boldsymbol{H}}_1^T, \\ \hat{\boldsymbol{G}}_2 &= \langle \dot{\boldsymbol{x}}\boldsymbol{\eta}^T \rangle = \langle \dot{\boldsymbol{x}}\dot{\boldsymbol{x}}^T \rangle - \langle \dot{\boldsymbol{x}}\boldsymbol{x}^T \rangle \hat{\boldsymbol{A}}^T \\ &= -\hat{\boldsymbol{D}} - \hat{\boldsymbol{B}}\hat{\boldsymbol{A}}^T, \\ \hat{\boldsymbol{G}}_3 &= \langle \ddot{\boldsymbol{x}}\boldsymbol{\eta}^T \rangle = \langle \ddot{\boldsymbol{x}}\dot{\boldsymbol{x}}^T \rangle - \langle \ddot{\boldsymbol{x}}\boldsymbol{x}^T \rangle \hat{\boldsymbol{A}}^T \\ &= \hat{\boldsymbol{E}} - \hat{\boldsymbol{D}}\hat{\boldsymbol{A}}^T, \end{aligned} \tag{14}$$

where $\hat{\boldsymbol{E}} = \langle \ddot{\boldsymbol{x}}\dot{\boldsymbol{x}}^T \rangle$. In deriving eqs. (14) we have used an identity for continuous variables

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle \boldsymbol{y}\boldsymbol{z}^T \rangle = \langle \dot{\boldsymbol{y}}\boldsymbol{z}^T \rangle + \langle \boldsymbol{y}\dot{\boldsymbol{z}}^T \rangle = 0 \tag{15a}$$

and

$$\langle \dot{\boldsymbol{y}}\boldsymbol{z}^T \rangle = -\langle \boldsymbol{y}\dot{\boldsymbol{z}}^T \rangle \tag{15b}$$

With all the three identities in eq. (14), we are reaching a closed form of matrix algebra, six equations for six unknown matrices. Solving $\hat{\boldsymbol{H}}_1$ from eq. (2), $\hat{\boldsymbol{H}}_2$ from eq. (9) and submitting them to eq. (10) we obtain eq. (16a). Solving $\hat{\boldsymbol{W}}_1$, $\hat{\boldsymbol{W}}_2$ in eqs. (11), (12) with $\hat{\boldsymbol{G}}_1, \hat{\boldsymbol{G}}_2, \hat{\boldsymbol{G}}_3$ given in eq. (14), and submitting them to eq. (13) we derive eq. (16b):

$$\hat{\boldsymbol{D}} - \hat{\boldsymbol{A}}\hat{\boldsymbol{B}} = -\hat{\boldsymbol{R}}(\hat{\boldsymbol{B}} - \hat{\boldsymbol{A}}\hat{\boldsymbol{C}}), \tag{16a}$$

$$\begin{aligned} &\hat{\boldsymbol{E}} - \hat{\boldsymbol{D}}\hat{\boldsymbol{A}}^T + \hat{\boldsymbol{A}}\hat{\boldsymbol{D}} + \hat{\boldsymbol{A}}\hat{\boldsymbol{B}}\hat{\boldsymbol{A}}^T = \\ &-\hat{\boldsymbol{R}}(-\hat{\boldsymbol{D}} - \hat{\boldsymbol{B}}\hat{\boldsymbol{A}}^T - \hat{\boldsymbol{A}}\hat{\boldsymbol{B}}^T + \hat{\boldsymbol{A}}\hat{\boldsymbol{C}}\hat{\boldsymbol{A}}^T), \end{aligned} \tag{16b}$$

$$\hat{\boldsymbol{Q}} = -\hat{\boldsymbol{W}}_2^s = -(\hat{\boldsymbol{W}}_2 + \hat{\boldsymbol{W}}_2^T)/2. \tag{16c}$$

Now in eqs. (16a), (16b) we have two matrix equations with two unknown matrices: $\hat{\boldsymbol{A}}$ for the interaction topology and $\hat{\boldsymbol{R}}$ for the noise correlation. Therefore, in principle the unknown matrices $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ can be solved from eqs. (16) and the inverse problem has a closed form for the solution. Besides, we can also depict the unknown matrix $\hat{\boldsymbol{Q}}$ in eq. (16c) (see eq. (15) in [19]) for white-noise driven networks, where $\hat{\boldsymbol{W}}_2^s$ is the symmetric part of $\hat{\boldsymbol{W}}_2$, which can be obtained with $\hat{\boldsymbol{B}}$ and other matrices directly computable from $\boldsymbol{x}(t)$ data and derived $\hat{\boldsymbol{A}}$ above.

