Sparse dynamical Boltzmann machine for reconstructing complex networks with binary dynamics

Yu-Zhong Chen[1] and Ying-Cheng Lai[1,2,*]
[1]*School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2]*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Received 7 April 2017; revised manuscript received 9 August 2017; published 28 March 2018)

Revealing the structure and dynamics of complex networked systems from observed data is a problem of current interest. Is it possible to develop a completely data-driven framework to decipher the network structure and different types of dynamical processes on complex networks? We develop a model named sparse dynamical Boltzmann machine (SDBM) as a structural estimator for complex networks that host binary dynamical processes. The SDBM attains its topology according to that of the original system and is capable of simulating the original binary dynamical process. We develop a fully automated method based on compressive sensing and a clustering algorithm to construct the SDBM. We demonstrate, for a variety of representative dynamical processes on model and real world complex networks, that the equivalent SDBM can recover the network structure of the original system and simulates its dynamical behavior with high precision.

DOI: 10.1103/PhysRevE.97.032317

I. INTRODUCTION

A central issue in complexity science and engineering is systems identification and dynamical behavior prediction based on experimental or observational data. For a complex networked system, often the network structure and the nodal dynamical processes are unknown but only time series measured from various nodes in the network can be obtained. The challenging task is to infer the detailed network topology and the nodal dynamical processes from the available data. This line of pursuit started in biomedical science for problems such as identification of gene regulatory networks from expression data in systems biology [1–4] and uncovering various functional networks in the human brain from activation data in neuroscience [5–8]. The inverse problem has also been an area of research in statistical physics where, for example, the inverse Ising problem in static [9–13] and kinetic [14–20] situations has attracted continuous interest. Recent years have witnessed the emergence and growth of a subfield of research in complex networks: data based network identification (or reverse engineering of complex networks) [21–35]. In these works, the success of mapping out the entire network structure and estimating the nodal dynamical equations partly relies on taking advantage of the particular properties of the system dynamics in terms of the specific types and rules. For example, depending on the detailed dynamical processes such as continuous-time oscillations [26,30–33,36,37], evolutionary games [27], or epidemic spreading [28], appropriate mathematical frameworks uniquely tailored at the specific underlying dynamical process can be formulated to solve the inverse problem.

In this paper, we address the following challenging question: Is it possible to develop a completely data-driven framework for extracting the network topology and identifying the dynamical processes, without the need to know *a priori* the specific types of network dynamics? An answer to this question would be of significant value not only to complexity science and engineering, but also to modern data science where the goal is to unearth the hidden structural information and to predict the future evolution of the system. We introduce a Boltzmann machine for complex networked systems with pairwise interactions and demonstrate that such a "machine" can indeed be developed for a large number of distinct types of binary network dynamical processes. Our approach will be a combination of numerical computation and physical reasoning. Since we are yet able to develop a rigorous mathematical framework, this work should be regarded as an initial attempt towards the development of a general framework for network reconstruction and dynamics prediction.

The key principle underlying our work is the following. In spite of the difference among the types of binary dynamics in terms of, e.g., the interaction patterns and state updating rules, there are two common features shared by many dynamical processes on complex networks: (1) they are stochastic, first-order Markovian processes, i.e., only the current state of the system determines its immediate future; and (2) the nodal interactions are local, i.e., a node typically interacts with its neighboring nodes, not all the nodes in the network. The two features are characteristic of a *Markov network* (or a *Markov random field*) [38,39]. In particular, a Markov network is an undirected and weighted probabilistic graphical model that is effective for determining the complex probabilistic interdependencies in situations where the directionality in the interaction between the connected nodes cannot be naturally assigned, in contrast to the directed Bayesian networks [38,39]. In this work, however, we demonstrate that we can make a proper modification to the undirected Markov network to accommodate networked systems with a directed topology. Note that a Markov network has two types of parameters: a nodal bias parameter that controls its preference of the state choice and a weight parameter characterizing the interaction

[*] ying-cheng.lai@asu.edu

strength of each undirected link. In our work, the weight parameter of an incoming (or outgoing) link is defined to characterize the interaction from (or to) a neighboring node in a directed network.

For a network of N nodes with x_j being the state of node j ($j = 1,\ldots,N$), the joint probability distribution of the state variables $\mathbf{X} = (x_1,x_2,\ldots,x_N)^{\mathrm{T}}$ is given by $P(\mathbf{X}) = \prod_C \phi(\mathbf{X}_C)/\sum_{\mathbf{X}}\prod_C \phi(\mathbf{X}_C)$, where $\phi(\mathbf{X}_C)$ is the potential function for a well-connected network clique C, and the summation in the denominator is over all possible system state $\mathbf{X}$. If this joint probability distribution is available, all conditional probability interdependencies can be obtained. The way to define a clique and to determine its potential function plays a key role in the Markov network's representation power of modeling the interdependencies within a particular system.

To be concrete, in this work we pursue the possibility of modeling the conditional probability interdependencies of a variety of binary dynamical processes on complex networked systems via a properly modified Ising-Markov network with its potential function having the form of the Boltzmann factor, $\exp(-E)$, where E is the energy determined by the local states and their interactions along with the network parameters (the incoming link weights and node biases) in a *log-linear* form [40]. This is effectively a sparse Boltzmann machine [40] that allows directed connections adopted to complex network topologies without hidden units. (Note that hidden units usually play a crucial role in ordinary Boltzmann machines [40].) We introduce a temporal evolution mechanism as a persistent sampling process for such a machine based on the conditional probabilities obtained via the joint probability, and generate a Markov chain of persistently sampled state configurations to form the state transition time series for each node. We call our model a *sparse dynamical Boltzmann machine* (SDBM).

For a binary dynamical process on complex networks, such as epidemic spreading or evolutionary game dynamics, the state of each node at the next time step is determined by the probability conditioned on the current states of its neighbors (and its own state in some cases). There is freedom to manipulate the conditional probabilities that dictate the system behavior in the immediate future through the adjustment of its parameter values, i.e., the weights and biases. A basic question is then, for an SDBM, is it possible to properly choose these parameters so that the conditional probabilities so produced are approximately identical to those of the network dynamical process with each given observed system state configuration? If the answer is affirmative, the SDBM can serve as a dynamics approximator of the original system, and the approximated conditional probabilities possess predictive power for the system state at the next time step. When such an SDBM is found for many types of dynamical process on complex networks of various directed and undirected topologies, it effectively serves as a dynamics approximator.

When an approximator has been found for the dynamics on a complex network, the topology of the SDBM is an approximate representation of the original network, providing a solution to the problem of network structure reconstruction. Previous works on the inverse static or kinetic Ising problems led to methods of reconstruction for Ising dynamics by maximizing the data likelihood (or pseudolikelihood) function via various gradient descent approaches [9–20]. Instead of adopting these methods, as a part of our methodology to extract the network structure, we articulate a compressive sensing [41–46] based approach, whose working power has been demonstrated for a variety of non-Ising type of dynamics on complex networks [26–29,47–49]. By incorporating the K-means clustering algorithm into the sparse solution obtained from compressive sensing, we demonstrate that nearly perfect reconstruction of the complex network topology can be achieved. Using 14 different types of dynamical processes on complex networks, we find that, if the time series data generated by these dynamical processes are assumed to be from its equivalent SDBMs, the reconstruction framework is capable of recovering the underlying network structure for each type of original dynamics with essentially zero error. This represents solid and concrete evidence that SDBM is capable of serving as a structural estimator for complex networks with directed and undirected interactions. In addition to being able to precisely reconstruct the network topology, the SDBM also allows the link weights and nodal biases to be calculated with high accuracy. An appealing feature of our method is that it is fully automated and does not require any subjective parameter choice.

Section II provides a general formulation of SDBM as a structural estimator with a focus on undirected networks. The use of compressive sensing and the implementation of K-means algorithm are described. A parameter estimation scheme and a degree guided solution strategy are introduced. The issue of estimating link weights and nodal bias is addressed. Section III presents reconstruction results for a variety of model and real networks coupled with 14 different types of dynamical processes. Section IV contains a general discussion of the SDBM method. A number of side issues together with certain details of the real networks are placed in the Appendices.

II. METHOD FORMULATION

A. SDBM as a structural estimator for undirected complex networks

1. Sparse dynamical Boltzmann machine and compressive sensing

An SDBM with symmetric link weights is effectively a classical Markov network. For an SDBM of size N, the probability that the system is in a particular binary state configuration $\mathbf{X}_{N\times 1} = (x_1,x_2,\ldots,x_N)^{\mathrm{T}}$ is given by

$$P(\mathbf{X}) = \frac{\exp(-E_{\mathbf{X}})}{\sum_{\mathbf{X}}\exp(-E_{\mathbf{X}})}, \tag{1}$$

where $E_{\mathbf{X}}$ is the total energy of the network in $\mathbf{X}$:

$$E_{\mathbf{X}} = \mathbf{X}^{\mathrm{T}}\cdot\mathbf{W}\cdot\mathbf{X} = \sum_{i\neq j} w_{ij}x_i x_j + \sum_{i=1}^{N} b_i x_i, \tag{2}$$

x_i and x_j are binary variables (0 or 1) characterizing the states of nodes i and j, respectively, and $\mathbf{W}$ is a weighted matrix with its off diagonal elements $w_{ij} = w_{ji}$ ($i,j = 1,\ldots,N$, $i \neq j$) specifying the weight associated with the link between nodes i and j. The ith diagonal element of $\mathbf{W}$ is the bias parameter b_i for node i ($i = 1,\ldots,N$), which determines node i's preference to state 0 or 1. The total energy $E_{\mathbf{X}}$ includes the interaction energies (the sum of all $w_{ij}x_i x_j$ terms) and the nodes' self-energies (the various $b_i x_i$ terms). The partition function of the system is given by

$$\mathcal{Z} = \sum_{\mathbf{X}} \exp(-E_{\mathbf{X}}), \tag{3}$$

where the summation is over all possible $\mathbf{X}$ configurations. The state of node i at the next time step is determined by the states of all other nodes at the present time step $\mathbf{X}_i^{\mathrm{R}}$ through the following conditional probability:

$$P\{x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)\} = \frac{P\{x_i(t+1)=1,\mathbf{X}_i^{\mathrm{R}}(t)\}}{P\{x_i(t+1)=1,\mathbf{X}_i^{\mathrm{R}}(t)\} + P\{x_i(t+1)=0,\mathbf{X}_i^{\mathrm{R}}(t)\}}, \tag{4}$$

where the two joint probabilities are given by

$$P\{x_i(t+1)=1,\mathbf{X}_i^{\mathrm{R}}(t)\} = \frac{1}{\mathcal{Z}} \exp\left[-\sum_{j=1,j\neq i}^{N} w_{ij}x_j(t) - b_i - \sum_{s=1,s\neq i}^{N}\sum_{j=1,j\neq i}^{N} w_{sj}x_s(t)x_j(t) - \sum_{s=1,s\neq i}^{N} b_s x_s(t)\right],$$

$$P\{x_i(t+1)=0,\mathbf{X}_i^{\mathrm{R}}(t)\} = \frac{1}{\mathcal{Z}} \exp\left[-\sum_{s=1,s\neq i}^{N}\sum_{j=1,j\neq i}^{N} w_{sj}x_s(t)x_j(t) - \sum_{s=1,s\neq i}^{N} b_s x_s(t)\right].$$

A Markov network defined in this fashion is in fact the kinetic Ising model [14–20]. With the joint probabilities, the conditional probability in Eq. (4) becomes

$$P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\} = \frac{1}{1+\exp\left[\sum_{j=1,j\neq i}^{N} w_{ij}x_j(t) + b_i\right]}. \tag{5}$$

We thus have

$$\ln\left(\frac{1}{P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}} - 1\right) = \sum_{j=1,j\neq i}^{N} w_{ij}x_j(t) + b_i.$$

Letting $Q_i(t) \equiv (P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\})^{-1} - 1$, we have

$$\ln Q_i(t) = (x_1(t),\ \ldots, x_{i-1}(t), x_{i+1}(t),\ \ldots, x_N(t), 1)\begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix}. \tag{6}$$

For M distinct time steps $t_1,t_2,\ldots,t_M$, we obtain the following matrix form:

$$\begin{pmatrix} \ln Q_i(t_1) \\ \ln Q_i(t_2) \\ \vdots \\ \ln Q_i(t_M) \end{pmatrix} = \begin{pmatrix} x_1(t_1), & \ldots, & x_{i-1}(t_1), & x_{i+1}(t_1), & \ldots, & x_N(t_1), & 1 \\ x_1(t_2), & \ldots, & x_{i-1}(t_2), & x_{i+1}(t_2), & \ldots, & x_N(t_2), & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(t_M), & \ldots, & x_{i-1}(t_M), & x_{i+1}(t_M), & \ldots, & x_N(t_M), & 1 \end{pmatrix}\begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix}, \tag{7}$$

which can be written concisely as

$$\mathbf{Y}_{M\times 1} = \ominus_{M\times N} \cdot \mathbf{V}_{N\times 1}, \tag{8}$$

where the vector $\mathbf{Y}_{M\times 1} \in R^M$ contains the values of $\ln Q_i(t)$ for M different time steps, the $M \times N$ matrix $\ominus_{M\times N}$ is determined by the states of all the nodes except i, and the first $(N-1)$ components of the vector $\mathbf{V}_{N\times 1} \in R^N$ are the link weights between node i and all other nodes in the network, as illustrated in Fig. 1, with its last entry being node i's intrinsic bias.

Since the conditional probability $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ depends solely on the state configuration of i's nearest neighbors, or i's *Markov blanket* [38,39] at time t, as shown in Fig. 1(b), identical configurations at other time steps imply identical conditional probabilities. Thus, given time series data of the dynamical process, the conditional probability can be estimated according to the law of large numbers by averaging over the states of i at all the time steps prior to the neighboring state configurations becoming identical. Note, however, that this probability needs to be conditioned on the state configuration of the entire system except node i, i.e., on $\mathbf{X}_i^{\mathrm{R}}(t)$, and the average of x_i is calculated over all the time steps $t_m + 1$ satisfying $\mathbf{X}_i^{\mathrm{R}}(t) = \mathbf{X}_i^{\mathrm{R}}(t_m)$. This means that there can be a dramatic increase in the configuration size, i.e., from k_i (the degree of node i) to $N - 1$ (the size of the vector $\mathbf{X}_i^{\mathrm{R}}$), which can make the number of exactly identical configurations too small to give any meaningful statistics. To overcome this difficulty, we allow a small amount of dissimilarity between $\mathbf{X}_i^{\mathrm{R}}(t)$ and $\mathbf{X}_i^{\mathrm{R}}(t_m)$ by introducing a tolerance parameter Γ $(0 \leqslant \Gamma \leqslant 0.5)$ to confine the corresponding Hamming distances normalized by N. In particular, we assume $\mathbf{X}_i^{\mathrm{R}}(t) \approx \mathbf{X}_i^{\mathrm{R}}(t_m)$ if the relative difference between them is not larger than Γ. This averaging process leads to

$$\begin{pmatrix} \ln q_i(t_1) \\ \ln q_i(t_2) \\ \vdots \\ \ln q_i(t_M) \end{pmatrix} \approx \begin{pmatrix} \langle x_1(t_1)\rangle, & \ldots, & \langle x_{i-1}(t_1)\rangle, & \langle x_{i+1}(t_1)\rangle, & \ldots, & \langle x_N(t_1)\rangle, & 1 \\ \langle x_1(t_2)\rangle, & \ldots, & \langle x_{i-1}(t_2)\rangle, & \langle x_{i+1}(t_2)\rangle, & \ldots, & \langle x_N(t_2)\rangle, & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle x_i(t_M)\rangle, & \ldots, & \langle x_{i-1}(t_M)\rangle, & \langle x_{i+1}(t_M)\rangle, & \ldots, & \langle x_N(t_M)\rangle, & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix}, \tag{9}$$

where $q_i(t) \equiv \langle x_i(t_1 + 1)\rangle^{-1} - 1$, with $\langle \ldots \rangle$ standing for the averaging over all instants of time at which the condition $\mathbf{X}_i^{\mathrm{R}}(t) = \mathbf{X}_i^{\mathrm{R}}(t_m)$ is met (see Appendix B for detailed derivation of this approximation). A schematic illustration of the whole process is presented in Fig. 1, with Eq. (9) shown graphically in Fig. 1(c).

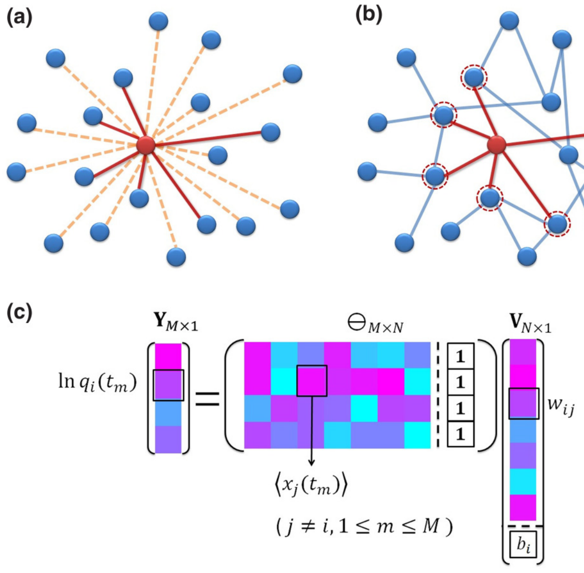

FIG. 1. A schematic illustration of structural reconstruction using SDBM for undirected complex networks. (a) Reconstruction of the local connection structure of the red node in a network of 20 nodes. The connections of the network are assumed to be unknown, so the red node can potentially be connected with any other node, as shown by the orange dashed links. Executing compressive sensing [(c)] and the K-means clustering (Fig. 2) for this node based on time series leads to its true connection structure, marked by the red links. The link weights and the nodal bias are represented by the vector $\mathbf{V}_{N\times1}$ in (c). (b) All true connections of the network are recovered through the process in (a) for each node. The Markov blanket of the red node in (a) consists of all its nearest neighbors, indicated by the nodes with dashed red circles. (c) A schematic illustration of the compressive sensing framework for structural reconstruction as in Eq. (9).

Further explanation of our approximations and their consequences is as follows. We first introduce the dissimilarity tolerance parameter Γ where, in order to obtain a sufficient number of similar system configurations, reasonably large values of Γ are necessary. While this procedure tends to induce dissimilar configurations, a large number of them have little effect on the estimation accuracy. This is because the true conditional probability to be estimated depends only on the configuration of a node's nearest neighbors, not that of the entire network. In terms of the mean field approximation, the degree of any node is always much smaller than the network size. Thus, for node i with degree k_i in a network of size N, the probability that a flipped node is not a nearest neighbor of i is $(N - k_i)/N$. Since there can be at most ΓN flipped nodes, the probability that none of them belong to the nearest neighbors of i is given by $[(N - k_i)/N]^{\Gamma N}$. To be concrete, we set $\Gamma = 35\%$. (Our extensive calculations show, however, that the results are robust for Γ values ranging from 25% to 40%.) Thus, in a representative case, if the number of similar configurations is $L = 7000$, for the typical ratio $k_i/N = 0.05$ the number of "good" configurations is $L[(N - k_i)/N]^{\Gamma N} \approx 1160$, which is generally sufficient for accurate estimation of the probability.

For a complex network, the degree of a typical node is small compared with the network size. For node i, the link weights are nonzero only for the connections with the immediate neighbors. The vector $\mathbf{V}_{N\times1}$ is thus typically sparse with majority of its elements being zero. The sparsity property renders applicable compressive sensing [41–46], through which an N dimensional sparse vector can be reconstructed via a set of M measurements, for $M \ll N$. By minimizing the L_1 norm of $\mathbf{V}_{N\times1}$, i.e., $\|\mathbf{V}_{N\times1}\|_1 = \sum_{j=1,j\neq i}^{N} |w_{ij}| + |b_i|$, subject to the constraint $\mathbf{Y}_{M\times1} = \ominus_{M\times N} \cdot \mathbf{V}_{N\times1}$, we can reconstruct $\mathbf{V}_{N\times1}$ to obtain the connection weights between node i and all other nodes in the network. One tempts to hope that, applying the procedure to each node would lead to the complete weighted adjacency matrix $\mathbf{W}$. However, for compressive sensing to be effective, a suitable clustering algorithm is necessary for distinguishing the existent from nonexistent links.

2. Necessity of *K*-means algorithm and implementation

In previous works on reconstruction of complex networks based on stochastic dynamical correlations [36,37] or compressive sensing [27,28], the existent (real) links can be distinguished from the nonexistent links by setting a single threshold value of certain quantitative measure. The success relies on the fact that the dynamics at various nodes are of the same type, and the reconstruction algorithm is tailored toward the specific type of dynamical process. Our task is significantly more challenging as the goal is to develop a system (or a machine) to replicate a diverse array of dynamical processes based on data for networks with pairwise interactions. For compressive sensing based reconstruction, the computational criteria to distinguish existent from nonexistent links differ substantially for different types of dynamics in terms of quantities such as the solution magnitude, peak value distribution, and background noise intensity. As a result, a more elaborate and sophisticated procedure is required for determining the threshold for each particular case, suggesting that a straightforward application of compressive sensing cannot lead to a general reconstruction algorithm. One may also regard the solutions of the existent links as a kind of *extreme events* [50–52] superimposed on top of the random background, but it is difficult to devise a general criterion to determine if a peak in the distribution of the quantitative measure represents the correct extreme event corresponding to an actual link.

Through extensive testing, we find that a previously developed unsupervised clustering measure, K means [38,39], possesses the desired traits that can be exploited, in combination with compressive sensing, to develop a reconstruction machine with pairwise interactions. K means can serve as a base for a highly effective structural estimator for various types of dynamics on networks of distinct topologies. Depending on the specific combination of the network topology and dynamics, the reconstruction accuracies vary to certain extent but are acceptable. Since the compressive sensing operation is node specific, the solutions obtained separately from different nodes may give conflicting information as to whether there is an actual link between the two nodes, requiring a proper resolution scheme. We develop such a scheme based on node degree consistency. Our reconstruction machine thus contains three main components: compressive sensing, K means, and conflict resolution. We shall demonstrate in Sec. III that the machine can separate the true positive solutions from the noisy background with high success rate, for all combinations of the nodal dynamics and the network topology tested.

Concretely, K means is one of the simplest and most popular clustering algorithms, which has been used widely for unsupervised clustering tasks [38]. It provides each data example an assignment to a cluster within which the data examples are relatively close (or similar) while being distant from the ones in other clusters. The main steps for solving a typical two-dimensional (2D) data clustering problem via K means are schematically illustrated in Fig. 2. (Note that compressive sensing solutions of the link weights w_{ij} are one-dimensional data points, which form two clusters in 1D, corresponding to the existent and nonexistent links, respectively. The use of 2D illustration is for better visualization.) For each vector $\mathbf{V}_{N\times 1}$ obtained from averaging over multiple applications of compressive sensing, picking out the elements corresponding to the existent links from the nonexistent ones with their fluctuating element values is equivalent to a one-dimensional clustering problem with two target clusters, which is one dimension lower than the case shown in Fig. 2. When implementing the K-means algorithm, we choose the initial cluster center positions for the two clusters as the maximum and minimum values of the vector elements with the bias b_i excluded. This is justified because the compressive sensing solution of b_i can have an overwhelmingly high absolute value, which does not provide any structural information but can severely disrupt the clustering process. Due to the sparsity of complex networks, the cluster with a smaller number of components is regarded as containing the solutions of the existent links, and the components of the other one correspond to the nonexistent links.

3. Conflict resolution for undirected complex networks

For network reconstruction, our SDBM method is essentially a "local" method as it identifies the links for each node. Because of the sparsity condition as required by compressive sensing, large errors can arise for the hub nodes. A relatively large error is thus indication that the underlying node is a hub. The errors, however, do not pose a serious difficulty in reconstructing the whole network, as the links associated with a hub can be inferred and reconciled from the links of the other nodes that the hub is connected with. More specifically, since the links are bidirectional, the node on each side provides a weighted solution of the same link. The two solutions may be quite different, giving contradictory indication of the existence of the link and resulting in an asymmetric adjacency matrix. For majority types of dynamical processes on complex networks, compressive sensing almost always gives higher prediction accuracy for lower degree nodes due to their connection sparsity, which holds true especially for nodes with their degrees smaller than the network average. Based on this fact, when encountering contradictory solutions, we determine the link existence by examining the lower degree side if the degree value is equal to or smaller than the network average. For cases where the degrees of both sides are larger than the network average, the false negative rates are often high. This is because compressive sensing tends to give oversimplified results such as inducing a more than necessary level of sparsity to the solution or causing large fluctuations with the nonexistent links so they are mixed up with the existent ones. As a result, contradictory solutions are treated as positive (existent) solutions. For some types of dynamics, the SDBM dynamics and link-update voter dynamics, or dynamics on real world networks, treating all contradictory solutions this way, regardless of degree values, can improve the reconstruction precision, albeit only slightly in some cases.

B. Parameter estimation scheme and degree guided solution substitution operation

1. Parameter estimation

From the reconstructed network structure, any node i's k_i immediate neighbors, m_1, m_2, …, m_{k_i}, and their state

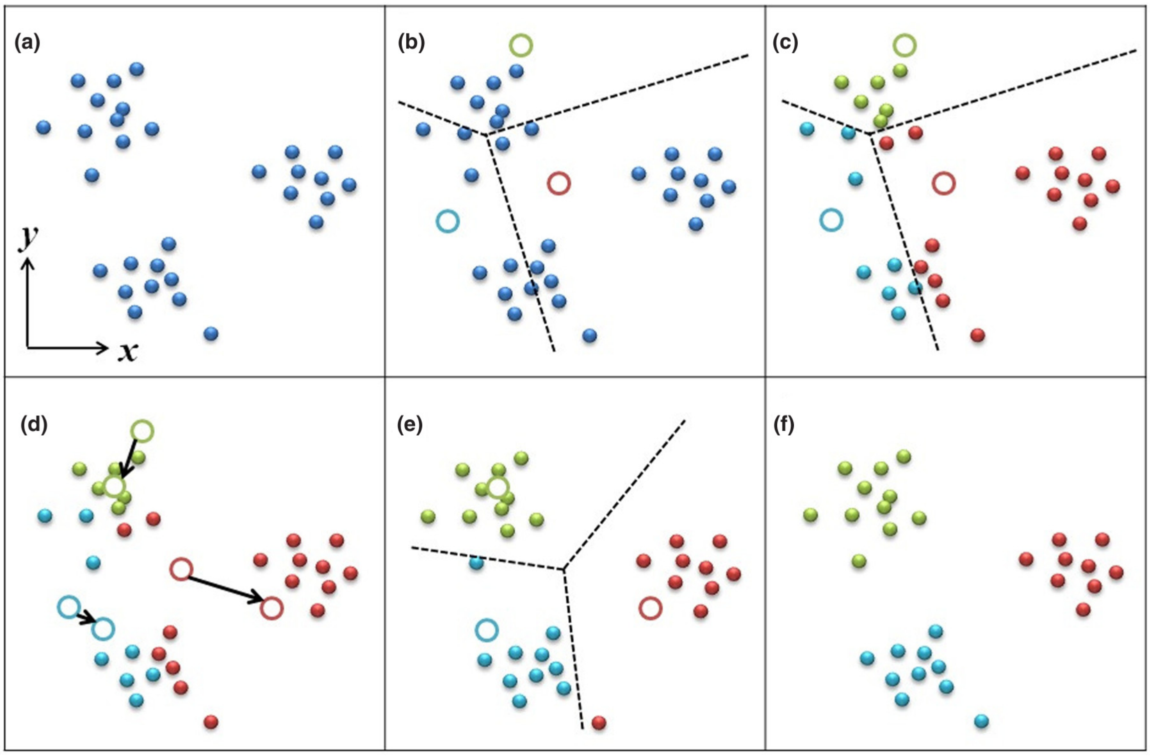

FIG. 2. A schematic illustration of the K-means algorithm for two-dimensional data clustering. (a) The data points (solid blue circles) to be clustered in a 2D feature space. (b) For random locations of the cluster centers (aqua, green, and red hollow circles), each data point can be associated with the closest center. (c) The 2D space is divided into three regions through three decision boundaries (black dashed lines). (d) Each center moves to the centroid of the data points currently assigned to it (movements shown by the black arrows). (e) The updated cluster assignments of the data points are obtained according to the new center locations. The steps in (c) and (d) are repeated until convergence is achieved. (f) The final cluster assignments.

configuration $\mathbf{X}_i^{\mathrm{MB}} = (x_{m_1}, x_{m_2}, \ldots, x_{m_{k_i}})^{\mathrm{T}}$, can be identified. Since the probability that i's state is 1 at the next time step depends only on i's immediate neighbors, $\mathbf{X}_i^{\mathrm{R}}(t)$ in Eq. (5) can be simplified as $\mathbf{X}_i^{\mathrm{MB}}(t)$, and Eq. (5) can be written as

$$P\left\{x_i(t+1)=1\middle|\mathbf{X}_i^{\mathrm{R}}(t)\right\} = P\left\{x_i(t+1)=1\middle|\mathbf{X}_i^{\mathrm{MB}}(t)\right\} = \frac{1}{1+\exp\left[\sum_{m=1}^{k_i} w_{im}x_m(t)+b_i\right]}. \tag{10}$$

Accordingly, Eq. (9) can be simplified as

$$\mathbf{Y}^{\mathrm{MB}}_{(k_i+1)\times 1} = \ominus^{\mathrm{MB}}_{(k_i+1)\times(k_i+1)} \cdot \mathbf{V}^{\mathrm{MB}}_{(k_i+1)\times 1},$$

i.e.,

$$\begin{pmatrix} \ln q_i(t_1) \\ \ln q_i(t_2) \\ \vdots \\ \ln q_i(t_{k_i+1}) \end{pmatrix} = \begin{pmatrix} \langle x_{m_1}(t_1)\rangle, & \langle x_{m_2}(t_1)\rangle, \ldots, & \langle x_{m_{k_i}}(t_1)\rangle, & 1 \\ \langle x_{m_1}(t_2)\rangle, & \langle x_{m_2}(t_2)\rangle, \ldots, & \langle x_{m_{k_i}}(t_2)\rangle, & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \langle x_{m_1}(t_{k_i+1})\rangle, & \langle x_{m_1}(t_{k_i+1})\rangle, \ldots, & \langle x_{m_{k_i}}(t_{k_i+1})\rangle, & 1 \end{pmatrix} \begin{pmatrix} w_{im_1} \\ w_{im_2} \\ \vdots \\ w_{im_{k_i}} \\ b_i \end{pmatrix}, \tag{11}$$

where the k_i+1 linear equations uniquely solve $w_{im_1}, \ldots, w_{im_{k_i}}$ and b_i simply via

$$\mathbf{V}^{\mathrm{MB}}_{(k_i+1)\times 1} = \left[\ominus^{\mathrm{MB}}_{(k_i+1)\times(k_i+1)}\right]^{-1} \cdot \mathbf{Y}^{\mathrm{MB}}_{(k_i+1)\times 1}.$$

Our parameter estimation formulation is illustrated schematically in Fig. 3(a).