Several crucial points in the above derivation are worth remarking. First, in eq. (2) there is a single equation with two matrices $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$ known from the output variable data $\boldsymbol{x}(t)$ and other two unknown matrices $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{H}}_1$ to be computed. The key idea in our algorithm is that the output data $\boldsymbol{x}$ contains information not only for the topology matrix $\hat{\boldsymbol{A}}$ but also for the noise statistic $\hat{\boldsymbol{H}}_1$. The only problem is how to extract more information from the available data to compute the $\hat{\boldsymbol{H}}_1$ matrix. This goal can be achieved by computing more correlation matrices (*e.g.*, matrix $\hat{\boldsymbol{D}}$). Second, we know almost nothing about noises: nothing about how many different types of colored noises; nothing about the intensities and correlation time distributions of all these noises; nothing about how these different noises are applied on different nodes of the network. And all the knowledge is hidden in the variable data. All these pieces of information included in matrix $\hat{\boldsymbol{R}}$ are fully explored through the matrix algebraic computations from eq. (2) to eqs. (16).

With the closed matrix formalism, eqs. (16), in our hand, the final task is to solve the matrices $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ with available $\boldsymbol{x}(t)$ data, precisely, with the known matrices $\hat{\boldsymbol{B}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{E}}$ computed from $\boldsymbol{x}(t)$ data. This is, however, again nontrivial. The coupled algebraic equations (16) contain $3N^2$ (including N^2 for the $\hat{\boldsymbol{Q}}$ matrix) unknown variables of matrix elements, and the equations are nonlinear. As N is large, it turns to be impossible to solve the matrix algebraic equations analytically, and also it is extremely difficult to numerically search for their solutions. Here we develop a matrix-iteration method to