For a particular neighboring state configuration of node i, $\mathbf{X}_i^{\mathrm{MB}}(t)$, its occurring frequency determines the precision in the estimation of $P\{x_i(t+1)|\mathbf{X}_i^{\mathrm{MB}}(t)\} \simeq \langle x_i(t+1)\rangle$, which in turn determines the solution precision of Eq. (11). The occurrence of different neighboring state configurations for the same node may differ dramatically. Furthermore, the accuracy in estimating the probability $P\{x_i(t+1)|\mathbf{X}_i^{\mathrm{MB}}(t)\}$ depends on the node degree due to the increasing difficulty in finding exactly the same configurations for larger degree nodes. Overcoming the estimation difficulty is a highly nontrivial

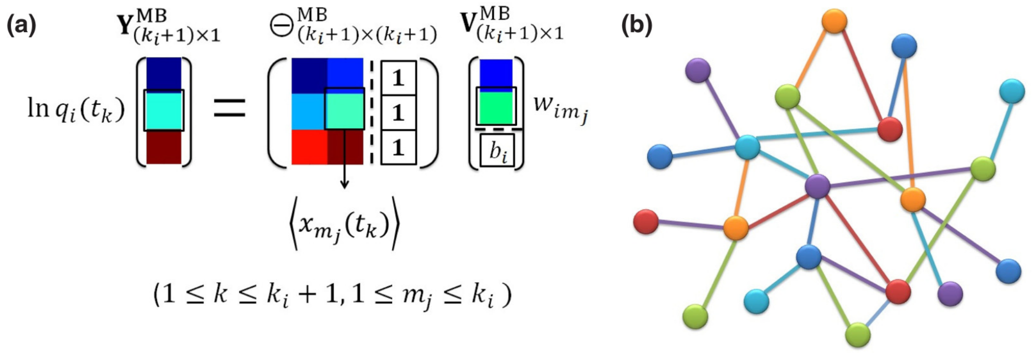
(a)
$\mathbf{Y}^{\mathrm{MB}}_{(k_i+1)\times 1}$
$\ominus^{\mathrm{MB}}_{(k_i+1)\times(k_i+1)}$
$\mathbf{V}^{\mathrm{MB}}_{(k_i+1)\times 1}$
$\ln q_i(t_k)$
=
1
1
1
w_{im_j}
b_i
$\langle x_{m_j}(t_k)\rangle$
$(1 \le k \le k_i+1, 1 \le m_j \le k_i)$
(b)

FIG. 3. A schematic illustration of SDBM parameter estimation for undirected complex networks. (a) For the SDBM of the same system in Figs. 1(a) and 1(b), the corresponding parameter estimation framework as in Eq. (11). The connections of the network have already been reconstructed, so the entries in Eq. (11) can be obtained from a node's Markov blanket [Fig. 1(b)]. The calculation is implemented for each node in the system. (b) The values of the link weights and the nodal biases (different colors) for the corresponding SDBM are calculated via the parameter estimation scheme in (a) and the degree guided solution substitution operation (Fig. 4).

problem. Exploiting the available data further, we develop a degree guided solution-substitution operation to cope with this difficulty, where sufficient estimation precision is guaranteed for most cases. An SDBM with estimated parameters is schematically shown in Fig. 3(b).

2. Link weights and nodal bias estimation

For each node i, we rank the occurrences of all the existing configurations of its neighbor's states. Among the top ones, $k_i + 1$ are selected to ensure that the coefficient matrix $\ominus^{\mathrm{MB}}_{(k_i+1)\times(k_i+1)}$ on the right-hand side of Eq. (11) has full rank so that the solutions $w_{im_1}, w_{im_2}, \ldots, w_{im_{k_i}}, b_i$ are unique (the selected configurations are not necessarily on the exact top since the real top $k_i + 1$ ones may not provide a full-rank coefficient matrix).

Due to insufficient number of samples or some particular features of a specific dynamical process $\langle x_i(t+1)\rangle$, the statistical estimation of $P\{x_i(t+1)|\mathbf{X}_i^{\mathrm{MB}}(t)\}$ [or $P\{x_i(t+1)|\mathbf{X}_i^{\mathrm{R}}(t)\}$], can be 0 or 1, which respectively makes $1/\langle x_i(t+1)\rangle$ or $\ln[1 - 1/\langle x_i(t+1)\rangle]$ in Eqs. (9) and (11) diverge. Without

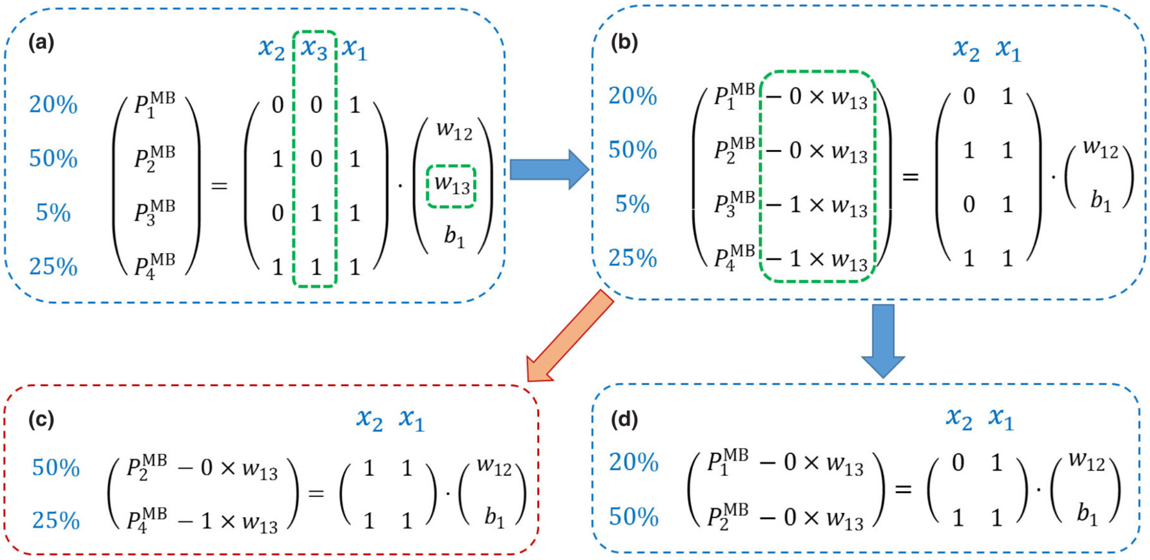
(a)
x_2 x_3 x_1
20% 50% 5% 25%
P_1^{MB} P_2^{MB} P_3^{MB} P_4^{MB}
= 0 0 1 / 1 0 1 / 0 1 1 / 1 1 1
w_{12} w_{13} b_1
(b)
x_2 x_1
20% 50% 5% 25%
$P_1^{\mathrm{MB}} - 0\times w_{13}$
$P_2^{\mathrm{MB}} - 0\times w_{13}$
$P_3^{\mathrm{MB}} - 1\times w_{13}$
$P_4^{\mathrm{MB}} - 1\times w_{13}$
= 0 1 / 1 1 / 0 1 / 1 1
w_{12} b_1
(c)
x_2 x_1
50% 25%
$P_2^{\mathrm{MB}} - 0\times w_{13}$
$P_4^{\mathrm{MB}} - 1\times w_{13}$
= 1 1 / 1 1
w_{12} b_1
(d)
x_2 x_1
20% 50%
$P_1^{\mathrm{MB}} - 0\times w_{13}$
$P_2^{\mathrm{MB}} - 0\times w_{13}$
= 0 1 / 1 1
w_{12} b_1

FIG. 4. An example of degree guided solution-substitution operation and final equation group construction for undirected networks. Target node 1, the node whose local connecting topology is to be inferred, has only two neighbors, nodes 2 and 3. Without loss of generality, we set $k_3 < k_1 = 2 < k_2$ so that the link weight between nodes 1 and 3, w_{13}, is already obtained. The quantity $P_s^{\mathrm{MB}} = \ln[1/\langle x_1(t+1)\rangle_s - 1]$ [$s = 1,\ldots,4$, as there are $2^{k_1} = 4$ possible (x_2,x_3) configurations in total] denotes the estimated probability for $x_1 = 1$ under configuration s at the next time step. The occurring probabilities of the corresponding (x_2,x_3) configurations are listed as percentages in the left column of each panel. (a) All four (x_2,x_3) configurations are shown by the first two columns of the 4×3 matrix in the equation. (b) The contribution of the known w_{13} and the configurations of x_3 as marked in (a) are moved to the left-hand side of the equation. (c) In order to solve for $(w_{12},b_1)^{\mathrm{T}}$, the two most precise P_s^{MB} values ($s = 2$ and 4) corresponding to the most frequent (x_2,x_3) configurations from (b) are chosen to build a linear 2-equation group. However, the 2×2 matrix on the right-hand side does not have full rank and, hence, this equation group is ill defined (d) Configurations $s = 1$ and 2 are selected to construct a 2-equation group with a full-rank 2×2 matrix on the right-hand side and relatively precise probability estimations, and this equation group is used to solve $(w_{12},b_1)^{\mathrm{T}}$.

TABLE I. Description of the 14 dynamical processes on model networks and the conditional probabilities. The quantity $P_i^{0\rightarrow 1}$ or $P_i^{1\rightarrow 0}$ is the probability that the state of node i (of degree k_i) becomes 1 at the next time step when its current state is 0, or vice versa, conditioned on its current neighboring state configuration. The number of i's neighbors in state 1 is n_i.

Category	Dynamics type	$P_i^{0\rightarrow 1}$	$P_i^{1\rightarrow 0}$
I	SDBM	$\frac{1}{1+\exp[\sum_{j=1,j\neq i}^{N} w_{ij}x_j(t)+b_i]}$	$\frac{\exp[\sum_{j=1,j\neq i}^{N} w_{ij}x_j(t)+b_i]}{1+\exp[\sum_{j=1,j\neq i}^{N} w_{ij}x_j(t)+b_i]}$
II	Ising-Glauber [53]	$\frac{1}{1+\exp[\frac{2J}{\kappa}(k_i-2n_i)]}$	$\frac{\exp[\frac{2J}{\kappa}(k_i-2n_i)]}{1+\exp[\frac{2J}{\kappa}(k_i-2n_i)]}$
	SQ-SG [54]	$\frac{1}{1+\exp[(rk_i-n_i)/\kappa]}$	$\frac{\exp[(rk_i-n_i)/\kappa]}{1+\exp[(rk_i-n_i)/\kappa]}$
	SQ-PDG [54]	$\frac{1}{1+\exp[(b-1)(k_i-n_i)/\kappa]}$	$\frac{\exp[(b-1)(k_i-n_i)/\kappa]}{1+\exp[(b-1)(k_i-n_i)/\kappa]}$
III	Minority game [55–59]	$\frac{k_i-n_i}{k_i}$	$\frac{n_i}{k_i}$
	Voter [60,61]	$\frac{n_i}{k_i}$	$\frac{k_i-n_i}{k_i}$
	Majority vote [60,61]	$\begin{cases} Q & \text{if } n_i < k_i/2 \\ 1/2 & \text{if } n_i = k_i/2 \\ 1-Q & \text{if } n_i > k_i/2 \end{cases}$	$\begin{cases} 1-Q & \text{if } n_i < k_i/2 \\ 1/2 & \text{if } n_i = k_i/2 \\ Q & \text{if } n_i > k_i/2 \end{cases}$
IV	Link-update voter [62,63]	$\frac{n_i}{\langle k\rangle}$	$\frac{k_i-n_i}{\langle k\rangle}$
	Language model[64,65]	$S(\frac{n_i}{k_i})^{\alpha}$	$(1-S)(\frac{k_i-n_i}{k_i})^{\alpha}$
	Kirman [66,67]	c_1+dn_i	$c_2+d(k_i-n_i)$
V	CP [68,69]	$\frac{n_i}{k_i}\lambda_i$	μ_i
	SIS [70–73]	$1-(1-\lambda_i)^{n_i}$	μ_i
VI	SG [74]	*	*
	PDG [75,76]	*	*

loss of generality, we set $\langle x_i(t+1)\rangle = \epsilon$ [or $\langle x_i(t+1)\rangle = 1-\epsilon$] if $\langle x_i(t+1)\rangle = 0$ (or 1), where ϵ is a small positive constant.

For node i of degree k_i, the total number of possible neighboring state configurations is 2^{k_i}, so large degree nodes have significantly more configurations. The process may then regard every particular type of configuration as useful and accordingly lead to inaccurate estimate of $P\{x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)\}$. Our computations reveal that properly setting a tolerance in the Hamming distance that allows similar but not exactly identical configurations to be treated as the same $\mathbf{X}_i^{\text{MB}}(t)$ can improve the estimation performance. For a particular neighboring state configuration with too fewer identical matches in the observed data, configurations with difference in one or more digits in the Hamming distance are used instead, until a sufficient number of matches are found. For nodes of degrees larger than, say 15, the Hamming distance tolerances within 2 or 3 usually lead to a reasonable number of matches for estimating $P\{x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)\}$. Extensive calculation shows that the small inaccuracy has little effect on the reconstruction accuracy.

3. *Degree guided solution substitution operation*

For node i's neighbors, the occurring frequency of a particular state configuration is generally higher for smaller values of k_i. Accordingly, the probability estimation conditioned on this configuration can be more accurate for smaller values of k_i. More precise conditional probabilities in turn lead to a higher estimation accuracy of link weights and node biases. Similar to resolving solution contradiction in structural reconstruction, for a pair of linked nodes, i and j of different degrees, w_{ij} solved from the equation group of node i is likely to be more accurate than w_{ji} obtained via node j, if $k_i < k_j$, even ideally they should have the identical values. As a result, using the solution obtained from a lower degree node as the value of the link weight provides a better estimation. We run the calculation for all nodes in the network one by one in a degree-increasing order so that the link weights of the smallest degree nodes are acquired earlier than for other nodes. For the larger degree nodes, the link weights shared with the smaller degree nodes are substituted by the previously obtained values, which are treated as known constants instead of variables waiting to be solved. This operation effectively removes the contribution of the lower degree neighbors from the equation and so reduces the number of unknown variables. A reduced equation group in a form similar to Eq. (11) can be built up based on the most frequently occurring state configurations of the remaining neighbors. When the full-rank condition is met, maximum possible precision of the remaining unknown link weights can be achieved. Consequently, with the substitution operation, the k_i link weights of node i, which can be inaccurate if solved from the original (k_i+1)-equation group, can be estimated with the maximum possible accuracy. An example of this substitution operation and the final equation group construction process is shown in Fig. 4.

Our calculations show that the application of the degree guided solution-substitution operation can significantly increase the accuracy in the estimation of the link weights and node biases. For undirected complex networks, this operation makes the SDBM a reliable approximator of the original network dynamics. For example, as will be described, for some dynamical processes described in Table I, the estimation errors are dominantly distributed at zero. For processes in other categories, aside from the dominant peak at zero, there exist multiple small peaks at nonzero error values. These small side peaks are a consequence of the model complexity of the Markov networks.

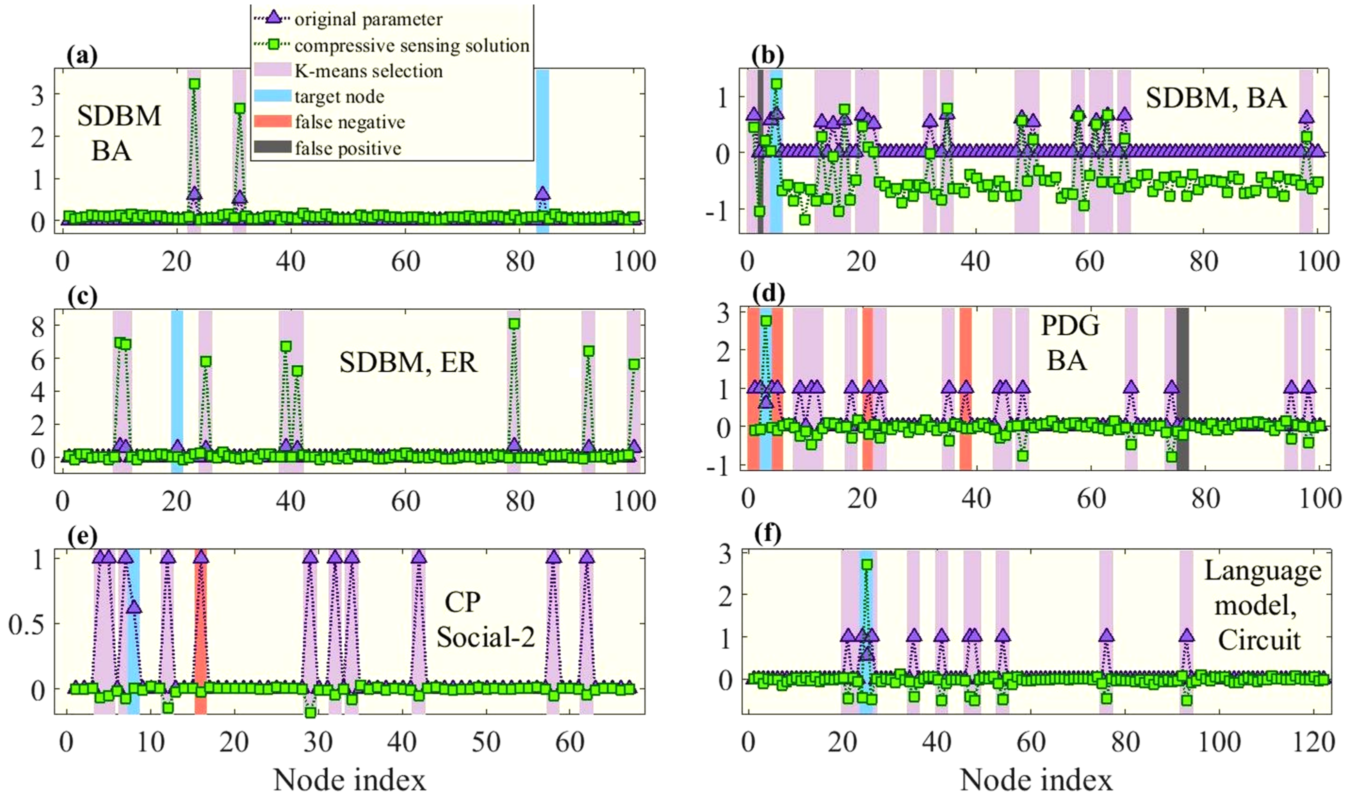

FIG. 5. Solution examples of compressive sensing and K-means clustering results. Presented are the reconstruction results for a single node, the target node, for different networks and dynamical processes. Each panel is for a specific network structure and a specific type of dynamical process, where the actual and null links associated with the target node are indicated. The compressive sensing solutions (green squares) are compared, elementwise, with the true values of the original connection vectors (dark purple triangles), where the x axis is the index of nodes with different degree values for different network topologies and dynamics types. The target nodes and the correctly detected existent connections via the K means are marked as light blue and light purple bars, respectively. The false negative and the false positive estimations are marked as red and gray bars, respectively. For all cases, the measurement amount is $M = 0.4N$. The restriction on the Hamming distances is set to be $\Gamma = 0.35$ for all examples, and a small change in Γ would not affect the qualitative results. (a) A node with the smallest degree $k = 2$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (b) The node of the largest degree $k = 18$ in the same network. (c) A node of degree $k = 8$ in an ER random network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (d) The node with the largest degree $k = 19$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with prisoner's dilemma game (PDG) dynamics. (e) A node of degree $k = 11$ in a real world social network of size $N = 67$ subject to CP dynamics. (f) A node of degree $k = 10$ in a real world electrical circuit network of size $N = 122$ with language model dynamics. In (d)–(f), the element values of the original connection vector are set to unity since no true weight values are given. For cases (a), (b), (c), and (f), there are no false negative detections, i.e., all existent links have been successfully detected. For cases (a), (c), and (e), there are no false positive detections, i.e., no nonexistent links are mistaken as existent ones. A detailed description of the dynamical processes is given in Table I.

III. RESULTS

A. Structural estimation: An illustrative example

Table I lists the 14 dynamical processes on model networks and the underlying conditional probabilities. The implementation of these processes on complex networks is detailed in Appendix C. For each type of dynamics, 10 different network realizations are generated for both Barabasi-Albert (BA) scale-free and Erdös-Rényi (ER) random topologies. For any node in a particular network realization, the corresponding connection weight vector $\mathbf{V}_{N\times 1}$ is obtained by averaging over 100 compressive sensing implementations. The elements corresponding to the existent nodes are distinguished from the nonexistent links by feeding the averaged $\mathbf{V}_{N\times 1}$ into the K-means algorithm. The unweighted adjacency matrix is then obtained. In practice, for very large networks, longer time series and higher computational resource are required due to the exponential increase in the number of system configurations, but theoretically there is no upper bound in the network size. The data amount is characterized by the number of measurements M normalized by N.

To gain insights, we test the structural estimation method for an SDBM itself [Figs. 5(a)–5(c)] and three different types of dynamical processes [Figs. 5(d)–5(f)] by feeding the time series data into the framework and comparing the reconstructed SDBM with the original machine. The time instants $t_1, t_2, \ldots, t_M$ needed to construct Eq. (9) are chosen randomly from T time instants in total. For each node, compressive sensing is implemented a multiple of times to obtain the averaged relevant quantities. As shown in Fig. 5, the averaged solutions from the compressive sensing algorithm appear as sharp peaks at places that correspond to the existent

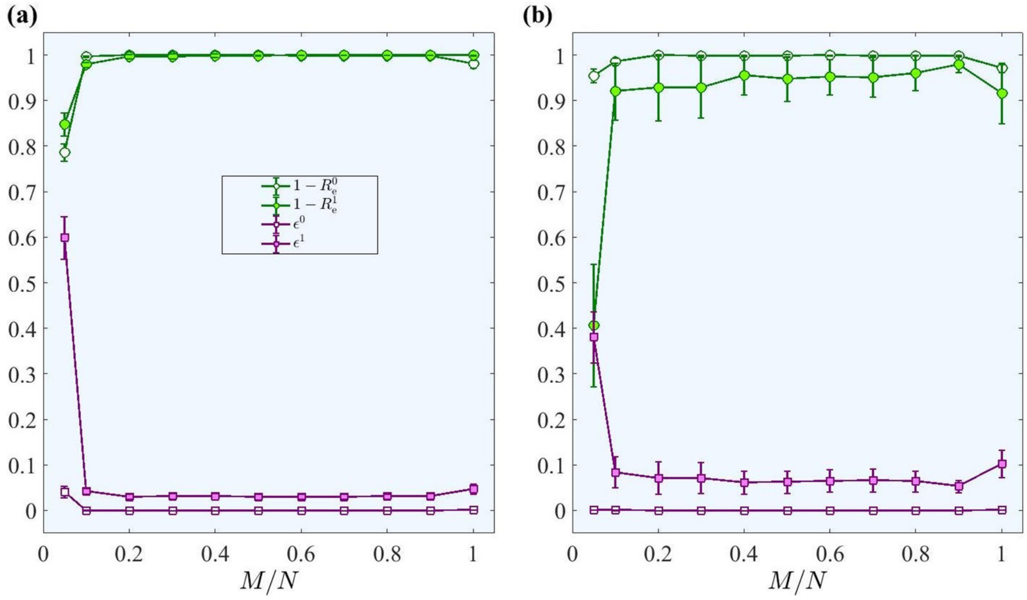

FIG. 6. Structural estimation success rates and parameter estimation errors versus the normalized data amount. For SDBM dynamics on (a) ER random and (b) BA scale-free networks, the success rates of detection of existent links and identification of nonexistent links, $1 - R_e^1$ (green circles) and $1 - R_e^0$ (white circles), respectively, versus the normalized data amount M/N with error bars. The absolute values of the errors in estimating the link weights and the nodal biases for the existent links (purple squares) and nonexistent links (white squares) are also shown. Most data points have quite small error bars, indicating the performance stability of our method. High performance structural and parameter estimations are achieved insofar as M/N exceeds about 10%.

links, despite the large differences among the values. This means that, while most existent links can be predicted against the null links, the accuracy of the solution so obtained is not sufficient for the actual element values of $\mathbf{V}_{N\times 1}$ to be determined. For the null links, the corresponding solutions generally appear as a noisy background. For the ideal case where w_{ij} is zero, the background noise can be quite large especially for the large degree nodes, as shown in Figs. 5(b) and 5(d). Further calculations indicate that averaging a larger number of simulation runs can suppress the background noise to certain extent, but it cannot be eliminated and may become quite significant for various types of dynamics.

B. Performance of SDBM for undirected networks

Our working hypothesis is that, for a networked system with a certain type of nodal dynamics, there exists an equivalent SDBM. Reconstructing the structure of the SDBM would simultaneously give the topology of the original networked dynamical system. Accordingly, the time series data generated from the original system can be used to reveal its underlying interaction structures through the corresponding SDBM. In particular, we directly feed the original time series into the framework of compressive sensing and K means for SDBM to generate the network structure, and test whether the structure generated by the SDBM based reconstruction represents that of the original network. The similarity can be quantified by the error rates of the existent and nonexistent links.

Figure 6 demonstrates the performance of network structural estimation for SDBM dynamics on random and scale-free networks in terms of the data amount for SDBMs built up for systems with different original topologies. We define R_e^1 and R_e^0 as the estimation error rates for the existent and nonexistent links, namely, the false positive and the false negative rates, respectively. We see that, for a wide range of the values of M/N, the success rates of the existent and nonexistent links, $1 - R_e^1$ and $1 - R_e^0$, respectively, are nearly 100% for homogeneous network topology. For heterogeneous (scale-free) networks, the success rate $1 - R_e^1$ tends to be slightly lower than 100%, due to the violation of the sparsity condition for hub nodes, leading to the difficulty in distinguishing the peaks in the distribution of the compressive sensing solutions from the noisy background. For $M/N = 1$, the success rates are reduced slightly, due to the nonzero dissimilarity tolerance Γ that introduces a linear dependence between different equations into Eq. (9). The reason is that, in a compressive sensing problem, the M measurements are required to be linearly independent of each other. However, if M is too large, linear independence between each pair of measurements may be lost, due to the finite length of the available time series. Since each measurement in Eq. (9) is from the average of many configurations that are similar but not exactly identical, for a large value of the tolerance parameter Γ, some configurations can be similar in more than one measurement and appear in multiple equations, destroying the linear independence between those measurements. Thus, increasing the value of M would compromise the linear independence of the equations, leading to poor reconstruction performance. Our calculation shows that this situation does not arise unless M is quite close to N. Since M must be smaller than N and the solution sought is sparse, the algorithm can indeed be regarded as a compressive sensing problem. Empirically, for computational efficiency, M is chosen below $0.5N$ (e.g., is $M = 0.4N$), which makes our SDBM framework free of any subjective parameters.

TABLE II. Reconstruction error rates for undirected networks. For the same dynamical processes in Table I, the error rates (in percentages) in uncovering the existent and nonexistent links R_e^1 and R_e^0, respectively. ER random and BA scale-free networks of size $N = 100$ and average degree $\langle k \rangle = 4$ are used as the underlying network supporting the various dynamical processes. The normalized number of measurements used in compressive sensing is $M/N = 0.4$ for all cases. All results for comparison (in parentheses) except one for the CP dynamics, which are available from the current literature, are from Ref. [35]. The comparison results for CP are from Ref. [28].

Category	Dynamics type	$R_\mathrm{e}^0/R_\mathrm{e}^1$ (%, ER)	$R_\mathrm{e}^0/R_\mathrm{e}^1$ (%, BA)
I	SDBM	0.0 / 0.1	0.1 / 3.4
II	Ising-Glauber [53]	0.0/0.0 (0.0/0.0)	0.0/0.0 (0.3/0.8)
	SQ-SG [54]	0.0 / 0.0	0.0 / 3.2
	SQ-PDG [54]	0.0 / 0.0	0.3 / 3.3
III	Minority game [55–59]	0.0 / 0.6	0.0 / 0.2
	Voter [60,61]	0.0/0.0 (0.0/0.0)	0.0/0.0 (0.2/0.6)
	Majority vote [60,61]	0.0/0.0 (0.0/0.0)	0.0/0.1 (0.1/0.3)
IV	Link-update voter [62,63]	0.0 / 0.0	1.8 / 3.6
	Language model [64,65]	0.0/0.0 (0.0/0.0)	0.0/0.4 (1.6/3.3)
	Kirman [66,67]	0.0/2.3 (0.0/0.0)	0.0/3.8 (7.0/14.2)
V	CP [68,69]	0.0/1.4 (0.1/0.0)	0.0/0.4 (0.3/0.0)
	SIS [70–73]	0.3/5.2 (0.2/0.5)	0.1/15.6 (2.3/1.4)
VI	SG [74]	0.0 / 1.7	0.0 / 9.4
	PDG [75,76]	0.1/9.6 (0.0/0.0)	0.1/9.8 (0.7/1.6)

Table II shows the performance in terms of the percentage error rates R_e^0 and R_e^1. The 14 types of dynamical processes are taken from the fields of evolutionary game theory, opinion dynamics, and spreading processes, covering a number of focused research topics in complex networks. Strikingly, for all the dynamics with diverse properties, we find that, for each and every dynamics-network combination, zero or nearly zero error rates are obtained for both the existent and nonexistent links, revealing a strong similarity between the original networks and the ones generated from SDBM, regardless of the type of dynamics. The nonzero error rates in Table II come mainly from the high degree nodes. Consequently, as indicated in Table II, the reconstruction accuracy for networks of homogeneous topology is generally higher than that for heterogeneous networks.