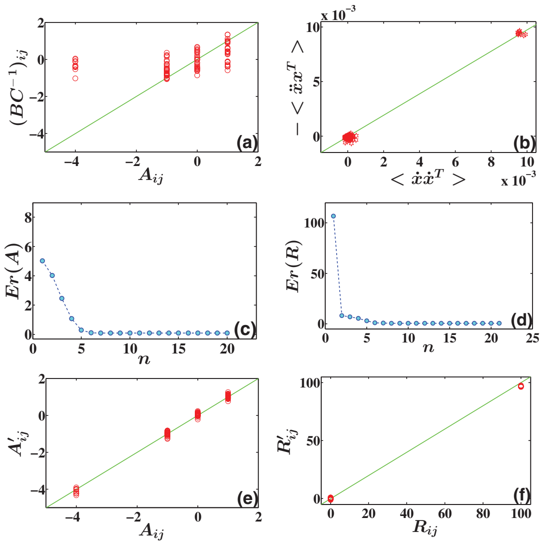

Fig. 1: (Color online) Numerical verifications of the matrix algebra theory and iteration approach. Matrix $\hat{\boldsymbol{A}}$ is given as follows: positive interactions $A(p)_{ij}=1$, negative interactions $A(n)_{ij}=-1$ (both have 33% probability for each crossing interaction), diagonal entries $A_{ii}=-4$. $N=10$, $d=1.0$ and $r=100$ for colored noises. The time step for the simulation is $\delta t=1\times10^{-4}$ and the step for velocity and acceleration computations is $\Delta t=2\times10^{-4}$. $L=5\times10^{6}$. (a) $(\boldsymbol{BC}^{-1})_{ij}$ (noise decorrelation $\hat{\boldsymbol{H}}_1=0$ in eq. (2) used in [19]) *vs.* actual $\hat{\boldsymbol{A}}_{ij}$. For colored noise, no noise decorrelation of $\hat{\boldsymbol{H}}_1=0$ in eq. (4) can be applied. (b) $\langle -\ddot{\boldsymbol{x}}\boldsymbol{x}^T\rangle$ plotted *vs.* $\langle \dot{\boldsymbol{x}}\dot{\boldsymbol{x}}^T\rangle$. The identity of eq. (15) is well justified. (c), (d) Error $Er(\hat{\boldsymbol{A}})$ and $Er(\hat{\boldsymbol{R}})$ of eq. (17) computed for different iteration numbers. $Er(\hat{\boldsymbol{A}})$ and $Er(\hat{\boldsymbol{R}})$ reduce quickly as n increases, and saturate to a small value, showing the effectiveness of the iteration approach. (e), (f) A'_{ij} and R'_{ij} computed by eqs. (16a), (16b) at the 20th iteration plotted *vs.* actual A_{ij} and R_{ij}. Both computed sets and actual network sets coincide with each other excellently.

overcome this difficulty. The procedures are: First, randomly and arbitrarily fixing elements of matrix $\hat{\boldsymbol{R}}(R_{ij})$; second, solving matrix $\hat{\boldsymbol{A}}$ with given $\hat{\boldsymbol{R}}$ in eq. (16a); third, solving matrix $\hat{\boldsymbol{R}}$ in eq. (16b) with the obtained $\hat{\boldsymbol{A}}$; fourth, continue these iterations with renewed $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ until $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ approach to saturated values. The iteration approach is described in the appendix. In the following we will see that these iterations work extremely well and effective, since the approach can correctly depict $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ with a few steps of iterations.

Now let us apply all the above analysis to network inferences. In fig. 1 we study the inverse problem of a small $N=10$ network driven by colored noises with a certain finite correlation time $1/r_i=\tau=0.01$. First, we find in fig. 1(a) that the noise-variable decorrelation approach [19] does no longer work, since the formula $\hat{\boldsymbol{B}}\hat{\boldsymbol{C}}^{-1}$ cannot correctly depict the network structure $\hat{\boldsymbol{A}}$. We then turn to the full algorithm of the matrix algebra eqs. (16). In fig. 1(b) we show that the identity eq. (15) works very well, which validates the matrix equations (16). In figs. 1(c), (d), we calculate the average error between the depicted network structure $\hat{\boldsymbol{A}}'$ (noise correlation matrix $\hat{\boldsymbol{R}}'$) and the actual structure $\hat{\boldsymbol{A}}$ ($\hat{\boldsymbol{R}}$) at different rounds of iterations:

$$
\begin{aligned}
Er(\hat{\boldsymbol{A}}) &= N^{-2}\sum_{i=1}^{N}\sum_{j=1}^{N}|A'_{ij}-A_{ij}|,\\
Er(\boldsymbol{R}) &= N^{-2}\sum_{i=1}^{N}\sum_{j=1}^{N}|R'_{ij}-R_{ij}|.
\end{aligned}
\tag{17}
$$

At the first round we start from a randomly distributed R'_{ij} corresponding to very large errors. $Er(\hat{\boldsymbol{A}})$ and $Er(\hat{\boldsymbol{R}})$ go down very quickly as n increases, and saturate to very small values as $n\geq 6$. In figs. 1(e), (f) we plot the depicted A'_{ij} and R'_{ij} *vs.* the actual A_{ij} and R_{ij}, respectively. All data locate around the diagonal line, indicating excellent inference performance. $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$ are perfectly inferred qualitatively in nature (*i.e.*, positive, negative or null), and satisfactorily inferred quantitatively. Increasing the data quality (*i.e.*, smaller Δt and larger L), the saturated $Er(\hat{\boldsymbol{A}},\hat{\boldsymbol{R}})$ can be further reduced.