For certain types of evolutionary game dynamics, especially for the snowdrift game (SG) [74] and the prisoner's dilemma game (PDG) [75,76] with the Fermi updating rule [76], information about the state configuration of the second nearest neighbors is required to calculate the payoffs of the first nearest neighbors. In such a system, the next move of a target node is determined by comparing its payoff with those of its neighbors. This implies that, using solely the state configuration information of the Markov blanket, without the aid of the payoff information that requires the state information of the second nearest neighbors, is insufficient to determine the state of the target node into the immediate future, rendering inapplicable our SDBM based reconstruction. Contrary to this intuition, we find that for both SG and PDG, high reconstruction accuracy can be achieved, as shown in Table II. Recall that our reconstruction method is formulated based on the independence assumption of Markov networks, i.e., in order to reconstruct the local structure of a target node, it should be completely independent of the rest of the system when the configuration of its Markov blanket is given. The results in Table II indicate that our SDBM based algorithm performs better than anticipated in terms of the reconstruction accuracy. In fact, the independence assumption can be made to hold by adopting a self-questioning (SQ) based updating rule. In this case, excellent reconstruction accuracy is obtained, as shown in Table II.

For evolutionary game dynamics (e.g., PDG and SG) and binary dynamical processes (e.g., CP and SIS), there were previous methods based on compressive sensing for network reconstruction [27,28]. In terms of the false positive and negative rates, our SDBM method in general does not offer better results. The reason is that the reconstruction task is still essentially based on compressive sensing. Thus, from the standpoint of inferring network structures, our SDBM method is not advantageous as compared with the previous methods. However, the SDBM method is applicable to wider classes of network dynamical processes. An additional appealing feature of the SDBM method lies in its ability to create a "replica" of the underlying networked dynamical system based solely on data. This may have potential applications in complex systems identification and prediction.

The structural estimation results reported so far are based on model network topologies. Real world complex networks have also been used to test our framework, with results exemplified in Figs. 5(e) and 5(f) (see also Appendices D and E). For various combinations of the network topology and dynamical process, high reconstruction accuracy is achieved, where for a number of cases the error rates are essentially zero. There are a few special cases where the errors are relatively large, corresponding to situations where the globally frozen or oscillating states dominate the dynamical process so that too few linearly independent measurements can be obtained. Overall, the equivalent SDBM correspondence holds and our reconstruction scheme for real world networks is effective.

C. Dynamics approximator for undirected complex networks

In Table I, the complex network dynamics is categorized in terms of the specific probabilities $P_i^{0\to 1}$ and $P_i^{1\to 0}$. In

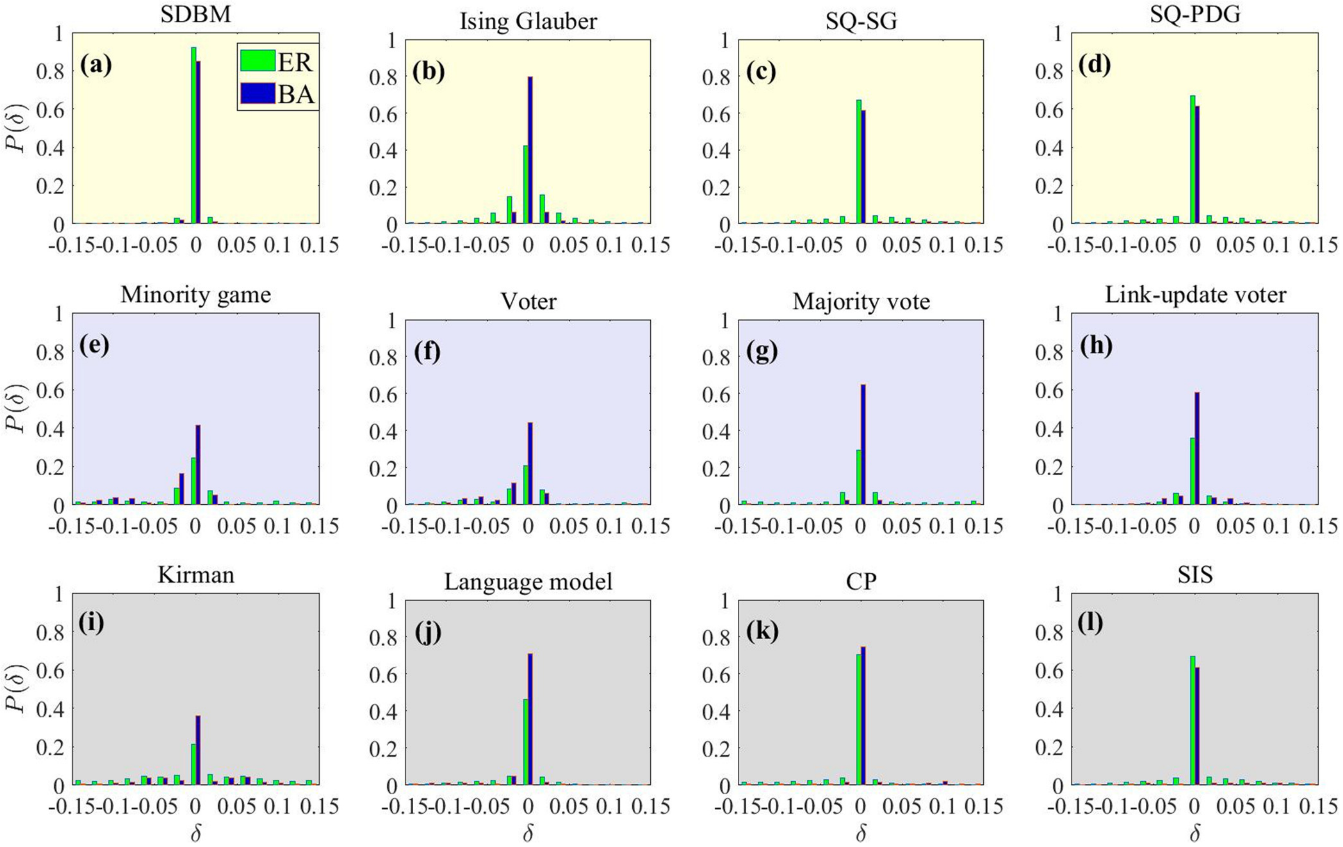

FIG. 7. Distributions of the conditional probability estimation errors for various complex network dynamics. The distributions of $\delta = |P\{x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)\} - P^{\mathrm{est}}[x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)]|$ for the dynamical processes in categories I to V from Table I for ER random (green bars) and BA scale-free (blue bars) networks, where $P^{\mathrm{est}}[x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)]$ denotes the conditional probability estimated from the corresponding SDBM approximator through the original state configuration times series, where $1 \leqslant i \leqslant N$ and $1 \leqslant t \leqslant T$.

particular, category I is for the SDBM dynamics and category II contains dynamical processes with $P_i^{0\to1}$ and $P_i^{1\to0}$ having a mathematical form similar to that of the SDBM. For category III, the forms of $P_i^{0\to1}$ and $P_i^{1\to0}$ for the three types of dynamics are quite different. Despite the differences among the dynamical processes in the three categories, they share a key property, i.e., $P_i^{0\to1} + P_i^{1\to0} = 1$, which plays a critical role in implementing the parameter estimation algorithm. For the processes in categories IV and V, we have $P_i^{0\to1} + P_i^{1\to0} \neq 1$, where $P_i^{1\to0}$ is a time-invariant constant for the nodal dynamics in category V.

We first test our parameter estimation scheme using the state time series generated by the SDBM, which can be validated through a direct comparison of the estimated parameter values with their true values, as exemplified in Figs. 5(a)–5(c). However, since the parameters only affect the system collectively (not individually) in the form of the product summation [Eq. (5)] and our goal is to assess the predictive power, we introduce an alternative validation scheme. For each particular system configuration, the acceptable parameter estimation results would serve as a base to generate another SDBM with identical conditional probabilities for each node at each time step, as compared with those of the original system. (A visual comparison between the conditional probabilities of the original system and that generated via the reconstructed SDBM is presented in Appendix F.) Figure 7(a) shows the error distributions of the estimated conditional probability time series, where an overwhelmingly sharp peak occurs at 0, indicating an excellent agreement between the estimated and the true parameter values.

For typical dynamical processes on complex networks, our goal is to find the equivalent SDBMs whose true parameter values are not available. In this case, the performance of the parameter estimation scheme can be assessed through the reconstructed conditional probabilities. Based on the recovered network structure, the time series of the network dynamics are fed into the parameter estimation scheme, and the link weights and the nodal biases are obtained to form a system obeying the SDBM dynamics. Corresponding to the categories in Table I, the distributions of the estimation errors between the original conditional probability $P\{x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ and the one generated by the newly constructed SDBM are shown in Fig. 7, where Figs 7(a)–7(f) show the error distributions corresponding to the six types of dynamics in categories II and III in Table I, respectively. In each case, a sharp peak at zero dominates the distribution, indicating the equivalence of the reconstructed SDBM to the original dynamics. Given a particular type of complex network dynamics, the SDBM resulting from our structural and parameter estimation framework is indeed equivalent to the original dynamical system. The limited amount of data obtained from the original system renders important state prediction of the system, a task that can be accomplished by taking advantage of the equivalence of the SDBM to the original system in the sense that the SDBM produces approximately equal state transition probabilities in the immediate future, given the current system configuration. The SDBM thus possesses a significant predictive power for the original system. Regardless of the type of the dynamical process, insofar it satisfies the relation $P_i^{0\to 1} + P_i^{1\to 0} = 1$, the reconstructed SDBM can serve as a dynamics approximator.

For an SDBM, the relation $P_i^{1\to 1} = 1 - P_i^{1\to 0} = P_i^{0\to 1}$ holds in general. However, for the dynamical processes in categories IV and V, we have $P_i^{1\to 1} \neq 1 - P_i^{1\to 0} = P_i^{0\to 1}$ so that a single SDBM is not sufficient to fully characterize the dynamical evolution. Our solution is to construct two SDBMs, A and B, each associated with one of the two cases: $x_i(t) = 0$ and $x_i(t) = 1$, respectively. The link weights w_{ij}^{A} (or w_{ij}^{B}) and the nodal bias b_i^{A} (or b_i^{B}) for node i in SDBM A (or B) are computed for all the time steps t satisfying $x_i(t) = 0$ [or $x_i(t) = 1$], leading to $P\{x_i(t+1) = 1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ for $x_i(t) = 0$ [or $x_i(t) = 1$] from SDBM A (or B). Using this strategy, the dominant peaks at zero persist in the error distributions for the dynamical processes in category IV, as shown in Figs. 7(g)–7(j). For the epidemic spreading dynamics (CP and SIS) in category V, the fixed value of $P_i^{1\to 0}$ for each node i can be acquired through $P_i^{1\to 0} \simeq \langle x_i(t_1+1)\rangle_{t_1}$, where $\langle x_i(t_1+1)\rangle_{t_1}$ stands for the average of $x_i(t_1+1)$ over all values of t_1 satisfying $x_i(t_1) = 1$. Through this approach, SDBM B is in fact a network without links but with each node's bias satisfying $\mu_i = \exp(b_i)/[1+\exp(b_i)]$, where μ_i is node i's recovery rate. Figures 7(k) and 7(l) show the error distributions of the conditional probability recovery for the spreading processes, where we see that the errors are essentially zero. If the method of solving SDBM B in category IV is adopted to the dynamical processes in other categories, i.e., without any prior knowledge about $P_i^{1\to 0}$, the resulted SDBM would have nearly identical link weight values with respect to SDBM A (in categories I–III) or have close-to-zero link weights and $\mu_i \simeq \exp(b_i)/[1+\exp(b_i)]$ for category V, despite that their conditional probability recovery errors may be slightly larger than those in Fig. 7. The persistent occurrence of a dominant peak at zero in the error distribution suggests the power of combined SDBMs as a dynamics approximator, regardless of the specifics of the transition probability. When limited prior knowledge about $P_i^{1\to 0}$ is available, SDBM B can be simplified or even removed without compromising the estimation accuracy. In general, the approximator has a significant short-term predictive power for arbitrary types of dynamics on complex networks.

The conditional probability recovery error is called the "training error" since it is obtained from the same data set used to build (or "train") the approximator, and the data points generated from the same system that have not been used in the training process can be exploited to validate or test the actual performance of the trained model [38], which in our case is the approximator. As a result, the time series generated from the original complex network system after the approximator is built can be used as the test data set. (Absence of hyperparameters in the reconstruction process means that cross validation is unnecessary.) In most cases, the training errors are generally smaller than the test errors since the training data set is already well fitted by the model (the approximator) in the training process, while the test data are new and may be out of the fitting range of the current model. Feeding the state configurations of the test data set into the approximator, we calculate the corresponding conditional probabilities using Eq. (5) and compare them to the true values. The results reveal a clear advantage of the approximator built from our scheme, i.e., the training and test errors are nearly identical, indicating the absence of any overfitting issues [38].

D. Extension to directed complex networks

Our SDBM methodology for undirected networks can be adapted to directed networks because of its nodewise implementation scheme. In a directed network, the link weights between nodes i and j, w_{ij} and w_{ji}, do not have equal values. In fact, these are weights of the two links in the opposite directions between the same pair of nodes. The weights solved by Eqs. (9) and (11) correspond to all the links pointing to node i and all pointing from i's nearest neighbors. By estimating the values of the inward link weights for each node in the network, we obtain a directed SDBM as a structural estimator and dynamics approximator of the underlying directed network. The structural and parameter estimation processes are illustrated in Figs. 8(a) and 8(b). Since the conflict resolution scheme designed for undirected networks applies to symmetric links only, it is not adopted for directed networks. The structural estimation results are presented in Table III. In most cases, we observe only small changes in the precision as compared with the undirected case. For some types of dynamics, R_{e}^{0} increases slightly, indicating that the main challenge of reconstructing directed networks is to reduce the false positives. Nonetheless, in general, the

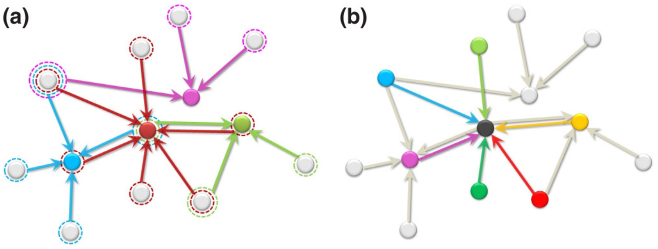

FIG. 8. A schematic illustration of reconstruction of network structure and parameter estimation for directed complex networks. (a) Reconstruction of the local inward connection structure of the blue, red, pink, and green nodes in a directed network of 13 nodes. Among these four nodes, there are bidirectional links between the blue and the red nodes and between the red and the green nodes. Links between other pairs of nodes are directional, if they exist. Executing compressive sensing and K-means clustering for each of the four nodes identifies its true incoming links, marked by the links with the same color of the node and pointing inward. (b) For the SDBM of the system in (a), the parameter estimation framework is implemented for the black node, and its incoming links are marked with different colors, indicating different values in general. The calculation is repeated for every node in the system.