For examining the general validity of all the above analytic matrix algebra and numerical iteration approach we infer different networks for different network sizes and different noise distributions in fig. 2. All the results convincingly justify the analytical formulas eqs. (16) and support the efficiency of the iteration technique. In particular, in figs. 2(e), (f), (g), we study a $N=100$ rather large network. The unknown matrix elements are up to 10^4 with complicated diversity, and 100 types of colored noises with different correlation times τ's and different amplitudes d's are applied to different network nodes. All these pieces of information are unknown. By applying compact matrix algebraic equations (16) together with the simple and explicit iteration method, we can achieve the inference goal in 20 iterations for all $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{Q}}$ (*i.e.*, d_i's).

Many facts can influence the quality of depiction. For studying the stability of our method we compute the errors given by eq. (17) against some typical facts. In fig. 3(a) $Er(\hat{\boldsymbol{A}})$ is plotted against the measurement frequency $f=1/\Delta t$. High (low) f represents the high (low) quality of data measurement. $Er(\hat{\boldsymbol{A}})$ decreases monotonously and stably as f increases, and saturates to a small value determined by other facts. We further consider the influence of the measurement noise $\boldsymbol{x}'(t)=\boldsymbol{x}(t)+\zeta(t)$, where $\boldsymbol{x}$ are the exact observables produced by dynamic networks, $\boldsymbol{x}'$ the actually measured values and $\zeta(t)$ (taken in the interval $2\delta std(x_i)\geq\zeta_i(t)\geq-2\delta sta(x_i)$ where $std(\bullet)$ denotes the standard variance and $\delta>0$) is the inevitable measurement error. In fig. 3(b) $Er(\hat{\boldsymbol{A}})$ is plotted *vs.* δ, which decreases stably while δ decreases. By using a certain variable average approach, the iteration approach can work well with even larger measurement noise. In fig. 3(c) $Er(\hat{\boldsymbol{A}})$ is plotted *vs.* the sample size L, which first decreases according to the rule of $1/\sqrt{L}$, and then saturates to a small finite value for sufficiently large L. All the behaviours show the stability, effectiveness and

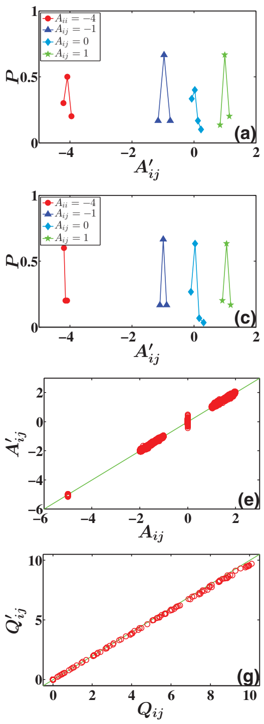

Fig. 2: (Color online) Comparisons of numerically computed $\hat{\boldsymbol{A}}'$ and $\hat{\boldsymbol{R}}'$ by eqs. (16) with the actual $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{R}}$, respectively, in various more complicated dynamic networks. (a), (b) Probability distributions of A'_{ij}, R'_{ij} computed by eqs. (16). A'_{ij} and R'_{ij} locate with large probabilities around their actual A_{ij} and R_{ij} values. N, τ and $\hat{\boldsymbol{A}}$ are the same as in fig. 1. d_i in eq. (5) is randomly given in the range of (0.1,10). (c), (d) N, d_i and $\hat{\boldsymbol{A}}$ are the same as in fig. 1. r_i is randomly chosen in (10,110) with uniform distribution. (c) The same as (a) with different parameter set. (d) The same as fig. 1(f) with different parameter set. (e), (f), (g) Network size $N = 100$. d_i, r_i and nonzero A_{ij} $(i \neq j)$ are chosen randomly in the following ranges: $d_i \in (0.1, 10.1)$, $r_i \in (50, 150)$, positive interactions $A(p)_{ij} \in (1.0, 2.0)$ and negative interactions $A(n)_{ij} \in (-2.0, -1.0)$ with uniform distributions. (e), (f) The same as figs. 1(e), (f), respectively, with different parameter set. The matrix algebra eqs. (16) and the iteration method are fully confirmed by different d_i, r_i, and A_{ij} distributions and different system size. (g) Q'_{ij} computed by eq. (16c) plotted against the actual Q_{ij} values, the agreement between Q'_{ij} and Q_{ij} satisfactorily confirms the analysis from eq. (2) to eqs. (16).