TABLE III. Reconstruction error rates for directed networks. For the same dynamical processes in Table I, the error rates (in percentages) in uncovering the existent and nonexistent links $R_{\rm e}^1$ and $R_{\rm e}^0$, respectively. ER random and BA scale-free networks of size $N = 100$ and average degree $\langle k \rangle = 4$ are used as the underlying network supporting the various dynamical processes. The normalized number of measurements used in compressive sensing is $M/N = 0.4$ for all cases.

Category	Dynamics type	$R_{\rm e}^0$/ $R_{\rm e}^1$ (%, ER)	$R_{\rm e}^0$/ $R_{\rm e}^1$ (%, BA)
I	SDBM	0.0 / 0.1	0.2 / 7.11
II	Ising-Glauber [53]	0.0 / 0.0	0.0 / 0.0
	SQ-SG [54]	0.0 / 0.0	0.0 / 1.3
	SQ-PDG [54]	0.5 / 1.2	0.7 / 2.5
III	Minority game [55–59]	0.0 / 1.6	0.0 / 0.0
	Voter [60,61]	0.0 / 0.0	0.3 / 0.5
	Majority vote [60,61]	0.0 / 0.0	0.0 / 0.1
IV	Link-update voter [62,63]	0.1 / 1.1	0.6 / 2.0
	Language model [64,65]	0.2 / 3.6	0.0 / 0.0
	Kirman [66,67]	0.0 / 0.0	0.1 / 2.5
V	CP [68,69]	0.0 / 2.1	0.0 / 2.5
	SIS [70–73]	1.6 / 4.5	1.2 / 14.2
VI	SG [74]	0.1 / 0.9	0.1 / 13.5
	PDG [75,76]	0.2 / 4.4	0.3 / 11.7

results demonstrate the effectiveness of the SDBM method for directed complex networks.

For a node (say i) in an undirected network, if the procedure yields an incoming link from node j, then j must necessarily have an outgoing link to i, with $w_{ij} \approx w_{ji}$ (except for a few hub nodes, where the local sparsity condition is violated with relatively large reconstruction errors). This property of approximately symmetric interactions, however, does not hold for most nodes in a directed network. This provides a practical strategy to distinguish an undirected from a directed network when the symmetry of the network topology is unknown *a priori*: if most of the reconstructed links are symmetric (asymmetric), there is a high likelihood that the underlying network is undirected (directed). Ambiguities can arise for the links associated with the hub nodes, for which the SDBM method would fail. Since real complex networks are typically sparse, the hub nodes are relatively few, and we expect that our method is still capable of providing an approximate picture of the interaction patterns as to whether majority of the links are undirected or directed.

IV. DISCUSSION

Reconstructing complex dynamical systems from data is a frontier field in network science and engineering with significant applications. We focus on binary dynamical processes and ask the following question: Is it possible to build a "machine" to reconstruct, from data only, the underlying complex networked dynamical system and to make predictions? While this paper does not provide a mathematically rigorous solution, significant and (in some cases) striking results are obtained, which give strong credence that such a machine is possible. In particular, we combine compressive sensing and clustering algorithm to construct a general class of network structural estimators and dynamics approximators. For networks with symmetric or asymmetric interactions, universality is fundamentally possible due to the fact that many dynamical processes on complex networks are of the Markov type and the interactions among the nodes are local. As a result, utilizing basic tools from statistical physics, we can build up an energy based Markov-network-like model (e.g., an SDBM) to construct a estimator and approximator for different types of complex network structures and dynamics. For a large number of representative dynamical processes studied in this paper, we demonstrate that such a SDBM can be reconstructed based on compressive sensing and the scheme of K means using data only, without requiring any extra information about the network structure or the dynamical process. The working of the SDBM is demonstrated using a large number of combinations of the network structure and dynamics, including many real world networks and classic evolutionary game dynamics. An SDBM with its parameters given by the equations constructed from the time series along with the estimated network structure is able to reproduce the conditional probabilities quantitatively and, accordingly, it is capable of predicting the state configuration at least in a short term. We demonstrate that, for certain types of binary dynamics, the approximator can reproduce the dynamical process statistically, indicating the potential of its serving as a generative model for long-term prediction in such cases (Appendix G).

While we assume binary dynamics, in principle the methodology can be applied to other types of dynamical processes. In particular, Eq. (4) can be readily extended to the conditional probability of each possible state value λ_j, i.e., $P\{x_i(t+1) = \lambda_j|\mathbf{X}_i^{\rm R}(t)\} = P\{x_i(t+1) = \lambda_j, \mathbf{X}_i^{\rm R}(t)\}/\sum_s P\{x_i(t+1) = \lambda_s, \mathbf{X}_i^{\rm R}(t)\}$. A potential difficulty is that the configuration space of the system states grows exponentially with the number of choices of λ_j so that, given a finite amount of data, the number of time instances corresponding to each particular configuration decreases exponentially, leading to a significant reduction in the estimation precision of the conditional probability. For network structural reconstruction, in principle there is no theoretical limitation. A practical limitation is that, due to the exponential growth of the number of possible configurations as the network size is increased, the required computations would increase dramatically. In our study, the largest network tested is a real world circuit network of 512 nodes. Dealing with networks of more than a few thousand nodes would be computationally extremely demanding at present. The sparsity condition as required by compressive sensing poses another practical limitation. In fact, performance on networks with high average degree is poor. Another limitation is that complete information (data) from all system components is needed. How to deal with systems where only partial information is available is an open issue.

Our effort represents an initial attempt to realize a general estimator and approximator, and the performance of our method is quite competitive in comparison with the existing reconstruction schemes designed for specific types of dynamics (on undirected networks) (see Table II). In realistic applications, the data obtained may be discontinuous or incomplete. In such cases, the short-term predictive power possessed by the estimator and approximator can be exploited to overcome the difficulty of missing data, as the Markov network nature of the SDBM makes backward inference possible so that the system configurations during the time periods of missing data may be inferred. When long-term prediction is possible, the approximator has the critical capability of simulating the system behavior and predicting the chance that the system state enters into a global absorption phase, which may find significant applications such as disaster early warning. Another interesting reverse-engineering problem lies in the mapping between the original dynamics and the corresponding parameter value distribution of the reconstructed SDBM. That is, a certain parameter distribution of the SDBM may indicate a specific type of the original dynamics. As such, the correspondence can be used for precisely identifying nonlinear and complex networked dynamical systems. It is also possible to assess the relative importance of the nodes and links in a complex network based on their corresponding biases and weights in the reconstructed SDBM for controlling the network dynamics. These advantages justify our idea of developing a machine for data-based reverse engineering of complex networked dynamical systems, calling for future efforts in this emerging direction to further develop and perfect the network structural estimator and dynamics approximator.

ACKNOWLEDGMENT

The authors would like to acknowledge financial support from the Vannevar Bush Faculty Fellowship program sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering and funded by the Office of Naval Research through Grant No. N00014-16-1-2828.

APPENDIX A: MODEL COMPLEXITY AND REPRESENTATION POWER

For dynamical processes in categories I and II, the transition probabilities $P_i^{0\to 1}$ all have the form $1/[1+\exp(An_i+Bk_i)]$, where A and B are constants, and n_i denotes the number of node i's neighbors whose states are 1. We have

$$[1+\exp(An_i+Bk_i)]^{-1} = \left\{1+\exp\left[\sum_{m=1}^{k_i} w_{im}x_m(t)+b_i\right]\right\}^{-1},$$

which gives

$$\sum_{m=1}^{k_i} Ax_m(t)+Bk_i = \sum_{m=1}^{k_i} w_{im}x_m(t)+b_i. \qquad \text{(A1)}$$

In the ideal case where an absolutely accurate estimate of $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ can be obtained, Eq. (A1) holds for any possible neighboring state configurations. We thus have $w_{im}=A$ and $b_i=Bk_i$, and the probabilities conditioned on these configurations sharing the same values of n_i in the approximator are all equal to $P_i^{0\to 1}$. This means that, in theory, the conditional probabilities can be reconstructed with zero errors. In fact, a one-to-one mapping between the coefficients indicates that the model complexity of the SDBM provides the approximator with sufficient representation power to model the dynamics processes in categories I and II. Practically, since the statistical estimations of $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ may not be absolutely identical under different neighboring configurations with the same values of n_i due to random fluctuations, we have $w_{im}\approx A$ so that the conditional probabilities generated by the approximator may differ from each other slightly and also from the true probability $P_i^{0\to 1}$. As a result, random recovery errors can occur.

For dynamical processes in categories other than I and II, the simple coefficient-mapping relation between the corresponding $x_m(t)$ terms on the two sides of Eq. (A1) become nonlinear, due to the fact that the specific forms of $P_i^{0\to 1}$ differ substantially from that of the SDBM. In this case, each particular neighboring state configuration produces a distinct equation. There are 2^{k_i} equations in total, while the number of unknown variables to be solved is only k_i+1 (w_{im} for $m=1,\dots,k_i$ and b_i). For $k_i\geqslant 2$, there are thus more equations than the number of unknown variables. As a result, the representation power of the SDBM approximator is limited by its finite model complexity so that, even in principle, the approximator may not be able to fully describe the original dynamical process. In our framework, we calculate the Markov link weights and the node biases according to the k_i+1 most frequently appeared neighboring state configurations. A consequence is that imprecise conditional probability estimations can arise for the configurations with relatively lower occurring frequency, giving rise to the nonzero peaks in the error distribution in Fig. 7. However, interestingly, with respect to the precision of the conditional probabilities produced by the approximator, the SDBM parameters do not show a significantly strong preference towards the most frequently occurring configurations. In practice, the estimated conditional probabilities for the majority of the less frequently occurring configurations also fall within a close range of the true values. This phenomenon suggests the power of the approximator to work beyond the limit set by its theoretical model complexity. This is the main reason for the emergence and persistence of the dominant peak at zero in the error distribution.

APPENDIX B: JUSTIFICATION FOR THE APPROXIMATION $\langle \ln Q\rangle \approx \ln q$

The approximation $\langle \ln Q\rangle \approx \ln q$ is used in the main text when deriving Eq. (9) from (7). For simplicity, we write $P\{x_i(t+1)=1,\mathbf{X}_i^{\mathrm{R}}(t)\}$ as p^1 in the following derivation. For one particular state configuration (measurement) in Eq. (7), we randomly shuffle all its similar configurations and partition them into l buckets of the same size. Letting $\ln(1/\tilde{p}_j^1-1)$ denote the estimation of $\ln Q$ in bucket j, we have

$$\begin{aligned}\langle \ln Q\rangle &= 7\left\langle \ln\left(\frac{1}{p^1}-1\right)\right\rangle = \frac{\sum_{j=1}^{l}\ln\left(1/\tilde{p}_j^1-1\right)}{l}\\ &= \ln\sqrt[l]{\prod_{j=1}^{l}\left(\frac{1}{\tilde{p}_j^1}-1\right)} \leqslant \ln\left[\frac{1}{l}\sum_{j=1}^{l}\left(\frac{1}{\tilde{p}_j^1}-1\right)\right]\\ &= \ln\left[\left\langle\frac{1}{p^1}\right\rangle-1\right], \end{aligned}\tag{B1}$$

where the equality holds if all the summation elements have equal values, so the geometrical and algebraic means are the same. When there are a sufficiently large number of similar configurations in each bucket, we have

$$\tilde{p}_1^1 \approx \tilde{p}_2^1 \approx \tilde{p}_3^1 \approx \ldots \approx \tilde{p}_l^1 \approx \langle x(t+1)\rangle \tag{B2}$$

and thus

$$\frac{1}{\tilde{p}_1^1} \approx \frac{1}{\tilde{p}_2^1} \approx \frac{1}{\tilde{p}_3^1} \approx \ldots \approx \frac{1}{\tilde{p}_l^1} \approx \frac{1}{\langle x(t+1)\rangle}, \tag{B3}$$

so that the equal sign in Eq. (B1) holds. Accordingly, we have $\langle \ln Q\rangle \approx \ln[\langle \frac{1}{p^1}\rangle - 1]$. Using supplemental Eqs. (B2) and (B3), we obtain

$$\left\langle\frac{1}{p^1}\right\rangle = \frac{1}{l}\sum_{j=1}^{l}\frac{1}{p_j^1} \approx \frac{1}{\langle x(t+1)\rangle}. \tag{B4}$$

Consequently, we have $\ln Q \approx \langle \ln Q\rangle \approx \ln q$.

This approximation can be further justified empirically. We have calculated $\ln q$ versus $\langle \ln Q\rangle$ for a wide range of the bucket size and found that, for different bucket size, the distributions all concentrate on the diagonal line with insignificant variances, indicating the effectiveness and accuracy of our approximation. Similar results have been obtained, regardless of network topology and dynamics property. We note that a similar approximation method was used by Shen *et al.* in their work of reconstruction of propagation networks from binary data [28].

APPENDIX C: IMPLEMENTATION DETAILS OF 14 DYNAMICAL PROCESSES ON COMPLEX NETWORKS

All types of network dynamical processes studied in the main text, except for PDG and SG, are implemented in the following way. At each time step t, the probabilities for node i to have a state $x_i(t+1) = 1$ and 0 at the next time step conditioned on the state configuration of i's neighbors, i.e., $P_i^{0\to 1}$ and $P_i^{1\to 0}$, are calculated according to Table I in the main text. If $x_i(t) = 0$ (or 1), then it is switched to state $x_i(t+1) = 1$ (or 0) with the probability $P_i^{0\to 1}$ (or $P_i^{1\to 0}$) or remains at 0 (or 1) with the probability $1 - P_i^{0\to 1}$ (or $1 - P_i^{1\to 0}$).

Listed below is a detailed description of the dynamical processes and the parameter settings. (Reasonable changes in parameter values do not affect the reconstruction performance.)

SDBM. The dynamics is described in detail in the main text. The network parameters (link weights and node biases) are uniformly chosen between 0.3 and 0.7 in our calculation.

Ising-Glauber. Opinion dynamics models often adopt the classic Ising model of ferromagnetic spins [53]. The temperature parameter κ characterizes the level of "rationality" of the individuals, which represents the uncertainties in accepting the opinion. The coupling strength parameter J (or the ferromagnetic-interaction parameter) characterizes the intensity of the interaction between the connected nodes. The simulation parameters are $\kappa = 1$ and $J = 0.1$.

Majority vote. The majority-vote model [60,61] is a nonequilibrium spin model. Spins tend to align with the neighborhood majority under the influence of noise with parameter Q quantifying the probability of misalignment. In our simulations, we set $Q = 0.3$.

Minority game. In the minority game model [55–59], each individual chooses the strategy adopted by the minority of its neighbors with a higher probability than that for the majority. The probability for node i to be 1 at the next time step is proportional to the number of its 0-state neighbors, $k_i - n_i$. We then have $P_i^{0\to 1} = (k_i - n_i)/k_i$ and $P_i^{1\to 0} = n_i/k_i$.