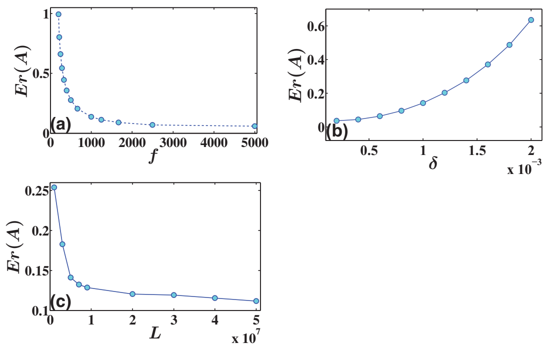

Fig. 3: (Color online) Error $Er(\hat{\boldsymbol{A}})$ plotted against various measurement quantities. All parameters are given in fig. 1 unless specified otherwise. (a) $Er(\hat{\boldsymbol{A}})$ of eq. (17) plotted *vs.* the measurement frequency $f = 1/\Delta t$ with Δt given after eq. (1). (b) $Er(\hat{\boldsymbol{A}})$ plotted *vs.* δ, the measurement noise intensity. $Er(\hat{\boldsymbol{A}})$ decreases as δ decreases. (c) $Er(\hat{\boldsymbol{A}})$ plotted *vs.* the sample size L. $Er(\hat{\boldsymbol{A}})$ first decreases, following the scaling rule of $1/\sqrt{L}$, and then saturates to a small finite value. In all (a), (b), (c) cases, we can observe stability, robustness and effectiveness of the inference computations with controllable errors.

robustness of the method of algebraic computation of matrices.

In conclusion we have studied the inference problem of dynamic networks driven by noises with finite correlation times. In the reverse problem we are aiming at depicting network structures with known output node variables. However, apart from network structures, some additional fact, in particular the noise fact, can influence the network dynamics. A central idea in this work is that the output variable data alone contain not only the information about network structures, but also about all influences from noise and other unknown forces. It is thus possible to accurately depict the network structures with unknown driving forces from the knowledge of output variable data only. With this idea, we develop an analytical matrix algorithm of network inference to extract more information by computing multiple matrices; conducting systematical matrix algebraic computation, which lead to a set of explicit, accurate and close matrix equation formulas. Moreover, a matrix iteration approach is proposed for computationally solving the equations and depicting both network topology and noise correlations. Numerical simulations convincingly verify all the theoretical ideas and derivations.

It is however emphasized that some important facts affecting the inverse computations are not considered here among which the effects of time delay in network dynamics is a very important one. Moreover, for applying the formalisms to analyze practical data, still many preparations should be done. First one should understand some properties of unknown driving noises and signals (*i.e.*, fast-varying driving or slow-varying impacts, and so on), this understanding is crucial for designing corresponding methods. Second, one has to find an effective method to judge the correctness of the obtained results. In practical inverse computations no comparisons like figs. 1(e), (f) and figs. 2(e), (f), (g) can be made, since we do not know the actual objects $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{R}}$, and $\hat{\boldsymbol{Q}}$. Some self-consistent checking algorithms should be developed for properly solving the practical inverse problems. All these facts will be further studied in our future works.

* * *

This work was supported by the National Natural Science Foundation of China (Grant No. 11135001, No. 11174034, No. 11075016 and No. 11475022).

Appendix

Before discussing the procedure of the matrix-iteration method, we firstly give an introduction to the Kronecker product and vec operator. Given $\hat{\boldsymbol{P}} = (p_{ij}) \in \mathbb{R}^{m_1 \times n_1}$ and $\hat{\boldsymbol{S}} = (s_{ij}) \in \mathbb{R}^{m_2 \times n_2}$, the Kronecker product of $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{S}}$ is denoted by $\hat{\boldsymbol{P}} \otimes \hat{\boldsymbol{S}}$ and is defined to be a $m_1 m_2 \times n_1 n_2$ matrix obtained by replacing each element p_{ij} of $\hat{\boldsymbol{P}}$ with the $m_2 \times n_2$ matrix $p_{ij}\hat{\boldsymbol{S}}$ as follows [22]:

$$\hat{\boldsymbol{P}} \otimes \hat{\boldsymbol{S}} = \begin{pmatrix} p_{11}\hat{\boldsymbol{S}} & p_{12}\hat{\boldsymbol{S}} & \cdots & p_{1n_1}\hat{\boldsymbol{S}} \\ p_{21}\hat{\boldsymbol{S}} & p_{22}\hat{\boldsymbol{S}} & \cdots & p_{2n_1}\hat{\boldsymbol{S}} \\ \vdots & \vdots & & \vdots \\ p_{m_1 1}\hat{\boldsymbol{S}} & p_{m_1 2}\hat{\boldsymbol{S}} & \cdots & p_{m_1 n_1}\hat{\boldsymbol{S}} \end{pmatrix}.$$

The vec operator [22] is denoted by the symbol $vec(\hat{\boldsymbol{P}})$, and is defined as

$$vec(\hat{\boldsymbol{P}}) = (\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \cdots, \boldsymbol{p}_{n_1}^T)^T, \tag{A.1}$$

where $\boldsymbol{p}_i$ for $i = 1, 2, \cdots, n_1$ represents the vector composed by the i-th column of the matrix $\hat{\boldsymbol{P}}$. Vec operator has an interesting property if it acts on the product of matrices. For $\hat{\boldsymbol{P}} = (p_{ij}) \in \mathbb{R}^{m_1 \times n_1}$, $\hat{\boldsymbol{S}} = (s_{ij}) \in \mathbb{R}^{m_2 \times n_2}$ and $\hat{\boldsymbol{Z}} = (z_{ij}) \in \mathbb{R}^{n_1 \times m_2}$, we have [22]

$$vec(\hat{\boldsymbol{P}}\hat{\boldsymbol{Z}}\hat{\boldsymbol{S}}) = (\hat{\boldsymbol{S}}^T \otimes \hat{\boldsymbol{P}}) vec(\hat{\boldsymbol{Z}}), \tag{A.2}$$

moreover, this property is particularly useful for solving matrix equations.

Given $\hat{\boldsymbol{R}}_n$ ($n = 1, 2, \cdots, K$ and K are the total iteration times), we can derive $\hat{\boldsymbol{A}}_{n+1}$ by solving eq. (16a). To solve eq. (16a), we firstly arrange it as

$$\hat{\boldsymbol{A}}_{n+1}\hat{\boldsymbol{B}} + \hat{\boldsymbol{R}}_n\hat{\boldsymbol{A}}_{n+1}\hat{\boldsymbol{C}} = \hat{\boldsymbol{D}} + \hat{\boldsymbol{R}}_n\hat{\boldsymbol{B}}, \tag{A.3}$$

and then we apply $vec(\bullet)$ on both sides of eq. (A.3)

$$(\hat{\boldsymbol{B}}^T \otimes \hat{\boldsymbol{I}}_N) vec(\hat{\boldsymbol{A}}_{n+1}) + (\hat{\boldsymbol{C}}^T \otimes \hat{\boldsymbol{R}}_n) vec(\hat{\boldsymbol{A}}_{n+1}) = vec(\hat{\boldsymbol{D}} + \hat{\boldsymbol{R}}_n\hat{\boldsymbol{B}}),$$

where $\hat{\boldsymbol{I}}_N$ is the unit matrix in the $N \times N$ matrix space. Solve $vec(\hat{\boldsymbol{A}}_{n+1})$ as eq. (A.4a), reshape vector $vec(\hat{\boldsymbol{A}}_{n+1})$ into $N \times N$ matrix $\hat{\boldsymbol{A}}_{n+1}$ and take it into eq. (16b). It is easy to get $\hat{\boldsymbol{R}}_{n+1}$ as eq. (A.4b).