Voter and link-update voter. In the voter model [60,82], each node i adopts the state of one of its randomly selected neighbors. If i is currently inactive (state 0), while there are n_i neighbors in the active state (state 1), it becomes active with the probability $P_i^{0\to 1} = n_i/k_i$. An active node i becomes inactive with the probability $P_i^{1\to 0} = (k_i - n_i)/k_i$. The voter model is similar to the minority game model but with the definition of $P_i^{0\to 1}$ and $P_i^{1\to 0}$ swapped. The link-update voter model is a variant of the voter model [62,63].

Language model. In the language model [64,65], the 0 and 1 states stand for an individual's two primary choices of the language. The probability of switching language is proportional to the fraction of the speakers in its neighborhood, raised to the power α and multiplied by the status parameter s or $1 - s$ of the respective language. In our simulations, we use $\alpha = 0.5$ and $s = 0.6$.

Kirman ant colony model. In the model [66,67], to choose between stock market trading strategies, nodes move from the nonadopted state (state 0) to the adopted state (state 1) with the probability $P_i^{0\to 1} = c_1 + dn_i$, where c_1 represents the individual strategy, which is independent of the influence of its neighbors. The parameter d quantifies the *herding behavior*, whereby a nonadopted-state individual copies the strategy of their n_i adopted-state neighbors. A node i with n_i of its k_i neighbors in the adopted state has $k_i - n_i$ neighbors in the nonadopted state. Due to the herding effect, the probability for the transition from an adopted state to a nonadopted state is $P_i^{1\to 0} = c_2 + d(k_i - n_i)$. In our simulations, we set $c_1 = c_2 = 0.1$ and $d = 10/N$, where N is the network size.

SIS and CP. In the susceptible-infected-susceptible (SIS) disease-spread model [70,72,83], an infected individual transmits disease to node i in its neighborhood with the probability λ_i. If a susceptible node i has n_i infected neighbors, the probability that it will be infected is $P_i^{0\to 1}$. The recovery rate $P_i^{1\to 0} = \mu_i$ is kept constant for node i. When simulating the SIS model, all infected neighbors of node i affect i. In particular, for each infected neighbor, a random number is drawn and checked if it is smaller than λ_i, and i gets infected if any of the random numbers are smaller than λ_i. We thus have $P_i^{0\to 1} = 1 - (1 - \lambda)^{n_i}$, where the second term gives the chance for i to remain susceptible.

For the contact process (CP) model [68,69], at a time step, each node i randomly selects one of its k_i neighbors and checks

TABLE IV. Description of the nine real world networks used in our study (N, number of nodes; L, number of edges).

Type	Index	Name	N	L	Description
Trust	1	College student [77,78]	32	96	Social network
	2	Prison inmate [77,78]	67	182	Social network
Protein	4	Protein-1 [78]	95	213	Protein network
	5	Protein-2 [78]	53	123	Protein network
	6	Protein-3 [78]	99	212	Protein network
Food web	12	Seagrass [79]	49	226	Food web
	13	Grassland [80]	88	137	Food web
	14	Ythan [80]	135	601	Food web
Circuits	3	s208a [81]	122	189	Logic circuit

whether it is an infected node. If yes, it will transmit the disease to i with the probability λ_i. Since the chance of getting an infected node selected is n_i/k_i, the probability of infecting node i at the current time step is $(n_i/k_i)\lambda_i$.

Evolutionary game models: SG, SQ-SG, PDG, and SQ-PDG. The implementation of evolutionary game dynamics, PDG [75,76,84,85] and SG [74], is slightly more complicated. In the PDG model, the following two processes occur at each time step. (1) *Game playing and payoffs.* Each agent plays the classical prisoner's dilemma game (PDG) with all its nearest neighbors, and the total payoff is the sum of the payoffs gained in its two-player games with all the connected agents. Each player may choose either to cooperate, C, or to defect, D, in a single encounter. If both players choose C, both will get payoff R. If one defects while the other cooperates, D gets T, while C gets S. If both defect, both get P, where $T > R > P > S$. The PDG parameters in our work are chosen to be $R = 1$, $T = b > 1$, and $S = P = 0$ [75]. (2) *Strategy updating.* At each time step, agent i randomly chooses a neighbor j and imitates j's strategy with the probability $P_{i\to j} = \{1 + \exp[-(U_j - U_i)/\kappa]\}^{-1}$, where U_i and U_j are the payoffs for agents i and j, and κ is the level of agents' "rationality" representing the uncertainties in assessing the best strategy. The only difference between PDG and SG is the choice of the game parameter values. In SG, the parameters are chosen to be $R = 1$, $T = 1 + r$, $S = 1 - r$, and $P = 0$, where $0 < r < 1$. The probability that node i chooses to be cooperative or defective at the next time step requires information about its neighbors' states and payoffs at the current time step. This violates the interdependency assumption underlying a Markov network. It is thus difficult to analytically calculate the probabilities $P_i^{0\to1}$ and $P_i^{1\to0}$. We find, however, that the structure of this non-Markov network can be successfully reconstructed by SDBM, a Markov network.

A self-questioning mechanism [54] can be introduced to ensure the interdependency properties of the Markov networks for PDG and SG. At the strategy updating stage, we define the probability that a cooperative node i chooses a defective strategy at the next time step to be $P_i^{0\to1} = \{1 + \exp[-(U_i^{\mathrm{C}} - U_i^{\mathrm{D}})/\kappa]\}^{-1}$, where U_i^{C} and U_i^{D} are the payoffs obtained while adopting the cooperative and defective strategies, respectively. The probability of choosing a cooperative strategy is $P_i^{1\to0} = 1 - P_i^{0\to1}$. For PDG, we have $U_i^{\mathrm{C}} = k_i - n_i$ and $U_i^{\mathrm{D}} = b(k_i - n_i)$. For SG, we have $U_i^{\mathrm{C}} = (k_i - n_i) + (1 - r)n_i$ and $U_i^{\mathrm{D}} = (1 + r)(k_i - n_i)$, where n_i denotes the number of defective neighbors. In our simulations, nodal states 0 and 1 denote cooperation and defection, respectively, and we set $b = 1.2$, $r = 0.7$, and $\kappa = 2$ for all cases of the evolutionary game dynamics.

APPENDIX D: REAL-WORLD NETWORKS

Table IV lists the names, types, sizes of the real world networks used for topology reconstruction in the main text.

TABLE V. Structure reconstruction error rates of 5 real world networks associated with 11 typical complex network dynamics. The percentage error rates of the nonexistent and the existent links detection ($R_{\mathrm{e}}^0/R_{\mathrm{e}}^1$, %) are listed for all combinations of the network structure and dynamics. Each entry denotes the error rates $R_{\mathrm{e}}^0/R_{\mathrm{e}}^1$ corresponding to a particular type of dynamics (shown in the column) for a real world network topology (shown in the row). The result for each combination is obtained with $M = 0.4N$ measurements used in each implementation. For some combinations, there exists a global absorption state in which the system state configuration freezes, or a vibration state where the system switches between only a small number of configurations. In such a case, no sufficient number of linearly independent measurements can be obtained to implement compressive sensing. The entries associated with these cases are marked by "* / *".

Dynamics type	Social-1	Social-2	Protein-1	Protein-2	Protein-3
Ising-Glauber	0.2 / 3.8	0.0 / 0.0	0.0 / 0.0	1.0 / 0.0	0.0 / 0.0
Minority game	0.0 / 0.0	0.0 / 7.0	3.7 / 0.0	* / *	* / *
Voter	0.0 / 0.0	0.0 / 4.2	3.9 / 0.8	* / *	* / *
Majority vote	0.0 / 0.0	0.0 / 32.4	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Link-update voter	3.0 / 2.5	0.1 / 7.0	0.2 / 0.0	0.0 / 0.0	2.7 / 0.0
Language model	0.2 / 0.0	0.0 / 0.0	0.1 / 0.0	2.1 / 0.0	* / *
Kirman	0.0 / 0.0	0.5 / 34.5	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
CP	0.2 / 0.0	0.0 / 7.0	0.0 / 0.0	* / *	* / *
SIS	4.7 / 2.5	0.0 / 14.1	0.4 / 0.0	* / *	* / *
SG	0.9 / 5.0	0.5 / 2.8	1.8 / 0.0	1.6 / 0.0	* / *
PDG	0.5 / 0.0	0.9 / 4.2	2.4 / 0.0	3.3 / 0.0	13.9 / 16.0

TABLE VI. Structure reconstruction error rates of 4 additional real world networks associated with 11 types of complex network dynamics. Table legends are the same as for Table V.

Dynamics type	Food web-1	Food web-2	Food web-3	Circuit
Ising-Glauber	* / *	0.0 / 0.0	* / *	0.0 / 0.0
Minority game	0.1 / 0.0	1.0 / 0.0	0.4 / 0.3	0.8 / 0.0
Voter	0.1 / 0.0	0.7 / 0.0	0.6 / 1.7	2.0 / 0.0
Majority vote	0.4 / 0.0	0.0 / 0.0	1.5 / 0.0	0.0 / 0.0
Link-update voter	1.1 / 0.0	15.7 / 0.0	* / *	* / *
Language model	0.2 / 0.0	0.3 / 0.0	0.0 / 4.4	0.0 / 0.1
Kirman	* / *	0.0 / 2.2	* / *	0.0 / 0.0
CP	0.4 / 0.5	0.0 / 0.7	5.3 / 1.5	0.0 / 0.0
SIS	1.3 / 9.4	4.7 / 8.0	17.3 / 17.3	* / *
SG	2.1 / 16.1	8.4 / 25.6	6.2 / 23.0	8.3 / 4.8
PDG	2.8 / 7.6	8.4 / 19.0	8.8 / 15.3	23.0 / 14.3

APPENDIX E: RECONSTRUCTION ACCURACY FOR REAL WORLD NETWORKS

Tables V and VI list the reconstruction accuracy for a number of real world networks with various dynamical processes.

APPENDIX F: VISUAL COMPARISON BETWEEN THE CONDITIONAL PROBABILITIES

Figures 9(a)–9(d) present a visual comparison between the conditional probabilities $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ of the original SDBM and of the estimated SDBM calculated from Eq. (5) in the main text. We see that the two time series of $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ are visually indistinguishable.

APPENDIX G: SDBM AS A GENERATIVE MODEL FOR DIFFERENT TYPES OF DYNAMICS

In real world applications, it is often the case that little information about the details of the dynamical processes is available, i.e., the time series data can be regarded as from a “black box.” Say the data generation mechanism can be uncovered based solely on the observed data, i.e., the black-box dynamics is simulated to reproduce the “pseudo”-time series bearing the same statistical properties of the original data. The model is deemed generative in the sense that it has the same data generating mechanism as that of the original system. Such a generative model captures the essential dynamical and statistical properties of the original system, which can be exploited for long-term statistical prediction through simulation. Here, we investigate the potential for the SDBM approximator to serve as a generative model of the original dynamics. A basic requirement is that the long-term conditional probability recovery error accumulated at each immediate time step does not drive the model behavior significantly off the trajectory generated by the original dynamical process. Due to the stochastic nature of the original dynamical system and its approximator, starting from a particular state configuration, reproducing similar conditional probabilities in the immediate future does not guarantee a similar state configuration. In fact, the deviation in the configuration at the next time step may lead to further and larger deviations, and so on.

Figure 10 shows three examples: two positive cases where the prediction is reasonable and one negative case where the task fails. For those examples, the approximator is initialized from a random configuration and runs for the same amount of time as the original time series. In the positive cases (SQ-SG and SIS), the conditional probability time series generated by the approximator are visually similar to the original data,

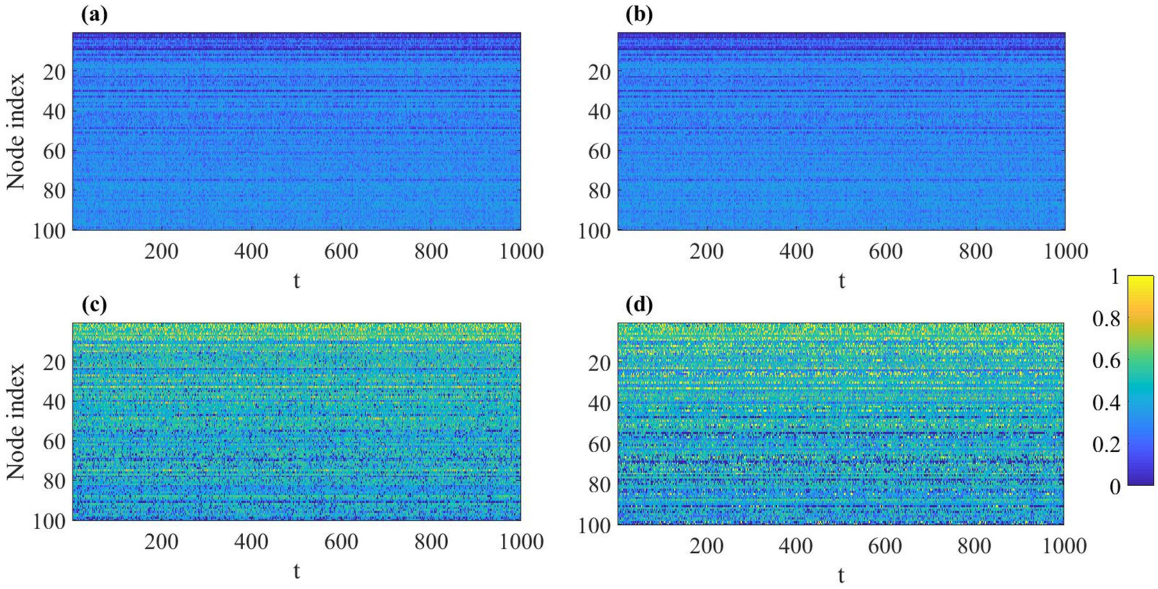

FIG. 9. Comparison between the conditional probabilities of the original system and that generated via the reconstructed SDBM. (a) Conditional probability $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ versus time t for the original system with SDBM dynamics for a BA scale-free network and (b) the conditional probability from the reconstructed approximator using the configuration time series of the original system, where $1 \leqslant i \leqslant N$ (the y axis) and $1 \leqslant t \leqslant T$ (the x axis). Time series of the conditional probability $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ from the original system with SIS dynamics for a BA scale-free network (c) and from the reconstructed approximator (d).

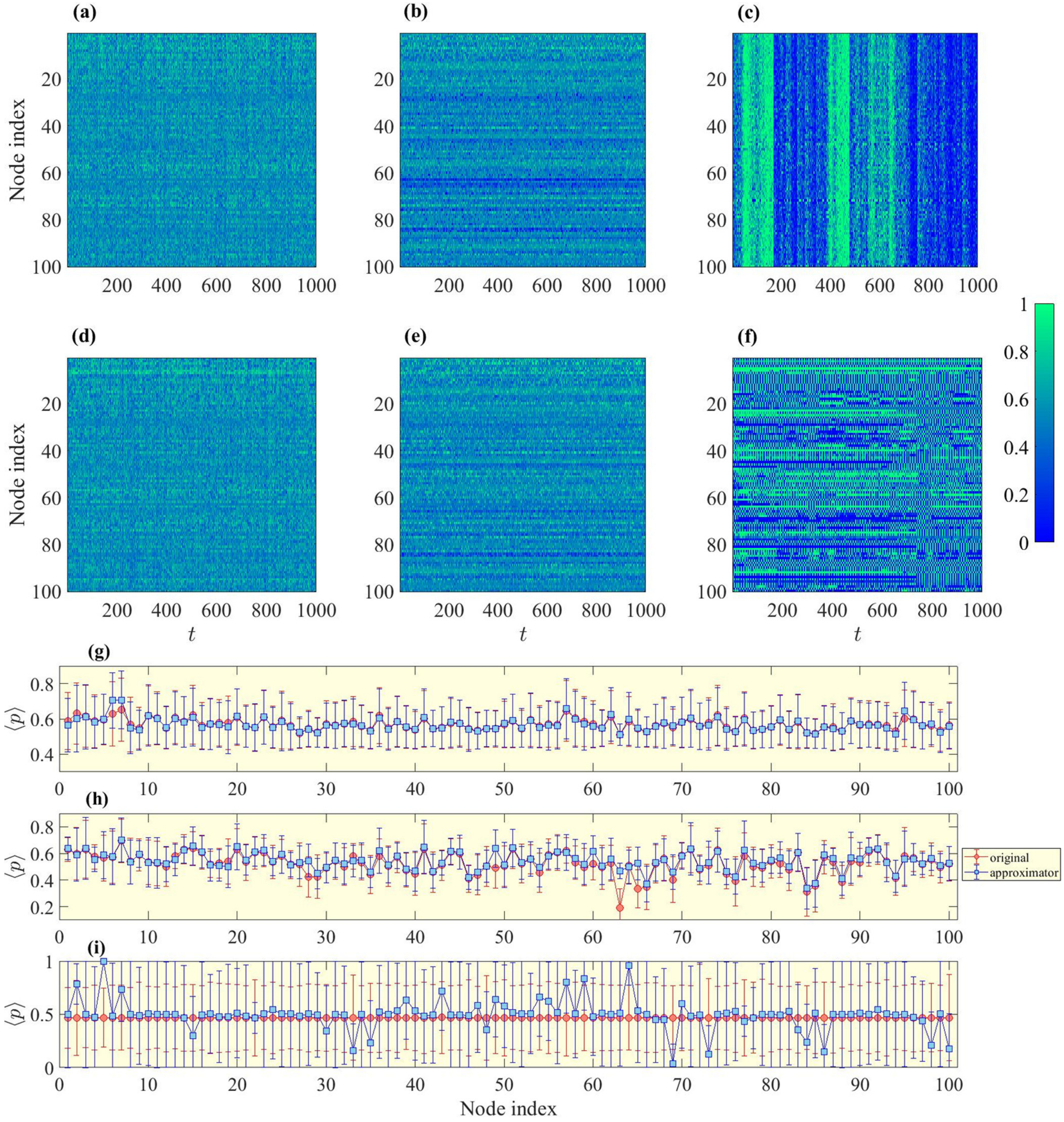

FIG. 10. Comparison between the conditional probabilities of the original system and the approximator using the self-generated state configuration time series. (a), (d), (b), (e), and (c), (f): the conditional probability $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ time series generated from the original state configuration time series (upper panel) and from the SDBM approximator using the state configuration time series generated by itself (lower panel) for SQ-SG, SIS, and voter dynamics on ER random networks, respectively. (g)–(i) The mean and standard deviations of $p=P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ for each node i for all time steps for SQ-SG (g), SIS (h), and voter dynamics (i), respectively, on ER random networks, calculated from the original system (red circles) and the approximator (blue squares), where $1 \leqslant i \leqslant N$ (the x axis).

indicating that the observed data are qualitatively and quantitatively reproduced by the approximator. In addition, for each node, its conditional probability value $P\{x_i(t+1)=1|\mathbf{X}_i^{\mathrm{R}}(t)\}$ can be treated as a random variable p, whose distributions in the true system should be similar to that produced by the approximator if it can indeed serve as a generative model. To characterize the similarity in the distributions, one could use some standard statistical measures such as the Kullback-Leibler divergence [86], the Jensen-Shannon divergence [87], the Bhattacharyya distance [88], or the Hellinger distance [89]. However, we find that the performance of these metrics is not stable. It is more effective to directly compare each node's $\langle p \rangle$

and Δp values from the original data and the approximator, e.g., for the three examples shown in Figs. 10(c), 10(f), and 10(i). The result is that, except for a few nodes, the values of $\langle p \rangle$ and Δp obtained from the data and SDBM agree with each other for the two positive cases, but there is a substantial difference for the negative case. Our calculation also shows that there are cases where the approximator reproduces similar values of $\langle p \rangle$ and Δp for a substantial fraction of the nodes in the network, but with deviations for the remaining nodes. We conclude that the SDBM approximator can serve as a generative model for some systems. Improvement is necessary to generalize the generative power of SDBM.

We have also carried out a comparison study between the conditional probabilities of the original system and of the approximator using its self-generated state configuration time series for all types of dynamics studied except for the non-Markovian classic PDG and SG, whose conditional probability time series are difficult to compute. We find that in about half of the cases, namely, SDBM itself, Ising-Glauber, SQ-SG, SQ-PDG, the Kirman model, CP, and SIS, the reconstructed SDBM approximators are capable of serving as a generative model of the original time series data, so the reconstructed SDBMs can be used to simulate the long-term behavior of the corresponding original systems.

[1] S. Grün, M. Diesmann, and A. Aertsen, Neural Comput. **14**, 43 (2002).

[2] S. Grün, A. Aertsen, and S. Rotter, Neural Comput. **14**, 121 (2002).

[3] G. Pipa and S. Grün, Neurocomputing **52-54**, 31 (2003).

[4] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, Science **301**, 102 (2003).

[5] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo, Mol. Syst. Biol. **3**, 78 (2007).

[6] F. Geier, J. Timmer, and C. Fleck, BMC Syst. Biol. **1**, 11 (2007).

[7] K. Supekar, V. Menon, D. Rubin, M. Musen, and M. D. Greicius, PLoS Comput. Biol. **4**, e1000100 (2008).

[8] M. Hecker, S. Lambeck, S. Toepferb, E. van Someren, and R. Guthke, BioSystems **96**, 86 (2009).

[9] T. Tanaka, Phys. Rev. E **58**, 2302 (1998).

[10] S. Cocco and R. Monasson, Phys. Rev. Lett. **106**, 090601 (2011).

[11] E. Aurell and M. Ekeberg, Phys. Rev. Lett. **108**, 090201 (2012).

[12] H. C. Nguyen and J. Berg, Phys. Rev. Lett. **109**, 050602 (2012).

[13] F. Ricci-Tersenghi, J. Stat. Mech. (2012) P08015.

[14] Y. Roudi and J. Hertz, Phys. Rev. Lett. **106**, 048702 (2011).

[15] M. Méard and J. Sakellariou, J. Stat. Mech. (2011) P08015.

[16] J. Sohl-Dickstein, P. B. Battaglino, and M. R. DeWeese, Phys. Rev. Lett. **107**, 220601 (2011).

[17] P. Zhang, J. Stat. Phys. **148**, 502 (2012).

[18] H.-L. Zeng, M. Alava, E. Aurell, J. Hertz, and Y. Roudi, Phys. Rev. Lett. **110**, 210601 (2013).

[19] L. Barnett, J. T. Lizier, M. Harré, A. K. Seth, and T. Bossomaier, Phys. Rev. Lett. **111**, 177203 (2013).

[20] A. Decelle and F. Ricci-Tersenghi, Phys. Rev. Lett. **112**, 070603 (2014).

[21] M. Timme, Phys. Rev. Lett. **98**, 224101 (2007).

[22] J. Bongard and H. Lipson, Proc. Natl. Acad. Sci. USA **104**, 9943 (2007).

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

[24] A. Clauset, C. Moore, and M. E. J. Newman, Nature (London) **453**, 98 (2008).

[25] Z. Levnajić and A. Pikovsky, Phys. Rev. Lett. **107**, 034101 (2011).

[26] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, Europhys. Lett. **94**, 48006 (2011).

[27] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J.-P. Ye, Phys. Rev. X **1**, 021021 (2011).

[28] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Nat. Commun. **5**, 4323 (2014).

[29] X. Han, Z. Shen, W.-X. Wang, and Z. Di, Phys. Rev. Lett. **114**, 028701 (2015).

[30] W. Pan, Y. Yuan, J. Goncalves, and G.-B. Stan, IEEE Trans. Autom. Control **61**, 182 (2016).

[31] W. Pan, A. Sootla, and G.-B. Stan, IEEE Proc. **47**, 3208 (2014).

[32] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).

[33] A. M. and J. Goncalves, in *2016 IEEE 55th Conference on Decision and Control (CDC)* (IEEE, Piscataway, NJ, 2016), pp. 6500–6506.

[34] J. Li, Z. Shen, W.-X. Wang, C. Grebogi, and Y.-C. Lai, Phys. Rev. E **95**, 032303 (2017).

[35] C. Ma, H.-S. Chen, Y.-C. Lai, and H.-F. Zhang, Phys. Rev. E **97**, 022301 (2018).

[36] W.-X. Wang, Q. Chen, L. Huang, Y.-C. Lai, and M. A. F. Harrison, Phys. Rev. E **80**, 016116 (2009).

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

[38] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006).

[39] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. (Prentice Hall, Englewood Cliffs, NJ, 2009).

[40] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cogn. Sci. **9**, 147 (2014).

[41] E. J. Candès, J. K. Romberg, and T. Tao, IEEE Trans. Inf. Theory **52**, 489 (2006).

[42] E. J. Candes, J. K. Romberg, and T. Tao, Commun. Pure Appl. Math. **59**, 1207 (2006).

[43] D. L. Donoho, IEEE Trans. Inf. Theory **52**, 1289 (2006).

[44] R. G. Baraniuk, IEEE Sig. Proc. Mag. **24**, 118 (2007).

[45] E. J. Candès and M. B. Wakin, IEEE Sig. Proc. Mag. **25**, 21 (2008).

[46] J. Romberg, IEEE Sig. Proc. Mag. **25**, 14 (2008).

[47] R.-Q. Su, X. Ni, W.-X. Wang, and Y.-C. Lai, Phys. Rev. E **85**, 056220 (2012).

[48] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, Phys. Rev. E **85**, 065201(R) (2012).

[49] R.-Q. Su, Y.-C. Lai, X. Wang, and Y.-H. Do, Sci. Rep. **4**, 3944 (2014).

[50] V. Kishore, M. S. Santhanam, and R. E. Amritkar, Phys. Rev. Lett. **106**, 188701 (2011).

[51] Y.-Z. Chen, Z.-G. Huang, and Y.-C. Lai, Sci. Rep. **4**, 06121 (2014).

[52] Y.-Z. Chen, Z.-G. Huang, H.-F. Zhang, D. Eisenberg, T. P. Seager, and Y.-C. Lai, Sci. Rep. **5**, 17277 (2015).

[53] R. J. Glauber, J. Math. Phys. (NY) **4**, 294 (1963).

[54] K. Gao, W.-X. Wang, and B.-H. Wang, Phys. A (Amsterdam) **380**, 528 (2007).

[55] D. Challet and Y.-C. Zhang, Phys. A (Amsterdam) **246**, 407 (1997).

[56] D. Challet, M. Marsili, and Y.-C. Zhang, *Minority Games: Interacting Agents in Financial Markets* (Oxford University Press, Oxford, 2005).

[57] M. Paczuski, K. E. Bassler, and A. Corral, Phys. Rev. Lett. **84**, 3185 (2000).

[58] T. Zhou, B.-H. Wang, P.-L. Zhou, C.-X. Yang, and J. Liu, Phys. Rev. E **72**, 046139 (2005).

[59] Z.-G. Huang, J.-Q. Zhang, J.-Q. Dong, L. Huang, and Y.-C. Lai, Sci. Rep. **2**, 703 (2012).

[60] T. M. Liggett, *Interacting Particle Systems* (Springer, New York, 1985).

[61] M. J. de Oliveira, J. Stat. Phys. **66**, 273 (1992).

[62] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, Cambridge, England, 2008).

[63] C. Castellano, S. Fortunato, and V. Loreto, Rev. Mod. Phys. **81**, 591 (2009).

[64] F. Vazquez, X. Castelló, and M. S. Miguel, J. Stat. Mech. (2010) P04007.

[65] D. M. Abrams and S. H. Strogatz, Nature (London) **424**, 900 (2003).

[66] A. Kirman, Quarterly J. Econom. **108**, 137 (1993).

[67] S. Alfarano, T. Lux, and F. Wagner, Comput. Econom. **26**, 19 (2005).

[68] C. Castellano and R. Pastor-Satorras, Phys. Rev. Lett. **96**, 038701 (2006).

[69] E. Volz and L. A. Meyers, J. R. Soc. Interface **6**, 233 (2009).

[70] N. T. J. Bailey, *The Mathematical Theory of Infectious Diseases* (Griffin, London, 1975).

[71] R. M. Anderson and R. M. May, *Infectious Diseases in Humans* (Oxford University Press, Oxford, England, 1992).

[72] R. Pastor-Satorras and A. Vespignani, Phys. Rev. Lett. **86**, 3200 (2001).

[73] R. Pastor-Satorras and A. Vespignani, Phys. Rev. E **63**, 066117 (2001).

[74] R. Sugden, *The Economics of Rights Co-operation and Welfare* (Blackwell, Oxford, UK, 1986).

[75] R. Axelrod and W. D. Hamilton, Science **211**, 1390 (1981).

[76] G. Szabó and C. Tőke, Phys. Rev. E **58**, 69 (1998).

[77] M. A. J. V. Duijn, M. Huisman, F. N. Stokman, F. W. Wasseur, and E. P. H. Zeggelink, J. Math. Sociol. **27**, 153 (2003).

[78] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, Science **303**, 1538 (2004).

[79] R. R. Christian and J. J. Luczkovich, Ecol. Modell. **117**, 99 (1999).

[80] J. A. Dunne, R. J. Williams, and N. D. Martinez, Proc. Natl. Acad. Sci. USA **99**, 12917 (2002).

[81] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Science **298**, 824 (2002).

[82] V. Sood and S. Redner, Phys. Rev. Lett. **94**, 178701 (2005).

[83] P. J. Antsaklis and A. N. Michel, *Linear Systems* (McGraw-Hill, New York, 1997).

[84] Y.-Z. Chen, Z.-G. Huang, S.-J. Wang, Y. Zhang, and Y.-H. Wang, Phys. Rev. E **79**, 055101(R) (2009).

[85] Y.-Z. Chen and Y.-C. Lai, Phys. Rev. E **86**, 045101(R) (2012).

[86] S. Kullback and R. A. Leibler, Ann. Math. Stat. **22(1)**, 79 (1951).

[87] J. Burbea and C. R. Rao, IEEE Trans. Inf. Theory **28**, 489 (1982).

[88] A. Bhattacharyya, Sankhyã: Indian J. Stat. **7**, 401 (1946).

[89] E. Hellinger, J. für die reine und angewandte Math. (Crelle's Journal) **136**, 210 (1909).