$$vec(\hat{\boldsymbol{A}}_{n+1}) = (\hat{\boldsymbol{B}}^T \otimes \hat{\boldsymbol{I}}_N + \hat{\boldsymbol{C}}^T \otimes \hat{\boldsymbol{R}}_n)^{-1} \bullet vec(\hat{\boldsymbol{D}} + \hat{\boldsymbol{R}}_n\hat{\boldsymbol{B}}), \tag{A.4a}$$

$$\hat{\boldsymbol{R}}_{n+1} = -(\hat{\boldsymbol{E}} - \hat{\boldsymbol{D}}\hat{\boldsymbol{A}}_{n+1}^T + \hat{\boldsymbol{A}}_{n+1}\hat{\boldsymbol{D}} + \hat{\boldsymbol{A}}_{n+1}\hat{\boldsymbol{B}}\hat{\boldsymbol{A}}_{n+1}^T) \bullet(-\hat{\boldsymbol{D}} - \hat{\boldsymbol{B}}\hat{\boldsymbol{A}}_{n+1}^T - \hat{\boldsymbol{A}}_{n+1}\boldsymbol{B^T} + \hat{\boldsymbol{A}}_{n+1}\hat{\boldsymbol{C}}\hat{\boldsymbol{A}}_{n+1}^T)^{-1}. \tag{A.4b}$$

REFERENCES

[1] Bullmore E. and Sporns O., *Nat. Rev. Neurosci.*, **10** (2009) 186.

[2] Schneidman E., Berry M. J., Segev R. and Bialek W., *Nature*, **440** (2006) 1007.

[3] Butte A. J., Tamayo P., Slonim D. *et al.*, *Proc. Natl. Acad. Sci. U.S.A.*, **97** (2000) 12182.

[4] C. Elegans Sequencing Consortium., *Science*, **282** (1998) 2012.

[5] Barabasi A. L. and Oltvai Z. N., *Nat. Rev. Genet.*, **5** (2004) 101.

[6] Gale D. M. and Kariv S., *Am. Econ. Rev.*, **97** (2007) 99.

[7] Eagle N., Pentland A. S. and Lazer D., *Proc. Natl. Acad. Sci. U.S.A.*, **106** (2009) 15274.

[8] Marbach D., Costello J. C., Küffner R. *et al.*, *Nat. Methods*, **9** (2012) 796.

[9] Barzel B. and Barabasi A.-L., *Nat. Biotechnol.*, **31** (2013) 720.

[10] Basso K., Margolin A. A., Stolovitzky G. *et al.*, *Nat. Genet.*, **37** (2005) 382.

[11] Timme M. and Casadiego J., *J. Phys. A: Math. Theor.*, **47** (2014) 343001.

[12] Yu D., Righero M. and Kocarev L., *Phys. Rev. Lett.*, **97** (2006) 188701.

[13] Shandilya S. G. and Timme M., *New J. Phys.*, **13** (2011) 013004.

[14] Ren J., Wang W.-X., Li B. and Lai Y.-C., *Phys. Rev. Lett.*, **104** (2010) 058701.

[15] Ching E. S. C., Lai P. Y. and Leung C. Y., *Phys. Rev. E*, **91** (2015) 030801.

[16] Jansen R., Yu H., Greenbaum D. *et al.*, *Science*, **302** (2003) 449.

[17] Haury A. C., Mordelet F., Vera-Licona P. *et al.*, *BMC Syst. Biol.*, **6** (2012) 145.

[18] Faith J. J., Hayete B., Thaden J. T. *et al.*, *PLoS Biol.*, **5** (2007) e8.

[19] Zhang Z., Zheng Z., Niu H., Mi Y., Wu S. and Hu G., *Phys. Rev. E*, **91** (2015) 012814.

[20] Levnaji Z. and Pikovsky A., *Sci. Rep.*, **4** (2014) 5030.

[21] Risken H., *Fokker-Planck Equation* (Springer, Berlin Heidelberg) 1984.

[22] Harville D. A., *Matrix Algebra from a Statistician's Perspective* (Springer, New York) 1997.